Annual DNA collection from Murphy & Tong Study

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.

Column	Type	Rows	Missing	Unique	Example_Value
State	character	50	0	50	AL
Black_DNA_Pct	numeric	50	0	50	46
Black_DNA_N	numeric	50	0	50	18253
Hispanic_DNA_Pct	numeric	50	11	35	2.8
Hispanic_DNA_N	numeric	50	11	39	23604
Asian_DNA_Pct	numeric	50	23	19	3

Column	Type	Rows	Missing	Unique	Example_Value
Asian_DNA_N	numeric	50	23	26	365
Native_American_DNA_Pct	numeric	50	15	28	43.1
Native_American_DNA_N	numeric	50	15	36	5191
White_DNA_Pct	numeric	50	0	47	54
White_DNA_N	numeric	50	0	50	21292
Black_Pop_Pct	numeric	50	0	47	26.8
Hispanic_Pop_Pct	numeric	50	0	43	4.2
Asian_Pop_Pct	numeric	50	0	38	1.4
Native_American_Pop_Pct	numeric	50	1	27	0.7
White_Pop_Pct	numeric	50	0	47	66
Black_Collection_Rate	numeric	50	0	31	1.4
Hispanic_Collection_Rate	numeric	50	11	15	0.7
Asian_Collection_Rate	numeric	50	24	14	0.8
Native_American_Collection_Rate	numeric	50	16	21	4.7
White_Collection_Rate	numeric	50	0	19	0.7

Data frame dimensions: 50 rows x 21 columns

3.2 Disparity Analysis

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.txt`) 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

Table 1: Racial Breakdown of Annual DNA Collection for Each State

```
# 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
```

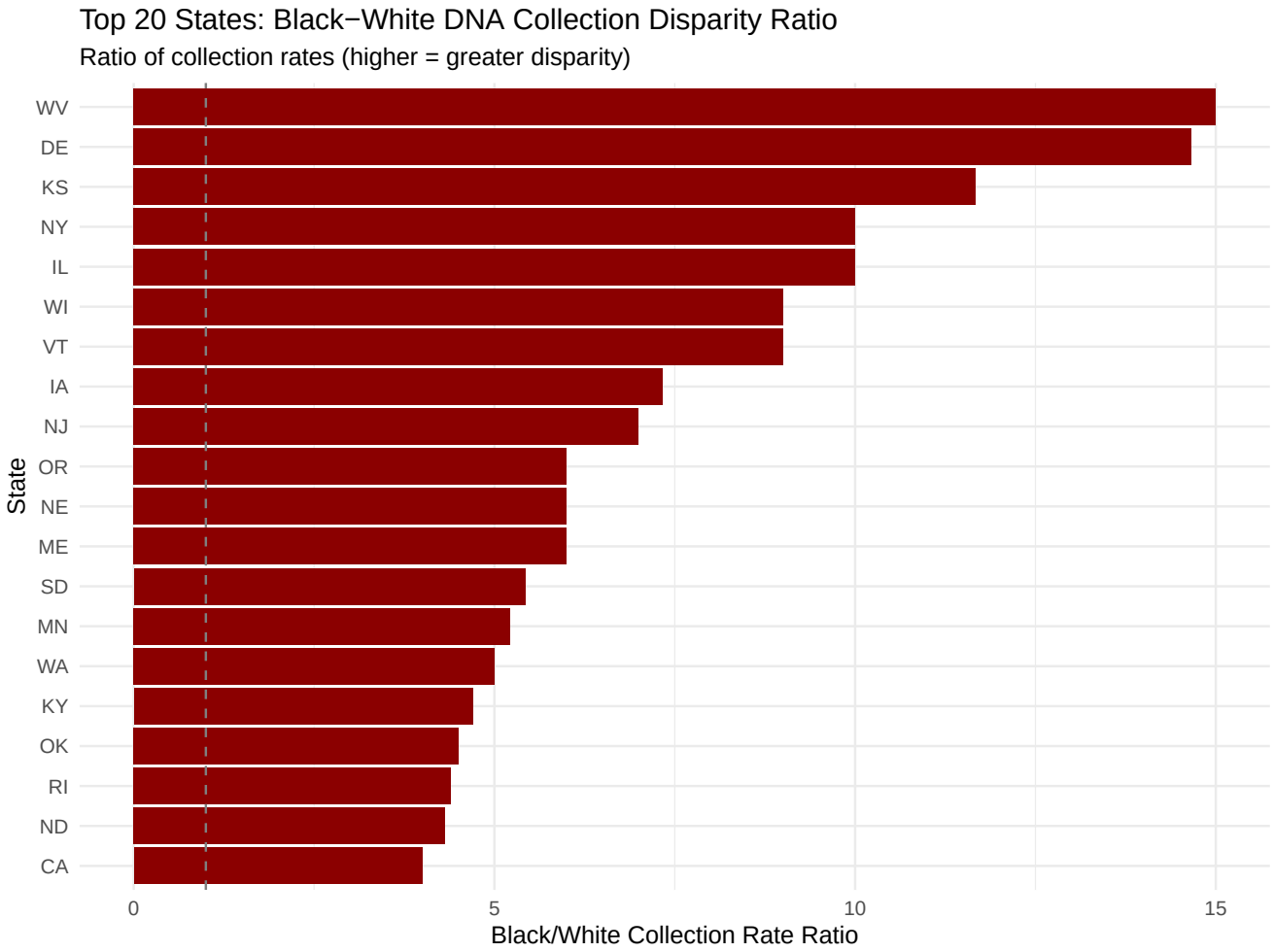

Figure 1: DNA Collection Rates by Race Relative to Population Percentage

- **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", "Connecticut
               "Delaware", "Florida", "Georgia", "Hawaii", "Idaho", "Illinois", "Indiana", "Iowa"
               "Kansas", "Kentucky", "Louisiana", "Maine", "Maryland", "Massachusetts", "Michigan
               "Minnesota", "Mississippi", "Missouri", "Montana", "Nebraska", "Nevada", "New Hamp
               "New Jersey", "New Mexico", "New York", "North Carolina", "North Dakota", "Ohio",
               "Oklahoma", "Oregon", "Pennsylvania", "Rhode Island", "South Carolina", "South Dak
               "Tennessee", "Texas", "Utah", "Vermont", "Virginia", "Washington", "West Virginia"
               "Wisconsin", "Wyoming")

# Read the text file
data_file <- file.path(here("data", "annual_dna_collection", "intermediate", "MurphyTong_States_P

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 Limitati
  data_source_df <- tibble(data_source_type = NA_character_, data_source_note = NA_character_)

  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|Ari
data_limitations <- NA_character_

if (!is.na(data_limitations_text)) {
  data_limitations_text <- str_remove(data_limitations_text, "Data Limitations:") %>% str_trim(
  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|Ha

  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_charac
      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") ~ "Co
      str_detect(collection_triggers, "(?i)all felony.*convictions.*arrests.*certain|specific") ~
      str_detect(collection_triggers, "(?i)all felony.*convictions.*arrests") ~ "Broad: All felon
      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 ca
      str_detect(data_limitations, "(?i)no.*data.*available|unavailable") ~ "Various data unavail
      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 Dat
      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
enhanced_glimpse(methodology_summary)
```

Column	Type	Rows	Missing	Unique	Example_Value
state	character	50	0	50	Alaska
legal_framework	character	50	0	50	AK Stat § 44.41.035 (2014)
collection_triggers	character	50	0	32	All felony and sex crime misdemeanor conviction...
collection_trigger_category	character	50	0	5	Comprehensive: All felonies + broad arrests
data_limitations	character	50	0	45	No specific conviction data; Racial makeup esti...
data_limitation_category	character	50	0	5	Other limitations
source_url	character	50	8	43	https://www.bjs.gov/content/pub/pdf/p14.pdf; ht...

Column	Type	Rows	Missing	Unique	Example_Value

Data frame dimensions: 50 rows x 7 columns

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.

Column	Type	Rows	Missing	Unique	Example_Value
state	character	50	0	50	Alabama
state_abbrev	character	50	0	50	AL

Column	Type	Rows	Missing	Unique	Example_Value
Black_DNA_Pct	numeric	50	0	50	46
Black_DNA_N	numeric	50	0	50	18253
Hispanic_DNA_Pct	numeric	50	11	35	2.8
Hispanic_DNA_N	numeric	50	0	39	0
Asian_DNA_Pct	numeric	50	23	19	3
Asian_DNA_N	numeric	50	0	26	0
Native_American_DNA_Pct	numeric	50	15	28	43.1
Native_American_DNA_N	numeric	50	0	36	0
White_DNA_Pct	numeric	50	0	47	54
White_DNA_N	numeric	50	0	50	21292
Total_DNA_Profiles	numeric	50	0	50	39545
Black_Pop_Pct	numeric	50	0	47	26.8
Hispanic_Pop_Pct	numeric	50	0	43	4.2
Asian_Pop_Pct	numeric	50	0	38	1.4
Native_American_Pop_Pct	numeric	50	1	27	0.7
White_Pop_Pct	numeric	50	0	47	66
Black_Collection_Rate	numeric	50	0	31	1.4
Hispanic_Collection_Rate	numeric	50	11	15	0.7
Asian_Collection_Rate	numeric	50	24	14	0.8
Native_American_Collection_Rate	numeric	50	16	21	4.7
White_Collection_Rate	numeric	50	0	19	0.7
legal_framework	character	50	49	2	AL Code § 36-18-25 (2013)
collection_triggers	character	50	49	2	All felony and sex crime misdemeanor conviction...
collection_trigger_category	character	50	49	2	Comprehensive: All felonies + broad arrests
data_limitations	character	50	49	2	No direct felony conviction data available; No ...
data_limitation_category	character	50	49	2	Missing conviction data
source_url	character	50	49	2	http://www.doc.state.al.us/docs/AnnualRpts/2014...

Data frame dimensions: 50 rows x 29 columns

6 Summary and Key Findings

6.1 Dataset Completeness

Table 5: 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