Annual DNA collection from Murphy & Tong Study

Tina Lasisi & João P. Donadio

2025-10-04

Table of contents

1 Executive Summary

This document provides complete transparency regarding the data sources and methodology used to compile racial disparities in DNA collection across U.S. states. The original data was collected in Summer 2017, with most data points from 2013-2016.

1.1 Key Findings

- **Consistent racial disparities**: Black populations show the highest DNA collection rates relative to their population percentage in nearly all states
- **Data limitations**: Many states lack comprehensive conviction data, requiring the use of prison admission proxies
- **Methodological challenges**: Hispanic/Latino populations often uncounted or miscategorized in state data

2 Methodology Overview

2.1 Data Collection Period

- Primary collection: Summer 2017
- Data years used: Single year per state (2012-2016, varies by availability)
- Census baseline: 2010 U.S. Census for demographic comparisons

2.2 General Challenges Encountered

1. **Conviction Data Scarcity**: Most states do not publicly disclose comprehensive felony conviction data
2. **Prison Admission Proxy**: Prison admissions used as substitute for conviction data in majority of states
3. **Racial Classification Inconsistencies**:

- `Many states only report "Black" and "White" categories`
- `Hispanic/Latino often classified as ethnicity rather than race`
- `"Other" category frequently used without specification`

4. **Arrest Data Gaps**: Racial makeup of arrests often unavailable or incomplete

3 National Summary Table

3.1 Methodology: Parsing DNA Collection Data

The national summary data was extracted from a structured text file (`MurphyTong_Racial_Breakdown.txt`) containing three distinct data sections for each state:

1. **DNA Collection Data**: Percentage and absolute counts of DNA profiles collected by race (e.g., "46% B (18,253)")
2. **Population Demographics**: Percentage of state population by race from 2010 Census

Processing Steps:

- **State Identification**: The parser identifies state entries using standard two-letter abbreviations (AL, AK, AZ, etc.)
- **Section Segmentation**: For each state, the text is divided into three sections based on the pattern of "% B" (Black percentage) occurrences
- **Data Extraction**: Regular expressions extract both percentages and counts from the first section, and percentages only from the demographic and rate sections
- **Pattern Matching**: The code uses regex patterns like `([0-9.]+)%\s*B\s*\(([0-9,]+)\)` to capture both percentage (e.g., 46%) and count (e.g., 18,253) for DNA collection data
- **Race Categories**: Data is extracted for five racial categories: Black (B), Hispanic (H), Asian (A), Native American (N), and White (W)

This automated parsing ensures reproducibility and allows for systematic extraction of data from the original Murphy & Tong study format.

```r
# List of required packages
required_packages <- c(
  "tidyverse",   # Data manipulation and visualization
  "knitr",       # Dynamic report generation
  "dplyr",       # Data manipulation
  "stringr",     # String manipulation
  "purrr",       # Functional programming
  "here",        # File path management
  "kableExtra"   # Table formatting for PDF
)

# Function to install missing packages
install_missing <- function(packages) {
  for (pkg in packages) {
    if (!requireNamespace(pkg, quietly = TRUE)) {
      message(paste("Installing missing package:", pkg))
      install.packages(pkg, dependencies = TRUE)
    }
  }
}

# Install any missing packages
install_missing(required_packages)

# Load all packages
suppressPackageStartupMessages({
  library(tidyverse)
```

```r
  library(knitr)
  library(dplyr)
  library(stringr)
  library(purrr)
  library(here)
  library(kableExtra)
})

# Verify all packages loaded successfully
loaded_packages <- sapply(required_packages, require, character.only = TRUE)

if (all(loaded_packages)) {
  message("All packages loaded successfully!")
} else {
  warning("The following packages failed to load: ",
          paste(names(loaded_packages)[!loaded_packages], collapse = ", "))
}
```

```r
# Function to parse DNA collection data from text file
parse_dna_data <- function(file_path) {

  # Read the entire file
  text_data <- readLines(file_path, warn = FALSE)

  # Remove empty lines
  text_data <- text_data[text_data != ""]

  # Initialize list to store parsed data
  parsed_data <- list()

  # State abbreviations (for reference)
  state_abbrevs <- c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA",
                     "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
                     "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ",
                     "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
                     "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY", "DC")

  # Function to extract percentage and count from patterns like "46% B (18,253)"
  extract_race_data <- function(text, race_letter) {
    pattern <- paste0("([0-9.]+)%\\s*", race_letter, "\\s*\\(([0-9,]+)\\)")
    matches <- str_match(text, pattern)
    if (!is.na(matches[1])) {
      pct <- as.numeric(matches[2])
      count <- as.numeric(gsub(",", "", matches[3]))
      return(list(pct = pct, count = count))
    }
    return(list(pct = NA, count = NA))
```

```
}

# Function to extract just percentage (for demographics and collection rates)
extract_percentage <- function(text, race_letter) {
  pattern <- paste0("([0-9.]+)%\\s*", race_letter)
  matches <- str_match(text, pattern)
  if (!is.na(matches[1])) {
    return(as.numeric(matches[2]))
  }
  return(NA)
}

# Process each line
i <- 1
while (i <= length(text_data)) {
  line <- text_data[i]

  # Check if line is a state abbreviation
  if (line %in% state_abbrevs) {
    state <- line

    # Initialize data structure for this state
    state_data <- list(
      State = state,
      Black_DNA_Pct = NA, Black_DNA_N = NA,
      Hispanic_DNA_Pct = NA, Hispanic_DNA_N = NA,
      Asian_DNA_Pct = NA, Asian_DNA_N = NA,
      Native_American_DNA_Pct = NA, Native_American_DNA_N = NA,
      White_DNA_Pct = NA, White_DNA_N = NA,
      Black_Pop_Pct = NA, Hispanic_Pop_Pct = NA, Asian_Pop_Pct = NA,
      Native_American_Pop_Pct = NA, White_Pop_Pct = NA,
      Black_Collection_Rate = NA, Hispanic_Collection_Rate = NA,
      Asian_Collection_Rate = NA, Native_American_Collection_Rate = NA,
      White_Collection_Rate = NA
    )

    # Collect all lines for this state until we hit the next state or end of file
    state_lines <- c()
    i <- i + 1
    while (i <= length(text_data) && !(text_data[i] %in% state_abbrevs)) {
      state_lines <- c(state_lines, text_data[i])
      i <- i + 1
    }

    # Combine all lines for this state into one text block
    state_text <- paste(state_lines, collapse = " ")
```

```r
    # Split into sections based on the pattern of % B occurrences
    # We need to identify the three sections: DNA Collection, Demographics, Collection Rates

    # Find all "% B" patterns to help identify sections
    b_patterns <- str_locate_all(state_text, "[0-9.]+%\\s*B")[[1]]

    if (nrow(b_patterns) >= 1) {
      # First section: DNA Collection (has counts in parentheses)
      dna_section_start <- 1
      dna_section_end <- if(nrow(b_patterns) >= 2) b_patterns[2,1] - 1 else nchar(state_text)
      dna_section <- substr(state_text, dna_section_start, dna_section_end)

      # Extract DNA collection data (with counts)
      black_dna <- extract_race_data(dna_section, "B")
      hispanic_dna <- extract_race_data(dna_section, "H")
      asian_dna <- extract_race_data(dna_section, "A")
      native_dna <- extract_race_data(dna_section, "N")
      white_dna <- extract_race_data(dna_section, "W")

      state_data$Black_DNA_Pct <- black_dna$pct
      state_data$Black_DNA_N <- black_dna$count
      state_data$Hispanic_DNA_Pct <- hispanic_dna$pct
      state_data$Hispanic_DNA_N <- hispanic_dna$count
      state_data$Asian_DNA_Pct <- asian_dna$pct
      state_data$Asian_DNA_N <- asian_dna$count
      state_data$Native_American_DNA_Pct <- native_dna$pct
      state_data$Native_American_DNA_N <- native_dna$count
      state_data$White_DNA_Pct <- white_dna$pct
      state_data$White_DNA_N <- white_dna$count
    }

    if (nrow(b_patterns) >= 2) {
      # Second section: Demographics (percentages only, no parentheses)
      demo_section_start <- b_patterns[2,1]
      demo_section_end <- if(nrow(b_patterns) >= 3) b_patterns[3,1] - 1 else nchar(state_text)
      demo_section <- substr(state_text, demo_section_start, demo_section_end)

      # Extract demographic percentages
      state_data$Black_Pop_Pct <- extract_percentage(demo_section, "B")
      state_data$Hispanic_Pop_Pct <- extract_percentage(demo_section, "H")
      state_data$Asian_Pop_Pct <- extract_percentage(demo_section, "A")
      state_data$Native_American_Pop_Pct <- extract_percentage(demo_section, "N")
      state_data$White_Pop_Pct <- extract_percentage(demo_section, "W")
    }

    if (nrow(b_patterns) >= 3) {
      # Third section: Collection Rates (percentages only, no parentheses)
```

```
        rate_section_start <- b_patterns[3,1]
        rate_section <- substr(state_text, rate_section_start, nchar(state_text))

        # Extract collection rate percentages
        state_data$Black_Collection_Rate <- extract_percentage(rate_section, "B")
        state_data$Hispanic_Collection_Rate <- extract_percentage(rate_section, "H")
        state_data$Asian_Collection_Rate <- extract_percentage(rate_section, "A")
        state_data$Native_American_Collection_Rate <- extract_percentage(rate_section, "N")
        state_data$White_Collection_Rate <- extract_percentage(rate_section, "W")
      }

      # Add to parsed data
      parsed_data[[length(parsed_data) + 1]] <- state_data

      # Don't increment i here since we already did it in the while loop
      i <- i - 1
    }

    i <- i + 1
  }

  # Convert to data frame
  df <- do.call(rbind, lapply(parsed_data, data.frame))

  return(df)
}

racial_data_path <- file.path(here("data", "annual_dna_collection", "intermediate", "MurphyTong

summary_data <- parse_dna_data(racial_data_path)

# Display table
kable(summary_data, booktabs = TRUE, longtable = TRUE, caption = "Complete state-by-state breal
  kable_styling(
    latex_options = c("striped", "scale_down", "repeat_header"),
    font_size = 9,
    position = "center"
  )
```

State	Black_DNA_Pct	Black_DNA_N	Hispanic_DNA_Pct	Hispanic_DNA_N	Asian_DNA_Pct	Asian_DNA_N
AL	46.0	18253	NA	NA	NA	NA
AK	7.6	914	2.8	23604	3.00	365
AZ	12.6	9313	32.0	23604	0.50	358
AR	42.3	4401	2.7	275	0.04	4
CA	25.0	141488	37.6	213361	NA	NA

State	Black_DNA_Pct	Black_DNA_N	Hispanic_DNA_Pct	Hispanic_DNA_N	Asian_DNA_Pct	Asian_DNA_N
CO	12.8	28486	24.0	53778	0.03	61
CT	25.5	2250	18.1	1591	NA	NA
DE	79.9	9399	5.6	660	NA	NA
FL	34.8	280152	0.5	4338	0.60	4420
GA	54.5	9851	2.4	433	0.40	72
HI	4.6	516	4.6	516	21.60	2399
ID	3.0	244	16.2	1300	NA	NA
IL	58.2	18094	12.6	3906	NA	NA
IN	24.8	3591	2.8	398	NA	NA
IA	20.8	4793	3.3	751	0.90	199
KS	26.8	13016	NA	NA	0.10	62
KY	29.3	34762	5.1	5993	NA	NA
LA	65.8	10840	4.0	657	NA	NA
ME	7.1	226	2.0	65	NA	NA
MD	62.8	34835	NA	NA	0.70	372
MA	18.5	2712	15.6	2288	NA	NA
MI	33.4	85888	0.1	357	0.40	946
MN	25.6	39743	NA	NA	2.70	4129
MS	59.6	6328	0.9	91	NA	NA
MO	31.4	3701	1.1	130	0.40	50
MT	1.9	47	NA	NA	1.00	24
NE	23.1	566	11.4	279	0.50	11
NV	28.5	1602	21.3	1201	2.40	135
NH	6.4	40	NA	NA	0.50	3
NJ	52.8	9444	8.4	1499	1.30	225
NM	6.0	157	52.0	1363	NA	NA
NY	51.8	68684	31.6	41970	3.70	4939
NC	52.3	12270	NA	NA	0.05	11
ND	9.2	2983	0.3	92	0.60	178
OH	36.2	26015	0.6	402	0.03	25
OK	26.7	2841	7.7	815	NA	NA
OR	10.2	515	16.3	824	NA	NA
PA	40.6	4159	9.7	991	NA	NA
RI	25.4	3672	20.3	2941	NA	NA
SC	54.0	3752	NA	NA	NA	NA
SD	7.3	2961	NA	NA	0.70	287
TN	35.9	14544	0.5	213	0.30	127
TX	31.2	22169	32.8	23267	NA	NA
UT	5.4	5993	0.1	141	2.40	2597
VT	10.7	295	NA	NA	0.90	25
VA	50.5	6249	2.4	294	NA	NA
WA	18.6	1376	14.4	1070	NA	NA
WV	28.0	955	6.0	205	NA	NA
WI	37.3	3482	NA	NA	1.20	111
WY	3.9	129	14.4	484	NA	NA

3.2 Disparity Analysis

```
# Calculate disparity ratios
disparity_data <- summary_data %>%
  filter(!is.na(Black_Collection_Rate) & !is.na(White_Collection_Rate)) %>%
  mutate(Black_White_Ratio = Black_Collection_Rate / White_Collection_Rate) %>%
  arrange(desc(Black_White_Ratio))

# Create visualization
ggplot(disparity_data %>% head(20), aes(x = reorder(State, Black_White_Ratio), y = Black_White_
  geom_bar(stat = "identity", fill = "darkred") +
  coord_flip() +
  labs(title = "Top 20 States: Black-White DNA Collection Disparity Ratio",
       subtitle = "Ratio of collection rates (higher = greater disparity)",
       x = "State",
       y = "Black/White Collection Rate Ratio") +
  theme_minimal() +
  geom_hline(yintercept = 1, linetype = "dashed", color = "gray50")
```

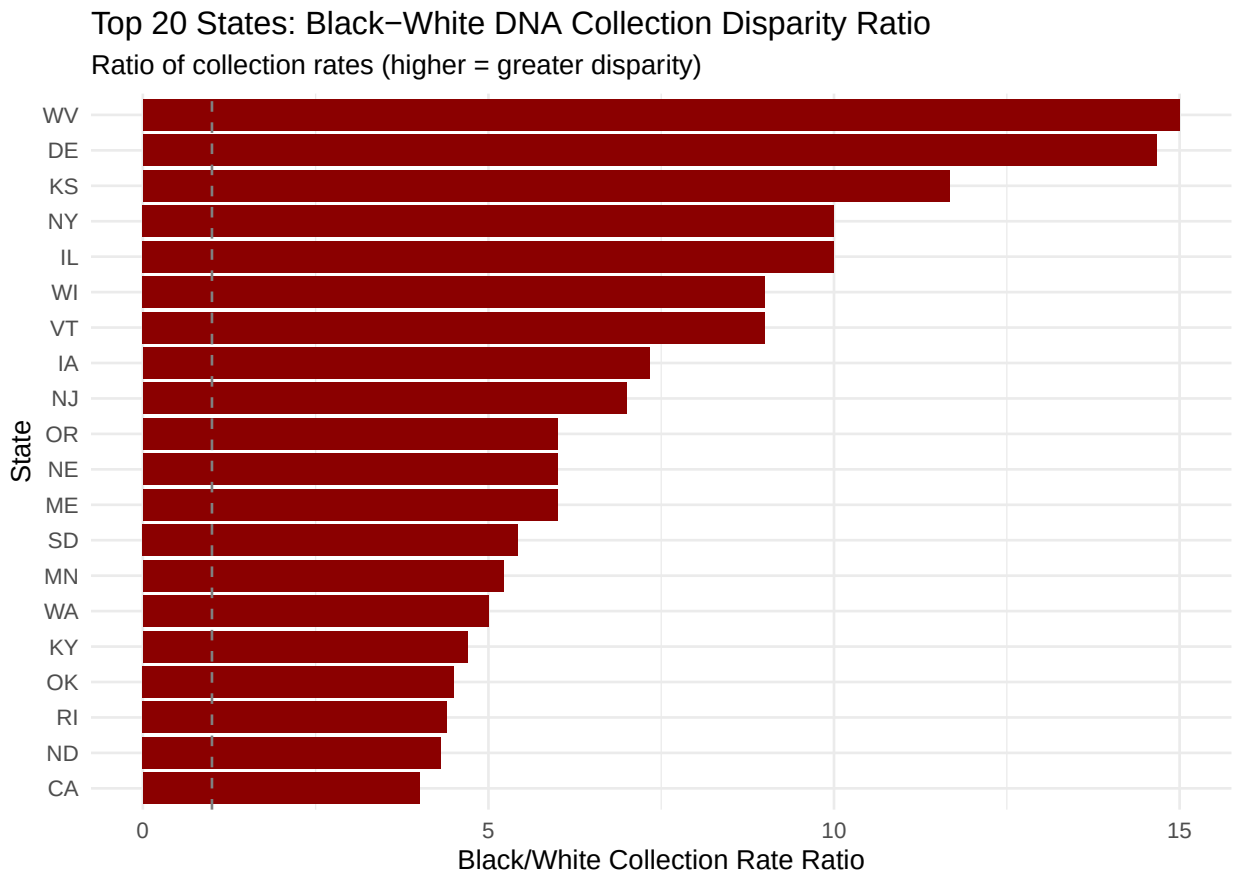

Figure 1: DNA Collection Rates by Race Relative to Population Percentage

4 State-by-State Detailed Methodology

4.1 Methodology: Parsing State Methodology Paragraphs

The detailed methodology for each state was extracted from a separate text file (`MurphyTong_States_Paragraphs.t` containing narrative descriptions of data collection approaches for all 50 states. This section explains how we systematically parsed this unstructured text into a structured dataset.

Processing Steps:

1. **State Detection**: The parser identifies state entries by searching for the 50 U.S. state names as they appear in the text
2. **Section Extraction**: For each state, the parser captures all text from the state name until the next state name appears
3. **Component Parsing**: Within each state's section, the code extracts four key components:
   - **Legal Framework**: The statutory basis for DNA collection in that state
   - **Collection Triggers**: Specific offenses or events that trigger DNA collection
   - **Data Sources**: Types of data used (e.g., conviction records, prison admissions, arrest data)
   - **Source URLs**: Web links to the original data sources
   - **Data Limitations**: Known gaps, proxies, or methodological caveats
4. **Structured Data Creation**: Each data source line is parsed into a type-note pair (e.g., "Prison admissions: Used as proxy for conviction data")
5. **Categorization**: The parser automatically categorizes:
   - **Collection trigger types**: Comprehensive, selective, felony-only, etc.
   - **Data limitation types**: Missing conviction data, ethnicity issues, limited racial categories, etc.
   - **Data source types**: Conviction data, arrest data, prison data, sex crime data, etc.

This systematic extraction allows for consistent comparison across states and identification of common patterns in data collection methodologies and limitations.

```
# Pre-define all 50 U.S. states
us_states <- c("Alabama", "Alaska", "Arizona", "Arkansas", "California", "Colorado", "Connecti
               "Delaware", "Florida", "Georgia", "Hawaii", "Idaho", "Illinois", "Indiana", "Io
               "Kansas", "Kentucky", "Louisiana", "Maine", "Maryland", "Massachusetts", "Michig
               "Minnesota", "Mississippi", "Missouri", "Montana", "Nebraska", "Nevada", "New Ha
               "New Jersey", "New Mexico", "New York", "North Carolina", "North Dakota", "Ohio
               "Oklahoma", "Oregon", "Pennsylvania", "Rhode Island", "South Carolina", "South
               "Tennessee", "Texas", "Utah", "Vermont", "Virginia", "Washington", "West Virgin
               "Wisconsin", "Wyoming")

# Read the text file
```

```r
data_file <- file.path(here("data", "annual_dna_collection", "intermediate", "MurphyTong_States

text_content <- readLines(data_file, warn = FALSE)

# Combine all lines into a single string
full_text <- paste(text_content, collapse = "\n")

# Create a pattern to match state names
state_pattern <- paste0("(?m)^[[:space:]]*(", paste(us_states, collapse = "|"), ")\\b")
state_matches <- str_locate_all(full_text, state_pattern)[[1]]

# Extract state sections
state_sections <- list()
state_names <- character()

if (nrow(state_matches) > 0) {
  for (i in 1:nrow(state_matches)) {
    start_pos <- state_matches[i, "start"]
    if (i < nrow(state_matches)) {
      end_pos <- state_matches[i + 1, "start"] - 1
    } else {
      end_pos <- nchar(full_text)
    }

    state_name <- substr(full_text, start_pos, state_matches[i, "end"])
    state_content <- substr(full_text, start_pos, end_pos)

    state_names <- c(state_names, state_name)
    state_sections <- c(state_sections, state_content)
  }
}

# Function to extract information from each state section
parse_state_section <- function(section, state_name) {
  if (is.na(state_name)) return(NULL)

  # Extract legal framework
  legal_framework <- str_extract(section, "Legal Framework:[^\n]+") %>%
    {ifelse(is.na(.), NA, str_remove(., "Legal Framework:") %>% str_trim())}

  # Extract collection triggers
  collection_triggers <- str_extract(section, "Collection Triggers:[^\n]+") %>%
    {ifelse(is.na(.), NA, str_remove(., "Collection Triggers:") %>% str_trim())}

  # Extract data sources - capture everything until Source URLs or Data Limitations
  data_sources_text <- str_extract(section, "Data Sources:[\\s\\S]*?(?=Source URLs:|Data Limita
  data_source_df <- tibble(data_source_type = NA_character_, data_source_note = NA_character_)
```

```r
if (!is.na(data_sources_text)) {
  data_sources_text <- str_remove(data_sources_text, "Data Sources:") %>% str_trim()
  # Split by newlines and clean up
  data_source_lines <- str_split(data_sources_text, "\\n")[[1]] %>%
    str_trim() %>%
    discard(~ .x == "" | str_detect(.x, "^Source URLs:|^Data Limitations:"))

  if (length(data_source_lines) > 0) {
    data_source_df <- map_df(data_source_lines, function(line) {
      if (str_detect(line, ":")) {
        tibble(
          data_source_type = str_extract(line, "^[^:]+") %>% str_trim(),
          data_source_note = str_remove(line, "^[^:]+:") %>% str_trim()
        )
      } else {
        tibble(
          data_source_type = line,
          data_source_note = NA_character_
        )
      }
    })
  }
}

# Extract source URLs
source_urls_text <- str_extract(section, "Source URLs:[\\s\\S]*?(?=Data Limitations:|$)")
source_urls <- character(0)

if (!is.na(source_urls_text)) {
  source_urls_text <- str_remove(source_urls_text, "Source URLs:") %>% str_trim()
  source_url_lines <- str_split(source_urls_text, "\\n")[[1]] %>%
    str_trim() %>%
    discard(~ .x == "" | str_detect(.x, "^Data Limitations:"))

  if (length(source_url_lines) > 0) {
    source_urls <- source_url_lines
  }
}

# Extract data limitations - capture everything until next state or end
data_limitations_text <- str_extract(section, "Data Limitations:[\\s\\S]*?(?=\\b(A|Ala|Alas|A
data_limitations <- NA_character_

if (!is.na(data_limitations_text)) {
  data_limitations_text <- str_remove(data_limitations_text, "Data Limitations:") %>% str_tri
  data_limitations_lines <- str_split(data_limitations_text, "\\n")[[1]] %>%
    str_trim() %>%
```

```
      discard(~ .x == "" | str_detect(.x, "^\\b(A|Ala|Alas|Ari|Arka|Cali|Colo|Conn|Del|Flo|Geo

    if (length(data_limitations_lines) > 0) {
      data_limitations <- paste(data_limitations_lines, collapse = "; ")
    }
  }

  # If no data sources were found, create at least one row for the state
  if (nrow(data_source_df) == 0) {
    data_source_df <- tibble(data_source_type = NA_character_, data_source_note = NA_character_
  }

  # Create result dataframe
  result_df <- data_source_df %>%
    mutate(
      state = state_name,
      legal_framework = legal_framework,
      collection_triggers = collection_triggers,
      source_url = ifelse(length(source_urls) > 0, paste(source_urls, collapse = "; "), NA_cha
      data_limitations = data_limitations,
      .before = everything()
    )

  return(result_df)
}

# Parse all state sections
state_data_list <- lapply(seq_along(state_sections), function(i) {
  parse_state_section(state_sections[[i]], state_names[i])
})

# Combine all results into one dataframe
state_data <- bind_rows(state_data_list)

# Clean up the data - remove rows where all data columns are NA
final_df <- state_data %>%
  mutate(across(where(is.character), ~ ifelse(.x == "" | is.na(.x), NA, .x))) %>%
  filter(!(is.na(data_source_type) & is.na(source_url) & is.na(data_limitations)))

# Fill in the missing legal_framework and other methodology for each state
final_df_clean <- final_df %>%
  group_by(state) %>%
  fill(legal_framework, collection_triggers, .direction = "downup") %>%
  ungroup()

# Now let's fill source_url and data_limitations across all rows for each state
final_df_clean <- final_df_clean %>%
```

```
  group_by(state) %>%
  mutate(
    source_url = ifelse(all(is.na(source_url)), NA,
                        paste(na.omit(unique(source_url)), collapse = "; ")),
    data_limitations = ifelse(all(is.na(data_limitations)), NA,
                              paste(na.omit(unique(data_limitations)), collapse = "; "))
  ) %>%
  ungroup()

# Create categorization columns for easier analysis
final_df_clean <- final_df_clean %>%
  mutate(
    # Categorize collection triggers
    collection_trigger_category = case_when(
      str_detect(collection_triggers, "(?i)all felony.*convictions.*arrests.*all felonies") ~ "
      str_detect(collection_triggers, "(?i)all felony.*convictions.*arrests.*certain|specific")
      str_detect(collection_triggers, "(?i)all felony.*convictions.*arrests") ~ "Broad: All fel
      str_detect(collection_triggers, "(?i)all felony.*convictions") ~ "Felony convictions only
      str_detect(collection_triggers, "(?i)felony.*misdemeanor.*convictions") ~ "Mixed: Felony
      TRUE ~ "Other/Unspecified"
    ),

    # Categorize data limitations
    data_limitation_category = case_when(
      is.na(data_limitations) ~ "No limitations noted",
      str_detect(data_limitations, "(?i)no direct.*conviction data") ~ "Missing conviction data
      str_detect(data_limitations, "(?i)prison admissions.*proxy") ~ "Prison data as proxy",
      str_detect(data_limitations, "(?i)hispanic|ethnicity") ~ "Ethnicity categorization issues
      str_detect(data_limitations, "(?i)racial.*limited|black.*white.*only") ~ "Limited racial
      str_detect(data_limitations, "(?i)no.*data.*available|unavailable") ~ "Various data unava
      TRUE ~ "Other limitations"
    ),

    # Categorize data source types
    data_source_category = case_when(
      str_detect(data_source_type, "(?i)conviction") ~ "Conviction Data",
      str_detect(data_source_type, "(?i)arrest") ~ "Arrest Data",
      str_detect(data_source_type, "(?i)sex|sexual") ~ "Sex Crime Data",
      str_detect(data_source_type, "(?i)prison|admission|correction") ~ "Prison/Incarceration D
      TRUE ~ "Other Data Source"
    )
  )

# Create a summary table for quick overview - ONE ROW PER STATE
methodology_summary <- final_df_clean %>%
  distinct(state, legal_framework, collection_triggers, collection_trigger_category,
           data_limitations, data_limitation_category, source_url) %>%
```

```
  arrange(state)

# Create table
kable(methodology_summary, booktabs = TRUE, longtable = TRUE, caption = "Summary of Methodology
  kable_styling(
    latex_options = c("striped", "scale_down", "repeat_header"),
    font_size = 8,
    position = "center"
  ) %>%
  column_spec(1:3, width = "6cm") %>%
  row_spec(0, bold = TRUE, background = "#f2f2f2")
```

state	legal_framework	collection_triggers
Alaska	AK Stat § 44.41.035 (2014)	All felony and sex crime misdemeanor convictions; arrests for all felonies and crimes against a person
Arizona	AZ Rev Stat §13-610 (2016)	All felony and sex crime misdemeanor convictions; arrests for certain dangerous/serious offenses and misdemeanors
Arkansas	AR Code §12-12-1109 & §12-12-1006	All felony and sex crime misdemeanor convictions; arrests for capital murder, first degree murder, kidnapping, rape, and sexual assault
California	CA Penal Code §296	All felony and sex crime misdemeanor convictions; any felony arrest
Colorado	CO Rev Stat §16-23-103 (2016)	All felony and sex crime misdemeanor convictions; any felony arrest
Connecticut	CT Gen Stat §54-102g (2012)	All felony and sex crime misdemeanor convictions; arrests for serious felonies with prior convictions
Delaware	29 DE Code §4713 (2017)	All felony and sex crime misdemeanor convictions
Florida	FL Stat §943.325 (2016)	All felony and sex crime misdemeanor convictions; stalking, voyeurism, obscene materials, gang-related offenses; felony arrests
Hawaii	HI Rev Stat §844D-31, 39 (2011)	Felony and sex crime misdemeanor convictions
Idaho	ID Code §19-5506 (2016)	All felony convictions
Illinois	730 ILCS §5/5-4-3 (2013)	All felony and sex crime misdemeanor convictions; specific violent crime arrests (with automatic expungement if not convicted)
Indiana	IN Code §10-13-6-10 (2017)	All felony convictions
Iowa	IA Code §81.2 (2016)	All felony, sex crime misdemeanor, and aggravated misdemeanor convictions
Kansas	KS Stat §21-2511 (2013)	All felony and sex crime misdemeanor convictions; felony and misdemeanor arrests
Kentucky	KY Rev Stat §17.170 (1996)	All felony and sex crime misdemeanor convictions
Louisiana	LA Rev Stat § 15:609 (2011)	All felony and sex crime misdemeanor convictions; specific misdemeanors; related arrests
Maine	25 ME Rev Stat § 1574 (2017)	All felony and sex crime misdemeanor convictions

state	legal_framework	collection_triggers
Massachusetts	MA Code Chapter 22E §3 (2006)	All felony convictions
Michigan	Mich. Comp. Laws §750.250m (2006)	All felony and sex crime misdemeanor convictions; violent felony arrests
Minnesota	MN Stat § 299C.105 (2016)	All felony and sex crime misdemeanor convictions; specific felony/misdemeanor arrests
Missouri	MO Rev. Stat § 650.055 (2011)	Felony convictions for offenses against the person, burglary, and sexual offenses; sex crime misdemeanor convictions; arrests for similar offenses.
Georgia	GA Code §35-3-160 (2014)	All felony convictions
Maryland	MD Public Safety Code Ann. §2-504 (2009)	All felony and sex crime misdemeanor convictions; 4th degree burglary; vehicle breaking/entering; related arrests
Mississippi	MS Code § 47-5-183 (2013)	All felony convictions
Montana	MT Code § 44-6-103 (2017)	All felony convictions.
Nebraska	NE Code § 29-4106 (2017)	All felony and certain misdemeanor convictions.
Nevada	NV Rev Stat § 176.09123 (2013)	All felony and certain misdemeanor convictions; all felony arrests.
New Hampshire	NH Rev Stat § 651-C:2 (2015)	All felony and sex crime misdemeanor convictions.
New Jersey	NJ Rev Stat § 53:1-20.20 (2016)	All felony and certain misdemeanor convictions; arrests for enumerated offenses (e.g., murder, manslaughter, sexual offenses).
New Mexico	NM Stat § 29-16-6 (1996) for convictions; § 29-3-10 (2016) for arrests.	All felony and sex crime misdemeanor convictions; all felony arrests.
New York	NY Exec L § 995-C (2014)	All felony and misdemeanor convictions.
North Carolina	NC Gen Stat § 15A-266.4 (2013)	All felony and sex crime misdemeanor convictions.
North Dakota	ND Century Code § 31-13-03 (2013)	All felony and sex crime misdemeanor convictions; all felony arrests.
Ohio	Ohio Rev Code § 2901.07 (2015)	All felony and sex crime misdemeanor convictions; all felony arrests.
Oklahoma	74 OK Stat § 74-150.27a (2015)	All felony and sex crime misdemeanor convictions; other specified misdemeanors; arrests of undocumented aliens.
Oregon	OR Rev Stat § 137.076 (2011)	All felony and sex crime misdemeanor convictions.
Pennsylvania	44 PA Cons Stat § 2316 (2017)	All felony and sex crime misdemeanor convictions.
Rhode Island	RI Gen L § 12-1.5-8 (2014)	All felony and sex crime misdemeanor convictions.
South Carolina	SC Code § 23-3-620 (2012)	All felony and sex crime misdemeanor convictions; certain arrests (expunged automatically).
South Dakota	SD Codified L § 23-5A-6 (2013) for convictions; § 23-5A-5.2 (2013) for arrests.	All felony and sex crime misdemeanor convictions; arrests for felonies, crimes of violence, and sex offenses.
Tennessee	TN Code § 40-35-321 (2010)	All felony and sex crime misdemeanor convictions; all felony arrests.
Texas	TX Gov't Code § 411.1471	All felony and sex crime misdemeanor convictions; arrests if arrestee had a prior conviction (expunged automatically).
Utah	UT Code § 53-10-404 (2006)	All felony, sex crime misdemeanor, and class A misdemeanor convictions; most felony arrests.

state	legal_framework	collection_triggers
Vermont	20 V.S.A. § 1932 (2017)	All felony and sex crime misdemeanor convictions; felony arrests (expunged automatically).
Virginia	VA Code § 19.2-310.2 (2006)	All felony and sex crime misdemeanor convictions; felony arrests (expunged automatically).
Washington	WA Rev Code § 43.43.754 (2005)	All felony and sex crime misdemeanor convictions.
West Virginia	WV Code § 15-2B-6 (2012)	All felony and sex crime misdemeanor convictions.
Wisconsin	WI Code § 973.047 (2011)	All felony and crimes of sexual contact convictions; arrests (expunged automatically).
Wyoming	WY Stat § 7-19-403 (1997)	All felony convictions.
Alabama	AL Code § 36-18-25 (2013)	All felony and sex crime misdemeanor convictions; arrests for all felonies and sexual offenses

5 Combining Information into Master Dataset

The final dataset combines quantitative DNA collection metrics with qualitative methodology to create a comprehensive one-row-per-state resource. This integration allows researchers to understand both the magnitude of DNA collection and the quality/limitations of the underlying data sources.

Integration Process:

1. **Primary Dataset**: The `summary_data` table contains all quantitative metrics:
    - DNA collection counts and percentages by race
    - State population demographics by race
    - Collection rates per 100,000 population by race
2. **Study Methodology Addition**: The `methodology_summary` table provides contextual information:
    - Legal framework for DNA collection
    - Specific collection triggers
    - Data source types and limitations
    - Source URLs for verification
3. **Joining Strategy**:
    - States are matched using full state names
    - A crosswalk table converts two-letter abbreviations to full names
    - Left join ensures all states from the summary data are retained
4. **Column Selection**: The final dataset includes:
    - **State identifier**: Full state name
    - **Collection metrics**: All DNA collection counts, percentages, and rates by race
    - **Methodology context**: Legal framework, collection triggers, data limitations

- **Quality flags**: Categorized data limitation types for filtering/analysis

5. **Output Format**: One row per state with 30+ columns covering demographics, collection rates, and methodology

This unified structure enables analyses that account for data quality differences across states when interpreting racial disparities in DNA collection.

```
# Create state name crosswalk for joining
state_crosswalk <- tibble(
  State = c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA",
            "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
            "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ",
            "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
            "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY", "DC"),
  state_full = c("Alabama", "Alaska", "Arizona", "Arkansas", "California", "Colorado",
                 "Connecticut", "Delaware", "Florida", "Georgia", "Hawaii", "Idaho",
                 "Illinois", "Indiana", "Iowa", "Kansas", "Kentucky", "Louisiana",
                 "Maine", "Maryland", "Massachusetts", "Michigan", "Minnesota",
                 "Mississippi", "Missouri", "Montana", "Nebraska", "Nevada",
                 "New Hampshire", "New Jersey", "New Mexico", "New York", "North Carolina",
                 "North Dakota", "Ohio", "Oklahoma", "Oregon", "Pennsylvania",
                 "Rhode Island", "South Carolina", "South Dakota", "Tennessee", "Texas",
                 "Utah", "Vermont", "Virginia", "Washington", "West Virginia",
                 "Wisconsin", "Wyoming", "District of Columbia")
)

# Join summary data with methodology
annual_dna_combined <- summary_data %>%
  left_join(state_crosswalk, by = "State") %>%
  left_join(methodology_summary, by = c("state_full" = "state")) %>%
  select(
    # State identifier
    state = state_full,
    state_abbrev = State,

    # DNA Collection metrics
    Black_DNA_Pct, Black_DNA_N,
    Hispanic_DNA_Pct, Hispanic_DNA_N,
    Asian_DNA_Pct, Asian_DNA_N,
    Native_American_DNA_Pct, Native_American_DNA_N,
    White_DNA_Pct, White_DNA_N,

    # Population demographics
    Black_Pop_Pct, Hispanic_Pop_Pct, Asian_Pop_Pct,
    Native_American_Pop_Pct, White_Pop_Pct,

    # Collection rates
```

```
    Black_Collection_Rate, Hispanic_Collection_Rate,
    Asian_Collection_Rate, Native_American_Collection_Rate,
    White_Collection_Rate,

    # Methodology
    legal_framework,
    collection_triggers,
    collection_trigger_category,
    data_limitations,
    data_limitation_category,
    source_url
  )

# Calculate total DNA profiles
annual_dna_combined <- annual_dna_combined %>%
  mutate(across(c(Black_DNA_N, Hispanic_DNA_N, Asian_DNA_N, Native_American_DNA_N, White_DNA_N)
  Total_DNA_Profiles = Black_DNA_N + Hispanic_DNA_N + Asian_DNA_N +
                       Native_American_DNA_N + White_DNA_N
  ) %>%
  relocate(Total_DNA_Profiles, .after = White_DNA_N)

# Show preview
kable(annual_dna_combined, booktabs = TRUE, longtable = TRUE, caption = "Complete Annual DNA Co
  kable_styling(
    latex_options = c("striped", "scale_down", "repeat_header"),
    font_size = 8,
    position = "center"
  )
```

state	state_abbrev	Black_DNA_Pct	Black_DNA_N	Hispanic_DNA_Pct	Hispanic_DNA_N	Asian_DNA_Pct	A
Alabama	AL	46.0	18253	NA	0	NA	
Alaska	AK	7.6	914	2.8	23604	3.00	
Arizona	AZ	12.6	9313	32.0	23604	0.50	
Arkansas	AR	42.3	4401	2.7	275	0.04	
California	CA	25.0	141488	37.6	213361	NA	
Colorado	CO	12.8	28486	24.0	53778	0.03	
Connecticut	CT	25.5	2250	18.1	1591	NA	
Delaware	DE	79.9	9399	5.6	660	NA	
Florida	FL	34.8	280152	0.5	4338	0.60	
Georgia	GA	54.5	9851	2.4	433	0.40	
Hawaii	HI	4.6	516	4.6	516	21.60	
Idaho	ID	3.0	244	16.2	1300	NA	
Illinois	IL	58.2	18094	12.6	3906	NA	
Indiana	IN	24.8	3591	2.8	398	NA	
Iowa	IA	20.8	4793	3.3	751	0.90	
Kansas	KS	26.8	13016	NA	0	0.10	
Kentucky	KY	29.3	34762	5.1	5993	NA	
Louisiana	LA	65.8	10840	4.0	657	NA	

state	state_abbrev	Black_DNA_Pct	Black_DNA_N	Hispanic_DNA_Pct	Hispanic_DNA_N	Asian_DNA_Pct	A
Maine	ME	7.1	226	2.0	65	NA	
Maryland	MD	62.8	34835	NA	0	0.70	
Massachusetts	MA	18.5	2712	15.6	2288	NA	
Michigan	MI	33.4	85888	0.1	357	0.40	
Minnesota	MN	25.6	39743	NA	0	2.70	
Mississippi	MS	59.6	6328	0.9	91	NA	
Missouri	MO	31.4	3701	1.1	130	0.40	
Montana	MT	1.9	47	NA	0	1.00	
Nebraska	NE	23.1	566	11.4	279	0.50	
Nevada	NV	28.5	1602	21.3	1201	2.40	
New Hampshire	NH	6.4	40	NA	0	0.50	
New Jersey	NJ	52.8	9444	8.4	1499	1.30	
New Mexico	NM	6.0	157	52.0	1363	NA	
New York	NY	51.8	68684	31.6	41970	3.70	
North Carolina	NC	52.3	12270	NA	0	0.05	
North Dakota	ND	9.2	2983	0.3	92	0.60	
Ohio	OH	36.2	26015	0.6	402	0.03	
Oklahoma	OK	26.7	2841	7.7	815	NA	
Oregon	OR	10.2	515	16.3	824	NA	
Pennsylvania	PA	40.6	4159	9.7	991	NA	
Rhode Island	RI	25.4	3672	20.3	2941	NA	
South Carolina	SC	54.0	3752	NA	0	NA	
South Dakota	SD	7.3	2961	NA	0	0.70	
Tennessee	TN	35.9	14544	0.5	213	0.30	
Texas	TX	31.2	22169	32.8	23267	NA	
Utah	UT	5.4	5993	0.1	141	2.40	
Vermont	VT	10.7	295	NA	0	0.90	
Virginia	VA	50.5	6249	2.4	294	NA	
Washington	WA	18.6	1376	14.4	1070	NA	
West Virginia	WV	28.0	955	6.0	205	NA	
Wisconsin	WI	37.3	3482	NA	0	1.20	
Wyoming	WY	3.9	129	14.4	484	NA	

6 Data Export and Versioning

```
# Create output directory structure
intermediate_dir <- here("data", "annual_dna_collection", "intermediate")
dir.create(intermediate_dir, recursive = TRUE, showWarnings = FALSE)

final_dir <- here("data", "annual_dna_collection", "final")
dir.create(final_dir, recursive = TRUE, showWarnings = FALSE)

frozen_dir <- here("data", "v1.0")
dir.create(frozen_dir, recursive = TRUE, showWarnings = FALSE)

# Export final combined dataset
output_path <- file.path(final_dir, "Annual_DNA_Collection.csv")
write_csv(annual_dna_combined, output_path)
```

```r
cat(paste("  Exported Annual DNA Collection dataset to:", output_path, "\n"))

# Create frozen version (v1.0) for long-term reference
frozen_path <- file.path(frozen_dir, "Annual_DNA_Collection.csv")
write_csv(annual_dna_combined, frozen_path)
cat(paste("  Created frozen version 1.0 at:", frozen_path, "\n\n"))

# Export the methodology summary separately for reference
methodology_output_path <- file.path(intermediate_dir, "State_Methodology.csv")
write_csv(methodology_summary, methodology_output_path)
cat(paste("  Exported study methodology summary to:", methodology_output_path, "\n"))
```

```
  Exported Annual DNA Collection dataset to: C:/Users/Donadio/Documents/PODFRIDGE_Databases/dat
  Created frozen version 1.0 at: C:/Users/Donadio/Documents/PODFRIDGE_Databases/data/v1.0/Annua

  Exported study methodology summary to: C:/Users/Donadio/Documents/PODFRIDGE_Databases/data/an
```

7 Summary and Key Findings

7.1 Dataset Completeness

```r
# Analyze data completeness by category
completeness_summary <- annual_dna_combined %>%
  summarise(
    States_with_Black_Data = sum(!is.na(Black_Collection_Rate)),
    States_with_Hispanic_Data = sum(!is.na(Hispanic_Collection_Rate)),
    States_with_Asian_Data = sum(!is.na(Asian_Collection_Rate)),
    States_with_Native_American_Data = sum(!is.na(Native_American_Collection_Rate)),
    States_with_White_Data = sum(!is.na(White_Collection_Rate)),
    States_with_Legal_Framework = sum(!is.na(legal_framework))
  )

kable(t(completeness_summary), col.names = "Count",
      caption = "Data Completeness Across States") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Table 4: Data Completeness Across States

	Count
States_with_Black_Data	50
States_with_Hispanic_Data	39
States_with_Asian_Data	26
States_with_Native_American_Data	34

States_with_White_Data	50
States_with_Legal_Framework	1