groupedHG: Hypergeometric & Binomial Group Sampling

Sumonkanti Das

2025-06-26

Contents

```r
library(groupedHG)
library(ggplot2)
library(ggpubr)
```

1 Introduction

This document introduces the **groupedHG** R package, which provides tools for designing and analyzing group (pooled) sampling under imperfect testing conditions. Below, we outline the motivation, applications, and key features of the package.

- **Problem statement:** Group (or pooled) sampling is a widely used technique in biological sciences and public health for efficiently detecting pest species, pathogens, or estimating disease prevalence in populations. However, most real-world testing processes—such as PCR or serological tests—are imperfect, exhibiting less-than-ideal sensitivity and specificity. These imperfections can significantly bias prevalence estimates and lead to suboptimal decision-making, especially in critical contexts such as disease outbreak response or biosecurity monitoring.

The `groupedHG` R package addresses this challenge by providing tools for designing and analyzing grouped-hypergeometric sampling strategies that explicitly incorporate test sensitivity and specificity. It supports a variety of sampling models, including those that respond to the presence of at least one contaminated item, or that scale with the number of contaminated items in a group. The package also includes methods based on Hellinger information, enabling users to design group-sampling strategies that maximize the accuracy of prevalence inference under imperfect test performance.

By formalizing these statistical foundations in a flexible and accessible framework, groupedHG supports researchers and public health practitioners in developing robust and efficient sampling plans tailored to real-world testing limitations. - **Applications:** The `groupedHG` R package supports robust group sampling designs across biosecurity, public health, and ecology by accounting for imperfect test sensitivity and specificity. It enables accurate prevalence estimation and detection from pooled samples, improving decision-making in surveillance, outbreak response, and ecological monitoring. - **Key features:** The `groupedHG` package provides exact hypergeometric formulas, binomial approximations, and Hellinger information-based tools to support accurate inference and optimal design of group sampling strategies under imperfect testing conditions.

2 Installation

You can install the development version of `groupedHG` as below.

```r
install.packages("devtools")
devtools::install_github("sumon148/groupedHG")
```

3 Mathematical Background

The `groupedHG` package implements statistical models for analyzing group (pooled) sampling data under both perfect and imperfect testing conditions. It is designed to support the inference of contamination prevalence and optimize sampling strategies in contexts such as biosecurity, public health, and ecology. The package provides exact formulas based on the hypergeometric distribution, as well as computationally efficient binomial approximations. It also accounts for test imperfections at both the group level (using parameters Δ and Λ) and the item level (using δ and λ). Additionally, the package offers tools for computing Hellinger information, which supports the design of statistically efficient group testing schemes.

3.1 Notation

- N: population size

- T_X: number of contaminated items

- b: number of pools

- $\bar{N}$: pool size
- t_y: number of positive pools

3.2 Core Formulas

- `pmf_hg_perfect`: Hypergeometric PMF under perfect testing
- `pmf_hg_group_imperfect`: Hypergeometric PMF with group-level imperfections (Δ/Λ)

- `pmf_hg_item_imperfect`: Hypergeometric PMF with item-level imperfections (δ/λ)

- `pmf_bn_perfect`: Binomial approximation under perfect testing

- `pmf_bn_group_imperfect`: Binomial approximation with group-level imperfections (Δ/Λ)

- `pmf_bn_item_imperfect`: Binomial approximation with item-level imperfections (δ/λ)

- `info_hg_tx_imperfect`: Hellinger information for hypergeometric model with imperfections

- `info_bn_tx_imperfect`: Hellinger information for binomial approximation with imperfections

3.3 Correspondence to Equations in Barnes et al. (2025) paper

The following equations from the Barnes et al. (2025) form the statistical foundation of the `groupedHG` package:

- **Equation (4):** Hypergeometric PMF under perfect testing

- **Equation (6):** Hypergeometric PMF with group-level imperfections

- **Equation (8):** Hypergeometric PMF with item-level imperfections

- **Equation (15):** Approximate binomial PMF under perfect testing

- **Equation (16):** Binomial approximation with group-level imperfections

- **Equation (17):** Binomial approximation with item-level imperfections

- **Equation (20):** Hellinger information computed from the PMF with respect to the considered distribution

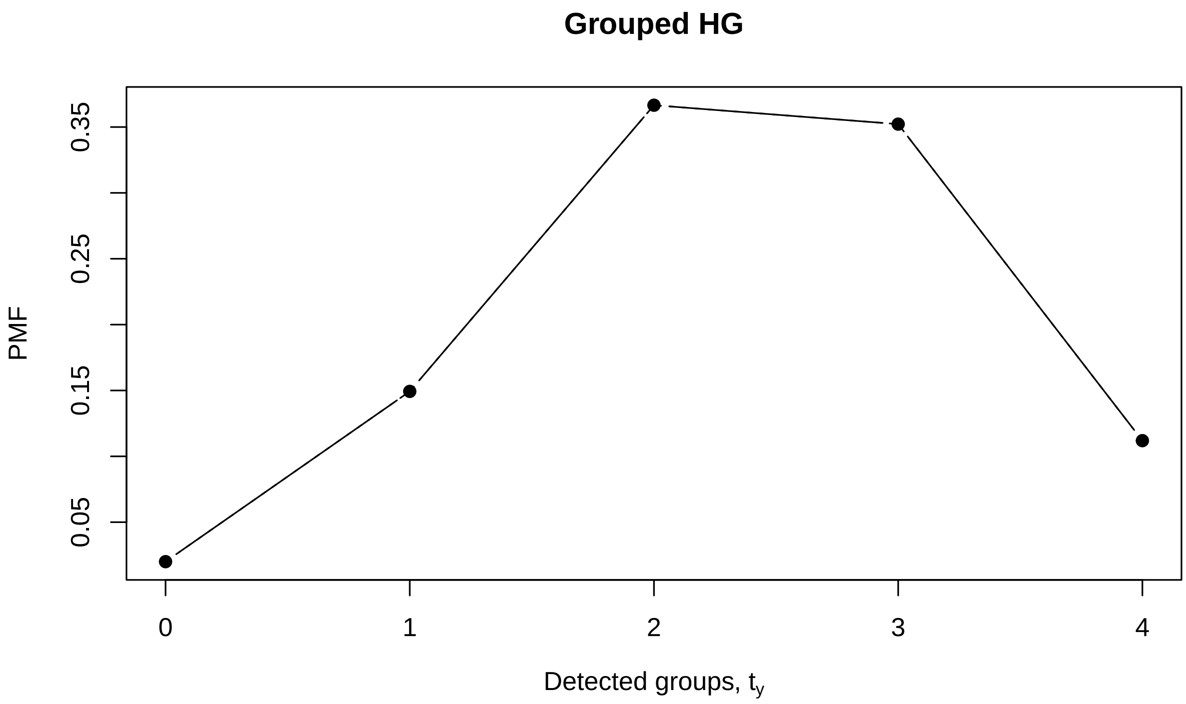
4 Basic Usage of HG modeling

4.1 PMF under perfect tests

Consider a study population of $N = 100$ items, among which $T_X = 20$ are contaminated. These items are divided into $b = 4$ groups of equal size. Our goal is to estimate the probability mass function (PMF) for the possible number of detected positive groups, denoted by t_y, under the assumption of perfect test sensitivity and specificity. Using the `pmf_hg_perfect()` function, we calculate the PMF values for all possible t_y from 0 to 4. This provides the probabilities of observing each count of detected groups, which can then be used to understand the distribution of group-level test outcomes in the population.

```
b=4
ty.values <- c(0:b)
PRty.HG.perfect <- sapply(ty.values,
function(ty) pmf_hg_perfect(
ty, N=100, barN=4, Tx=20, b=b))
PRty.HG.perfect
#> [1] 0.02003047 0.14929714 0.36656593 0.35220780 0.11189866
```

```
plot(0:b, PRty.HG.perfect, type = "l", col = "black", xlab=expression(paste("Detected groups, ", t[y]))
     ylab = "PMF", main = "Grouped HG",lty=2)
points(0:b,PRty.HG.perfect,type = "b", pch = 19, col = "black")
```



Next, we extend the analysis by examining the PMF functions that account for imperfect testing at both the group and item levels. By applying the pmf_hg_group_imperfect() and pmf_hg_item_imperfect() functions, we can evaluate how test sensitivity and specificity affect the distribution of detected positive groups. We begin by calculating the PMF for the number of detected positive groups assuming a perfect test with 100% sensitivity (`delta=1`) and specificity (`lambda=1`). This serves as a baseline for comparison.

```
PRty.HG.group.perfect <- sapply(ty.values,
function(ty) pmf_hg_group_imperfect(
ty, N=100, barN=4, Tx=20, b=b,
delta=1, lambda=1,
verbose = FALSE))
```

```
PRty.HG.group.perfect
#> [1] 0.02003047 0.14929714 0.36656593 0.35220780 0.11189866
```

In similar way, the function for item-level imperfect test provides the same results.

```
PRty.HG.item.perfect <- sapply(ty.values,
function(ty) pmf_hg_item_imperfect(
ty, N=100, barN=4, Tx=20, b=b,
delta=1, lambda=1,
verbose = FALSE))
PRty.HG.item.perfect
#> [1] 0.02003047 0.14929714 0.36656593 0.35220780 0.11189866
```

4.2 PMF under imperfect tests

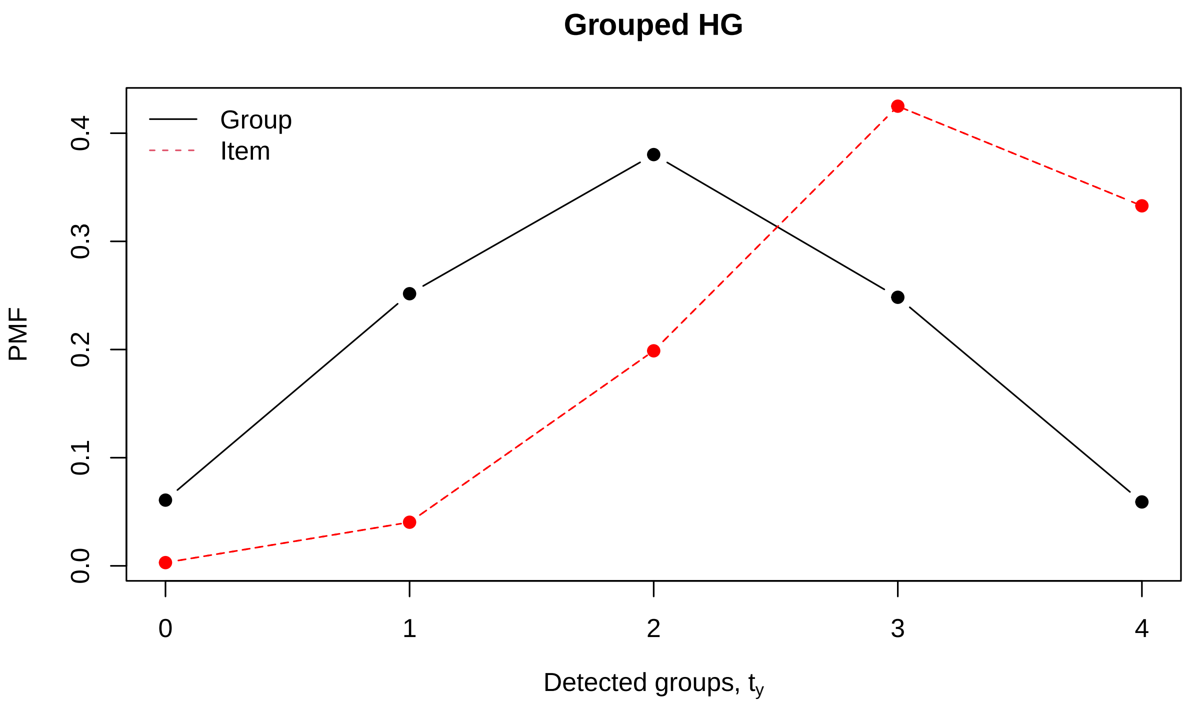
To investigate the impact of imperfect testing, we compute the PMF using both group-level and item-level sensitivity and specificity. In both cases, we assume a sensitivity of $\Delta = \delta = 0.7$ and a specificity of $\Lambda = \lambda = 0.8$. Upper and lower case Greek letters refer to respectively group and item level test accuracy. The function `pmf_hg_group_imperfect()` models the scenario where the test responds to the presence or absence of any contaminated item in a group (group-level accuracy), while `pmf_hg_item_imperfect()` accounts for the influence of each individual item's contamination status (item-level accuracy). As a result, the PMF distributions produced by these functions differ: group-level imperfections tend to **underestimate the number of positive groups** when contamination is sparse, while item-level imperfections may result in **greater uncertainty**, especially when multiple contaminated items exist within a group. Comparing these distributions to the perfect test case reveals how diagnostic accuracy directly influences the detection probabilities and, ultimately, prevalence inference.

```
PRty.HG.group.imperfect <- sapply(ty.values,
function(ty) pmf_hg_group_imperfect(
ty, N=100, barN=4, Tx=20, b=b,
delta=0.70, lambda=0.80,
verbose = FALSE))
PRty.HG.group.imperfect
#> [1] 0.06076479 0.25161691 0.38022944 0.24830774 0.05908113
```

```
PRty.HG.item.imperfect <- sapply(ty.values,
function(ty) pmf_hg_item_imperfect(
ty, N=100, barN=4, Tx=20, b=b,
delta=0.70, lambda=0.80,
verbose = FALSE))
PRty.HG.item.imperfect
#> [1] 0.002993711 0.040346066 0.198811210 0.424987223 0.332861789
```

The difference due to the group- and item-level test accuracy is protrain in the below figure.

```
plot(0:b, PRty.HG.item.imperfect, type = "l", col = "red", xlab=expression(paste("Detected groups, ", t
     ylab = "PMF", main = "Grouped HG",lty=2)
points(0:b,PRty.HG.group.imperfect,type = "b", pch = 19, col = "black")
points(0:b,PRty.HG.item.imperfect,type = "b",lty=2, pch = 19, col = "red")
legend("topleft",legend=c("Group","Item"),col=c(1,2),lty=c(1,2),bty = "n")
```



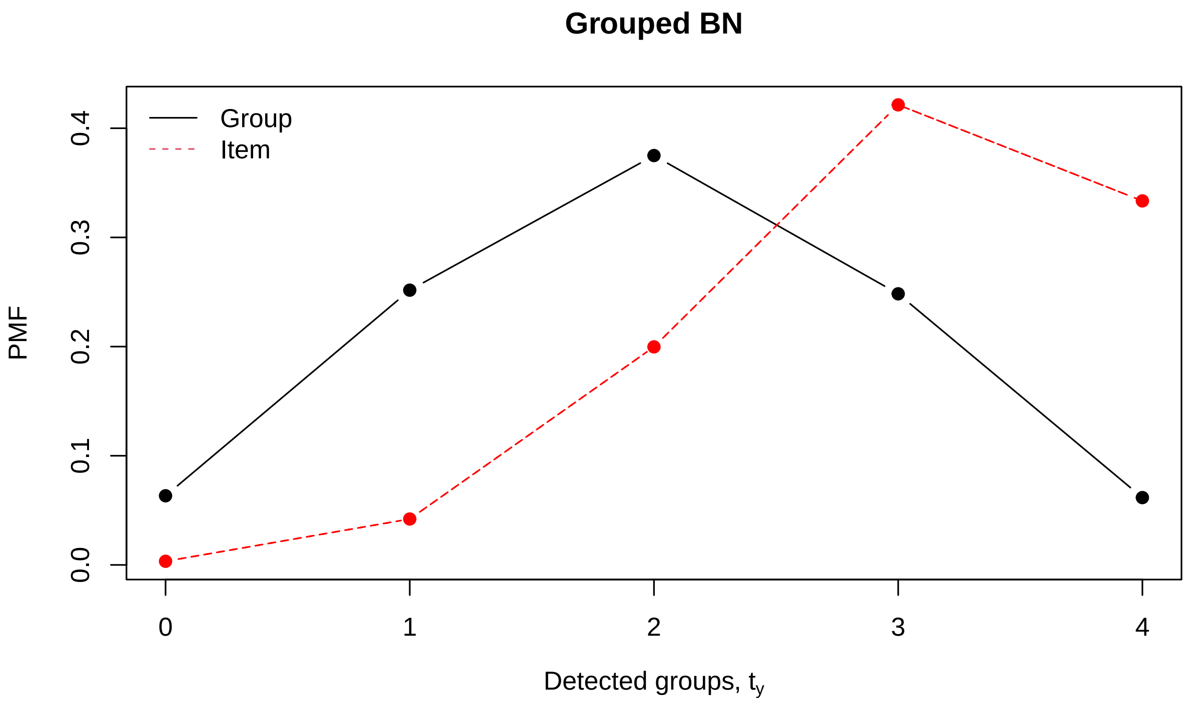
5 Binomial Approximations of HG model

Now we will use binomial approximation of HG model to estimate the probability of having positive groups under perfect and imperfect test. Let use the R functions developed considering test accuracy to estimate the probability of having positive groups $t_y = 0, 1, 2, 3, 4$.

```
PRty.BN.perfect <- sapply(ty.values,
function(ty) pmf_bn_perfect(
ty, N=100, barN=4, Tx=20, b=b))
PRty.BN.group.perfect <- sapply(ty.values,
function(ty) pmf_bn_group_imperfect(
ty, N=100, barN=4, Tx=20, b=b,delta=1,lambda=1))
PRty.BN.item.perfect <- sapply(ty.values,
function(ty) pmf_bn_item_imperfect(
ty, N=100, barN=4, Tx=20, b=b,delta=1,lambda=1))
rbind(ty=c(0:b),PRty.BN.perfect,PRty.BN.group.perfect,PRty.BN.item.perfect)
#>                             [,1]      [,2]      [,3]      [,4]      [,5]
#> ty                    0.00000000 1.0000000 2.0000000 3.0000000 4.0000000
#> PRty.BN.perfect       0.02646535 0.1566017 0.3474933 0.3426999 0.1267397
#> PRty.BN.group.perfect 0.02646535 0.1566017 0.3474933 0.3426999 0.1267397
#> PRty.BN.item.perfect  0.02814750 0.1622879 0.3508842 0.3371778 0.1215026
```

Assuming group level test inaccuracy, we wish to estimate the probability and then compare with those assuming item-level test inaccuracy.

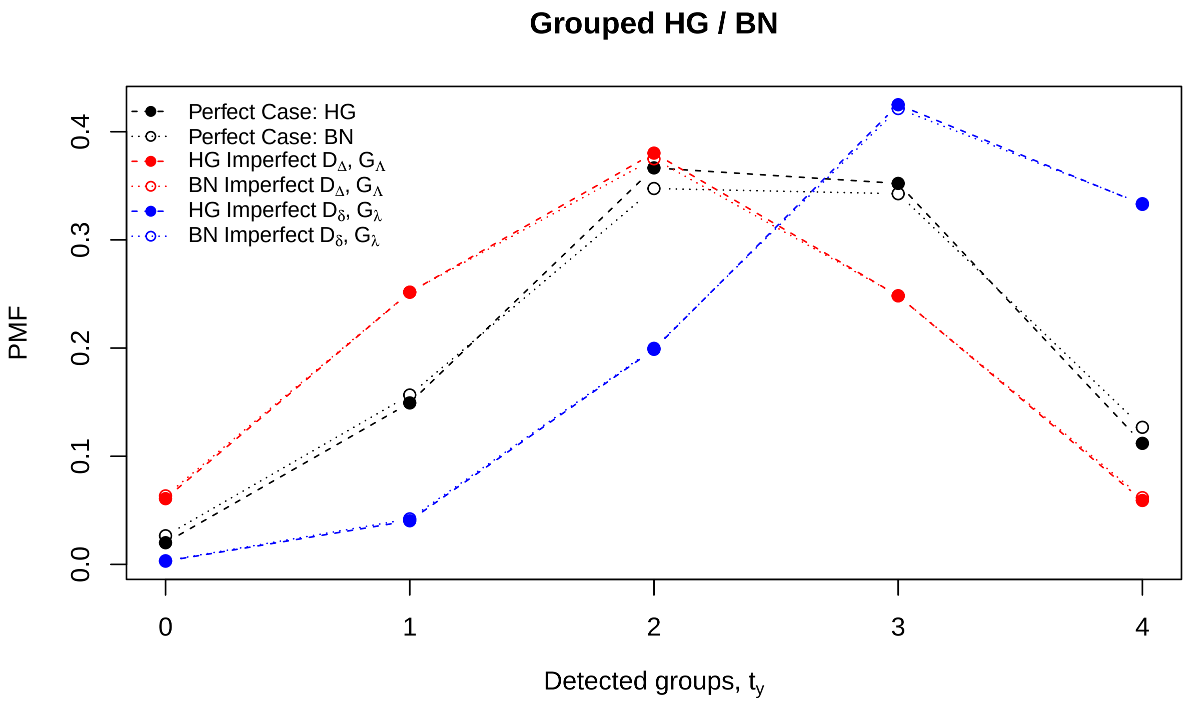
```
PRty.BN.group.imperfect <- sapply(ty.values,
function(ty) pmf_bn_group_imperfect(
ty, N=100, barN=4, Tx=20, b=b,
delta=0.70, lambda=0.80))
PRty.BN.item.imperfect <- sapply(ty.values,
function(ty) pmf_bn_item_imperfect(
ty, N=100, barN=4, Tx=20, b=b,
delta=0.70, lambda=0.80))
plot(0:b, PRty.BN.item.imperfect, type = "l", col = "red", xlab=expression(paste("Detected groups, ", t
     ylab = "PMF", main = "Grouped BN",lty=2)
points(0:b,PRty.BN.group.imperfect,type = "b", pch = 19, col = "black")
points(0:b,PRty.BN.item.imperfect,type = "b",lty=2, pch = 19, col = "red")
legend("topleft",legend=c("Group","Item"),col=c(1,2),lty=c(1,2),bty = "n")
```



The differences in the probability mass function (PMF) between the exact HG model and its BN approximation can be examined below. We plotted the PMFs under both perfect and imperfect testing scenarios for both HG and BN sampling models. In the context of imperfect testing, group-level test characteristics are denoted by D_Δ (sensitivity) and D_Λ (specificity), while item-level characteristics are denoted by D_δ and D_λ, respectively. The plot reveals that the BN approximation closely matches the exact HG model.

```
range.pmf <- range(c(PRty.HG.perfect,PRty.BN.perfect,PRty.HG.group.imperfect,PRty.HG.item.imperfect,PRty
plot(0:b, PRty.HG.perfect, type = "b", lty=2,  pch = 19, col = "black", xlab=expression(paste("Detected
     ylab = "PMF", main = "Grouped HG / BN",ylim=range.pmf)
points(0:b,PRty.BN.perfect,type = "b", lty=3, pch = 21, col = "black")
points(0:b,PRty.HG.group.imperfect,type = "b",lty=2, pch = 19, col = "red")
points(0:b,PRty.BN.group.imperfect,type = "b",lty=3, pch = 21, col = "red")
points(0:b,PRty.HG.item.imperfect,type = "b", lty=2, pch = 19, col = "blue")
points(0:b,PRty.BN.item.imperfect,type = "b",lty=3, pch = 21, col = "blue")
```

```
legend("topleft",
       legend = c(
         "Perfect Case: HG",
         "Perfect Case: BN",
         bquote(HG ~ Imperfect ~ D[Delta] * "," ~ G[Lambda]),
         bquote(BN ~ Imperfect ~ D[Delta] * "," ~ G[Lambda]),
         bquote(HG ~ Imperfect ~ D[delta] * "," ~ G[lambda]),
         bquote(BN ~ Imperfect ~ D[delta] * "," ~ G[lambda])
       ),
       col = c("black", "black", "red", "red", "blue", "blue"),
       lty = c(2, 3, 2, 3, 2, 3),     # match line types for each model
       pch = c(19, 21, 19, 21, 19, 21), # match point types for each model
       bty = "n",
       cex = 0.8)
```



6 Hellinger Information

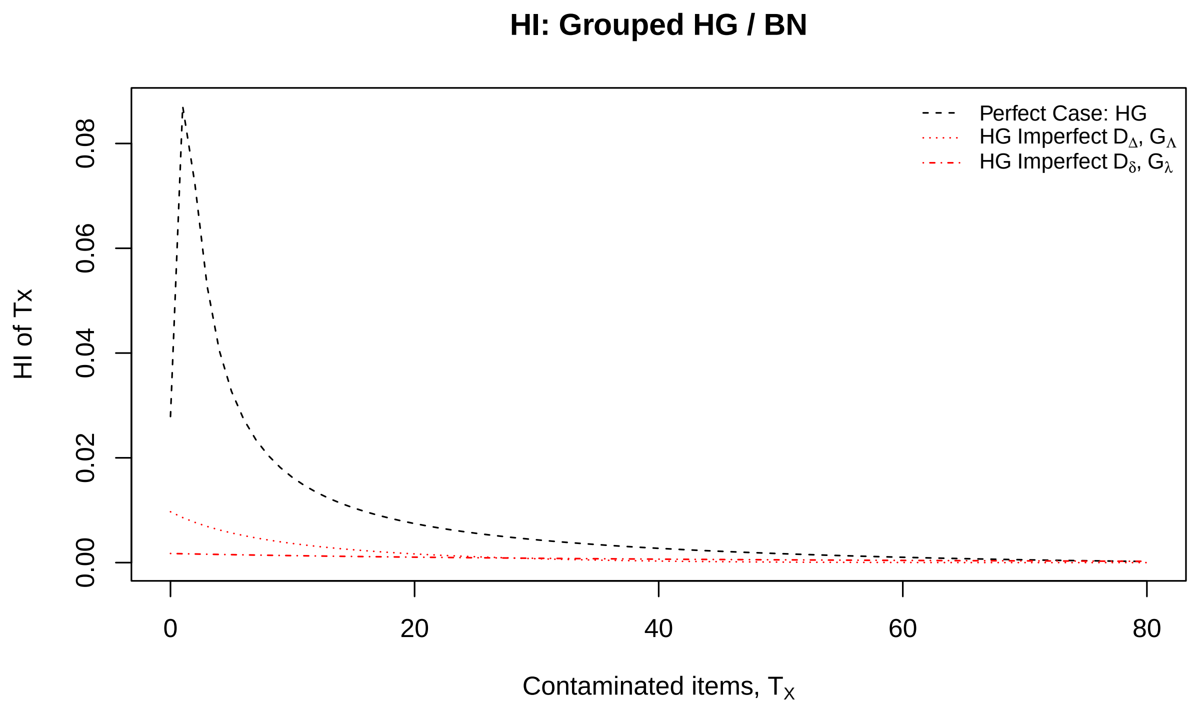
Now we work with Hellinger information, a generalized version of Fisher Information, to evaluate the amount of information available for estimating prevalence given a fixed number of contaminated items (T_X). This metric quantifies the sensitivity of the PMF to changes in T_X, and helps assess the efficiency of different group sampling designs—especially under imperfect testing conditions.

The `info_hg_tx_imperfect()` function computes Hellinger information for the grouped hypergeometric model under imperfect testing conditions. It supports both group- and item-level test characteristics, allowing comparison of different information measures such as analytically driven formula (method="AD"). The below chunk estimates HI for $T_X = 0 : 80$ under HG sampling with perfect and imperfect scenarioes.

```r
Tx.values <- c(0:80)
# Different cases: Item level sensitivity
FI.HG.Perfect <- lapply(
  Tx.values, function(Tx) info_hg_tx_imperfect(
    Tx, N=100, b=4, barN=4, delta=1.0, lambda=1.0,
    method = "PMF-HI", type = "item",verbose = FALSE))
FI.Tx.HG.Perfect <- sapply(FI.HG.Perfect, function(x) x$FI_Tx)
FI.HG.Group.Imperfect <- lapply(
  Tx.values, function(Tx) info_hg_tx_imperfect(
    Tx, N=100, b=4, barN=4, delta=0.7, lambda=0.8,
    method = "PMF-HI", type = "group",verbose = FALSE))
FI.Tx.HG.Group.Imperfect <- sapply(FI.HG.Group.Imperfect, function(x) x$FI_Tx)
FI.HG.Item.Imperfect <- lapply(
  Tx.values, function(Tx) info_hg_tx_imperfect(
    Tx, N=100, b=4, barN=4, delta=0.7, lambda=0.8,
    method = "PMF-HI", type = "item",verbose = FALSE))
FI.Tx.HG.Item.Imperfect <- sapply(FI.HG.Item.Imperfect, function(x) x$FI_Tx)
```

We now compare the Hellinger information across perfect testing, and group- and item-level imperfect testing scenarios, for the full range of contaminated item counts $T_X = 0$ to 80, under the hypergeometric (HG) sampling model.

```r
range.HI <- range(c(FI.Tx.HG.Perfect,FI.Tx.HG.Group.Imperfect,FI.Tx.HG.Item.Imperfect))
plot(Tx.values, FI.Tx.HG.Perfect, type = "l", lty=2,  col = "black", xlab=expression(paste("Contaminate
     ylab = "HI of Tx", main = "HI: Grouped HG / BN",ylim=range.HI)
points(Tx.values,FI.Tx.HG.Group.Imperfect,type = "l",lty=3, col = "red")
points(Tx.values,FI.Tx.HG.Item.Imperfect,type = "l", lty=4, col = "red")
legend("topright",
       legend = c(
         "Perfect Case: HG",
         bquote(HG ~ Imperfect ~ D[Delta] * "," ~ G[Lambda]),
         bquote(HG ~ Imperfect ~ D[delta] * "," ~ G[lambda])
         ),
       col = c("black", "red", "red"),
       lty = c(2, 3, 4),      # match line types for each model
       bty = "n",
       cex = 0.8)
```



In the similar way we can estimate HI for the BN sampling model as below.

```r
Tx.values <- c(0:80)
# Different cases: Item level sensitivity
FI.BN.Perfect <- lapply(
  Tx.values, function(Tx) info_bn_tx_imperfect(
    Tx, N=100, b=4, barN=4, delta=1.0, lambda=1.0,
    method = "PMF-HI", type = "item"))
FI.Tx.BN.Perfect <- unlist(FI.BN.Perfect)
FI.BN.Group.Imperfect <- lapply(
  Tx.values, function(Tx) info_bn_tx_imperfect(
    Tx, N=100, b=4, barN=4, delta=0.7, lambda=0.8,
    method = "PMF-HI", type = "group"))
FI.Tx.BN.Group.Imperfect <- unlist(FI.BN.Group.Imperfect)
FI.BN.Item.Imperfect <- lapply(
  Tx.values, function(Tx) info_bn_tx_imperfect(
    Tx, N=100, b=4, barN=4, delta=0.7, lambda=0.8,
    method = "PMF-HI", type = "item"))
FI.Tx.BN.Item.Imperfect <- unlist(FI.BN.Item.Imperfect)
```

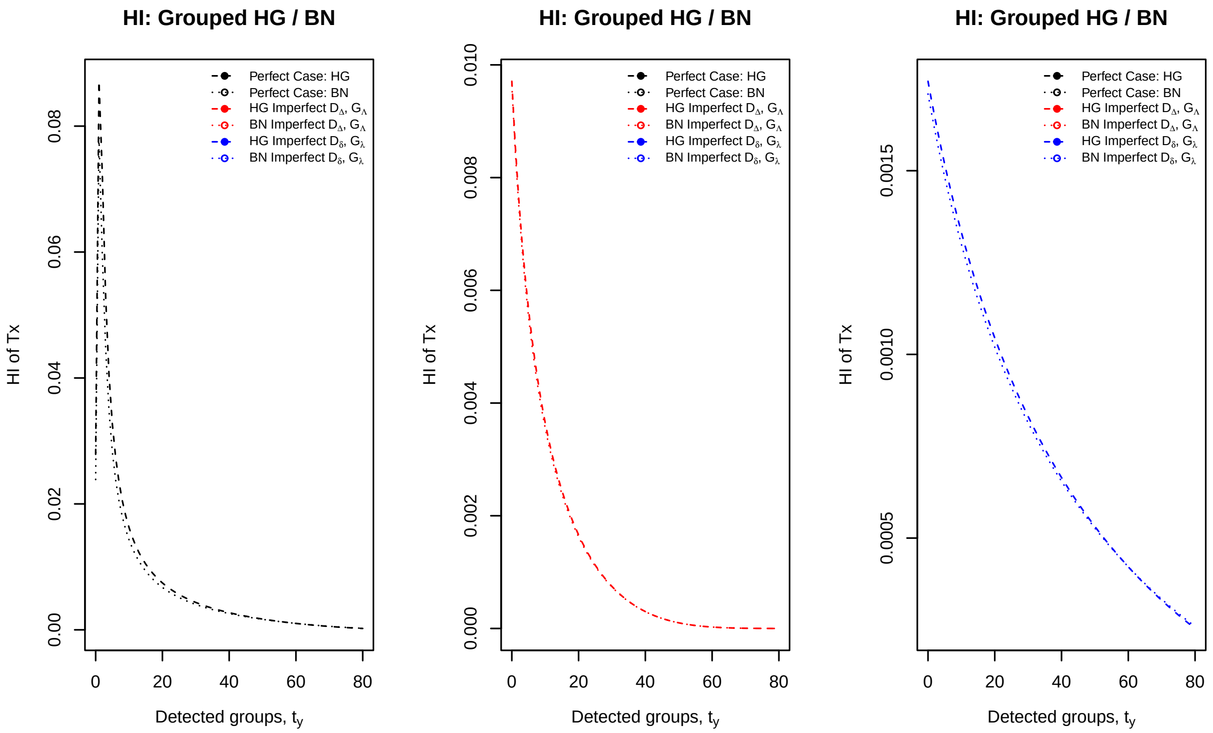
Now we will compare the HI for T_X under HG and BN approximation.

```r
par(mfrow=c(1,3))
Tx.values <- c(0:80)
range.HI <- range(c(FI.Tx.HG.Perfect,FI.Tx.BN.Perfect))
plot(Tx.values, FI.Tx.HG.Perfect, type = "l", lty=2,  pch = 19, col = "black", xlab=expression(paste("D
     ylab = "HI of Tx", main = "HI: Grouped HG / BN",ylim=range.HI)
points(Tx.values,FI.Tx.BN.Perfect,type = "l", lty=3, pch = 21, col = "black")
```

```r
legend("topright",
       legend = c(
         "Perfect Case: HG",
         "Perfect Case: BN",
         bquote(HG ~ Imperfect ~ D[Delta] * "," ~ G[Lambda]),
         bquote(BN ~ Imperfect ~ D[Delta] * "," ~ G[Lambda]),
         bquote(HG ~ Imperfect ~ D[delta] * "," ~ G[lambda]),
         bquote(BN ~ Imperfect ~ D[delta] * "," ~ G[lambda])
       ),
       col = c("black", "black", "red", "red", "blue", "blue"),
       lty = c(2, 3, 2, 3, 2, 3),     # match line types for each model
       pch = c(19, 21, 19, 21, 19, 21), # match point types for each model
       bty = "n",
       cex = 0.8)

Tx.values <- c(0:80)
range.HI <- range(c(FI.Tx.HG.Group.Imperfect,FI.Tx.BN.Group.Imperfect))
plot(Tx.values, FI.Tx.HG.Group.Imperfect, type = "l",lty=2, pch = 19, col = "red", xlab=expression(paste
     ylab = "HI of Tx", main = "HI: Grouped HG / BN",ylim=range.HI)
points(Tx.values,FI.Tx.BN.Group.Imperfect,type = "l",lty=3, pch = 21, col = "red")
legend("topright",
       legend = c(
         "Perfect Case: HG",
         "Perfect Case: BN",
         bquote(HG ~ Imperfect ~ D[Delta] * "," ~ G[Lambda]),
         bquote(BN ~ Imperfect ~ D[Delta] * "," ~ G[Lambda]),
         bquote(HG ~ Imperfect ~ D[delta] * "," ~ G[lambda]),
         bquote(BN ~ Imperfect ~ D[delta] * "," ~ G[lambda])
       ),
       col = c("black", "black", "red", "red", "blue", "blue"),
       lty = c(2, 3, 2, 3, 2, 3),     # match line types for each model
       pch = c(19, 21, 19, 21, 19, 21), # match point types for each model
       bty = "n",
       cex = 0.8)


Tx.values <- c(0:80)
range.HI <- range(c(FI.Tx.HG.Item.Imperfect,FI.Tx.BN.Item.Imperfect))
plot(Tx.values, FI.Tx.HG.Item.Imperfect, type = "l", lty=2, pch = 19, col = "blue", xlab=expression(past
     ylab = "HI of Tx", main = "HI: Grouped HG / BN",ylim=range.HI)
points(Tx.values,FI.Tx.BN.Item.Imperfect,type = "l",lty=3, pch = 21, col = "blue")
legend("topright",
       legend = c(
         "Perfect Case: HG",
         "Perfect Case: BN",
         bquote(HG ~ Imperfect ~ D[Delta] * "," ~ G[Lambda]),
         bquote(BN ~ Imperfect ~ D[Delta] * "," ~ G[Lambda]),
         bquote(HG ~ Imperfect ~ D[delta] * "," ~ G[lambda]),
         bquote(BN ~ Imperfect ~ D[delta] * "," ~ G[lambda])
       ),
       col = c("black", "black", "red", "red", "blue", "blue"),
       lty = c(2, 3, 2, 3, 2, 3),     # match line types for each model
       pch = c(19, 21, 19, 21, 19, 21), # match point types for each model
       bty = "n",
       cex = 0.8)
```



7 Case Study: Salmonella in Pig Pens

7.1 Application to Salmonella Surveillance on Pig Farms

Levels of *Salmonella* bacteria on pig farms are regulated due to the high prevalence of infections and their significant contribution to human salmonellosis cases (Arnold, 2005). Farmed pigs are typically housed in pens of 30–100 individuals, and pooled faecal samples are tested routinely. Group testing is cost-efficient and improves detection because *Salmonella* shedding in faeces is often intermittent. Furthermore, pooling can enhance the likelihood of detection due to collective sample representation. Farms are frequently ranked based on test outcomes (Arnold, 2005), and high prevalence may result in financial penalties. Therefore, understanding how different pooling strategies influence inferred prevalence is crucial for both epidemiological assessment and farm management.

Using a grouped-hypergeometric sampling framework, we extend the results in Arnold (2005) by incorporating biologically informed test-sensitivity models. We also show how Hellinger information can be applied to design efficient sampling strategies that improve the accuracy of prevalence estimates while controlling costs.

Now we assumes a single pen containing N pigs, among which T_X are infected. Faecal samples are collected and pooled for testing, with each pool composed of equal contributions from $\bar{N}$ individuals. Each individual contributes to at most one pool. The test sensitivity model, adapted from Cannon (2002) and applied by Arnold (2005), accounts for organism clustering in faecal matter, the pooling and homogenisation process, and the sub-sampling for PCR analysis. For a pool containing faecal material from $\bar{N}$ individuals, the individual-level test sensitivity δ is given by:

$$1-\delta = \exp\left(-\frac{w_p C}{\bar{N}}\left(1-\exp\left(-\frac{\widehat{\rho}}{w_p}\right)\right)\right) = \exp\left(-w_s C\left(1-\exp\left(-\frac{\widehat{\rho}}{w_s\bar{N}}\right)\right)\right), \qquad (1)$$

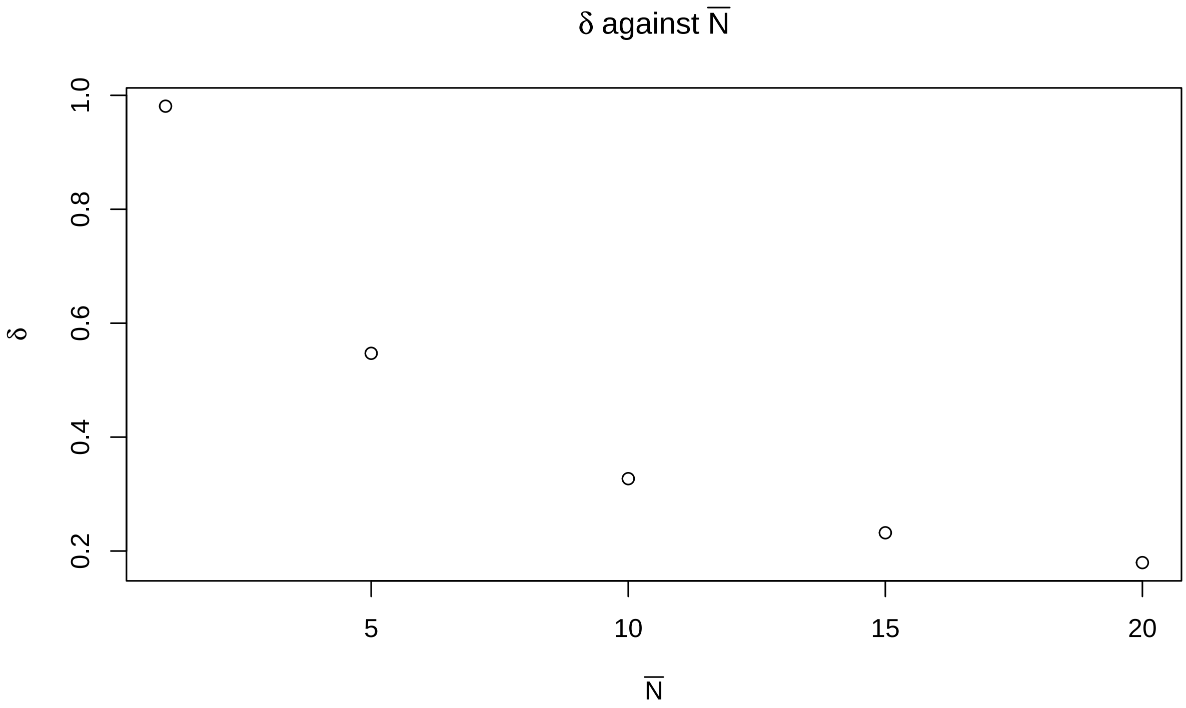
where:

- w_s is the weight of faeces contributed by each individual,
- $w_p = w_s\bar{N}$ is the total pool weight,
- w_a is the aliquot weight used for culturing,
- C is the average number of clusters per gram,
- $\widehat{\rho} \approx \mathbb{E}(M)\rho w_a$ combines organism count per cluster, amplification probability, and aliquot weight.

For this application, $C = 7.3$ and $\widehat{\rho} = 0.55$ were estimated empirically in Arnold (2005). Assuming no false positives, the test specificity is set to $\lambda = 1$. The grouped-hypergeometric PMF incorporating imperfect test sensitivity is used to model the probability distribution of observed positive group tests. Based on this distribution, the Hellinger information are computed, providing a principled way to compare and optimize alternative sampling designs.

```
N <- 60                          # Population size
barN <- c(20, 15, 10, 5, 1)  # Different group sizes
b <- rep(1,5)                    # Number of groups
lambda <- 1                      # Item-level specificity
wp <- 20                         # Total weight of the pooled sample
ws <- wp/barN                    # Weight per item
rho_hat <- 0.55                  # Empirical parameter for PCR
C <- 7.3                         # Average number of clusters per gram
Tx_values<-seq(0,60,by= 1)   # Range of contaminated items

# Sensitivity function (Equation 19)
calculate_delta_ws <- function(barN, ws, rho_hat, C) {
  1 -  exp(-ws * C * (1 - exp(-rho_hat / (ws*barN))))
}
# Calculate item-level sensitivity (delta)
item_level_sensitivities_case2 <- calculate_delta_ws(barN, ws, rho_hat, C)
plot(barN,item_level_sensitivities_case2,xlab = expression(bar(N)),ylab=expression(delta),main=expressi
```



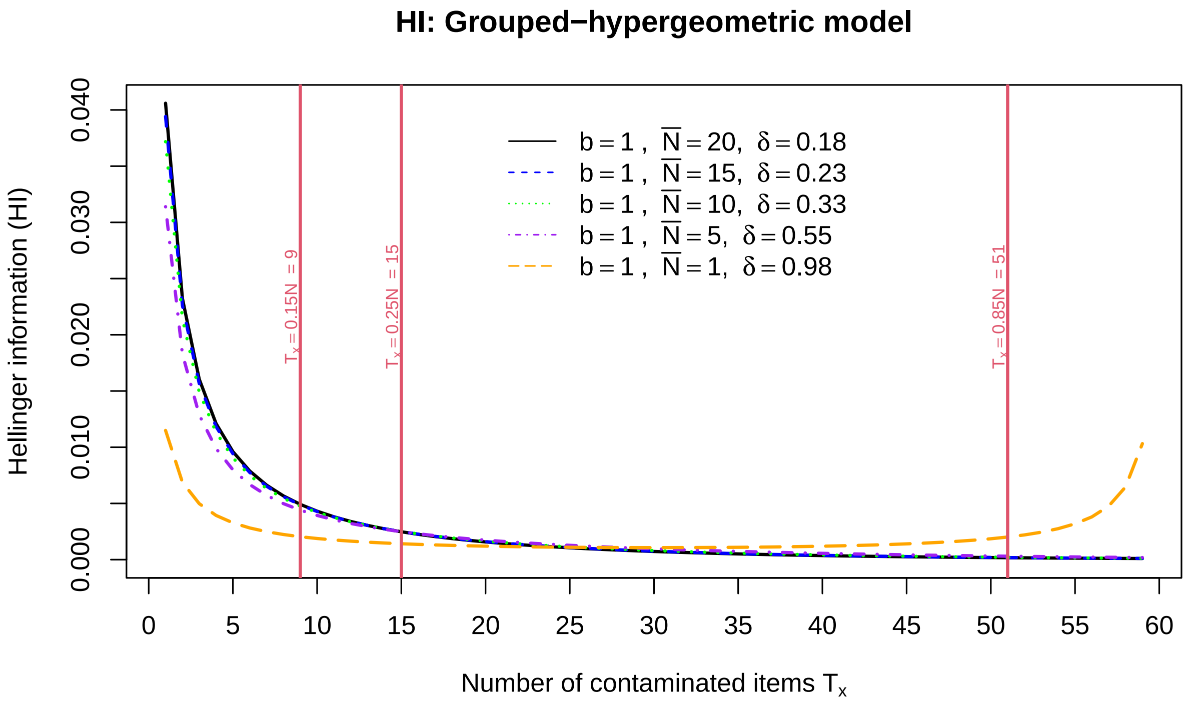
Now we can estimate HI for T_X values under different item-level sensitivity.

```
fisher_results_HG_PMF <- lapply(seq_along(barN), function(i) {
n <- barN[i]; b_i <- b[i]; ws_i <- ws[i]
delta <- calculate_delta_ws(barN=n, ws=ws_i, rho_hat=rho_hat,C=C)
lambda <- 1
fisher_info <- lapply(Tx_values, function(Tx) {
    info_hg_tx_imperfect(
      N = N, barN = n, Tx = Tx, b = b_i,
      delta = delta, lambda = lambda,
      method = "PMF-HI", type = "item",verbose = FALSE
    )
  })
  HI <- sapply(fisher_info, function(x) x$FI_Tx)
  list(barN = n, b=b_i,ws=ws_i,delta=delta,lambda=lambda,HI = HI)
})
```

Now we plot the trends of HI for given T_X in below Figure. The Figure shows Hellinger information for the grouped-hypergeometric distribution as a function of the number of infected individuals (T_X), assuming a pen size of $N = 60$ and a total sample weight of 20 grams. The sample is treated as a single group ($b = 1$), with different pooling sizes ($\bar{N} = 20, 10, 5, 4, 2, 1$) representing how many individuals contributed to the pooled sample. Red vertical lines indicate reference prevalence levels: the average reported ($T_X = 15$), a low ($T_X = 9$), and a high ($T_X = 51$) contamination level. Sensitivity is derived from a fecal concentration model shown above; specificity is assumed perfect ($\lambda = 1$).

```
colors <- c("black", "blue", "green", "purple", "orange")
range_FI <- range(fisher_results_HG_PMF[[1]]$HI,
                  fisher_results_HG_PMF[[2]]$HI,
                  fisher_results_HG_PMF[[3]]$HI,
                  fisher_results_HG_PMF[[4]]$HI,
                  fisher_results_HG_PMF[[5]]$HI)

plot(Tx_values[c(2:60)], fisher_results_HG_PMF[[1]]$HI[c(2:60)], type = "l",
  xlab = expression(paste("Number of contaminated items ", T[x])), ylab = "Hellinger information (HI)",
  main = "HI: Grouped-hypergeometric model",
  col = colors[1], lwd = 2,ylim=range_FI,xaxt = "n",yaxt="n")
axis(side = 1, at = seq(0, 60, by = 5))
axis(side = 2, at = seq(0, range_FI[2], by = 0.005))
for (i in 2:length(barN)) {
  lines(Tx_values[c(2:60)], fisher_results_HG_PMF[[i]]$HI[c(2:60)], type = "l", col = colors[i], lwd = 2
}
legend_labels <- lapply(seq_along(barN), function(i) {
  substitute(b == x * ~ ", " ~ bar(N) == y * ", " ~ delta == z, list(x = b[i], y =  barN[i], z =  round
})
legend(
  20,0.04, legend = legend_labels,
  col = colors, lty = 1:length(barN), cex = 1, bty = "n"
)
abline(v = c(9, 15, 51), col = c(2, 2, 2), lwd = 2)
line_labels <- c(expression(T[x] == 0.15 * N ~ " = 9"),
                 expression(T[x] == 0.25 * N ~ " = 15"),
                 expression(T[x] == 0.85 * N ~ " = 51"))
text(x = c(9, 15, 51), y = par("usr")[4]*0.5, labels = line_labels, pos = 3, col = 2, cex = 0.7,srt = 9
```



7.2 Design exploration: Grid search over b and $\bar{n}$

Here we wish to explore an optimal design based on HI. For this purpose, we estimates HI for all of sort combination of b and $\bar{N}$ for a given $T_X = 15$. We use the pig data as well where sensitivity is a function of $\bar{N}$.

First we create a grid of b, $\bar{N}$, w_s and δ and then we will estimate HI for $T_X = 15$. Later we will use the same grid to estimate HI assuming approximate BN model. Then we will compare the HI heatmap to explore a better sampling design based on the HI.

```
N <- 60 # Population size
max_group_size <- 20
max_groups <- 20
wn <- 20          # a total weight of faecal matter tested
lambda <- 1       # Specificity for item-level
rho_hat <- 0.55   # Empirical parameter for PCR
C <- 7.3  # Average number of clusters per gram
Tx_values <- c(15)
ws <- c()
b_store <- c()
barN_store <- c()
for (b in 1:max_groups) {
  for (N_bar in 1:max_group_size) {
    if (b * N_bar <= N) {
      b_store <- c(b_store, b)
      barN_store <- c(barN_store, N_bar)
      ws <- c(ws, wn / (b * N_bar))
    }
  }
}
combinations <- data.frame(b = b_store, barN = barN_store, ws = ws)
combinations$delta <- calculate_delta_ws(combinations$barN, ws=combinations$ws, rho_hat, C)
head(combinations)
#>   b barN        ws     delta
#> 1 1    1 20.000000 0.9809425
#> 2 1    2 10.000000 0.8619512
#> 3 1    3  6.666667 0.7328911
#> 4 1    4  5.000000 0.6284508
#> 5 1    5  4.000000 0.5470888
#> 6 1    6  3.333333 0.4831742
```

Now we calculate HI for each combination based on HG model.

```
fisher_results_HG_PMF <- lapply(1:dim(combinations)[1], function(i) {
  n <- combinations$barN[i]
  b_i <- combinations$b[i]
  ws_i <- combinations$ws[i]
  delta <- combinations$delta[i]
  lambda <- 1
  fisher_info <- lapply(Tx_values, function(Tx) {
    info_hg_tx_imperfect(
      N = N, barN = n, Tx = Tx, b = b_i,  # Use correct `b_i`
      delta = delta, lambda = lambda,
      method = "PMF-HI", type = "item",verbose=FALSE
```

```r
    )
  })
  HI <- sapply(fisher_info, function(x) x$FI_Tx)
  list(barN = n, b=b_i,ws=ws_i,delta=delta,lambda=lambda,HI = HI)
})
HI_HG_Tx_15 <- rep(NA, dim(combinations)[1])
for (i in 1: dim(combinations)[1]) {
  HI_HG_Tx_15[i] <- fisher_results_HG_PMF[[i]]$HI
}
combinations$HI_HG_Tx_15 <- HI_HG_Tx_15
combinations_HG_Tx_15 <- combinations[,c("b","barN","HI_HG_Tx_15")]
head(combinations_HG_Tx_15)
#>   b barN HI_HG_Tx_15
#> 1 1    1 0.001413078
#> 2 1    2 0.002100933
#> 3 1    3 0.002349936
#> 4 1    4 0.002445416
#> 5 1    5 0.002484719
#> 6 1    6 0.002501055
```

In similar way we calculate HI for each combination based on approximate BN model.

```r
fisher_results_BN_PMF <- lapply(1:dim(combinations)[1], function(i) {
  n <- combinations$barN[i]
  b_i <- combinations$b[i]  # Ensure correct mapping of b to barN
  ws_i <- combinations$ws[i]
  delta <- combinations$delta[i]
  lambda <- 1
  fisher_info <- sapply(Tx_values, function(Tx) {
    info_bn_tx_imperfect(
      N = N, barN = n, Tx = Tx, b = b_i,
      delta = delta, lambda = lambda,
      method = "PMF-HI", type = "item"
    )
  })
  list(barN = n, b=b_i,ws=ws_i,delta=delta,lambda=lambda,HI = fisher_info)
})
HI_BN_Tx_15 <- rep(NA, dim(combinations)[1])
for (i in 1: dim(combinations)[1]) {
  HI_BN_Tx_15[i] <- fisher_results_BN_PMF[[i]]$HI
}
combinations$HI_BN_Tx_15 <- HI_BN_Tx_15
combinations_BN_Tx_15 <- combinations[,c("b","barN","HI_BN_Tx_15")]
head(combinations_BN_Tx_15)
#>   b barN HI_BN_Tx_15
#> 1 1    1 0.001413078
#> 2 1    2 0.002086308
#> 3 1    3 0.002330314
#> 4 1    4 0.002426585
#> 5 1    5 0.002468537
#> 6 1    6 0.002487868
```

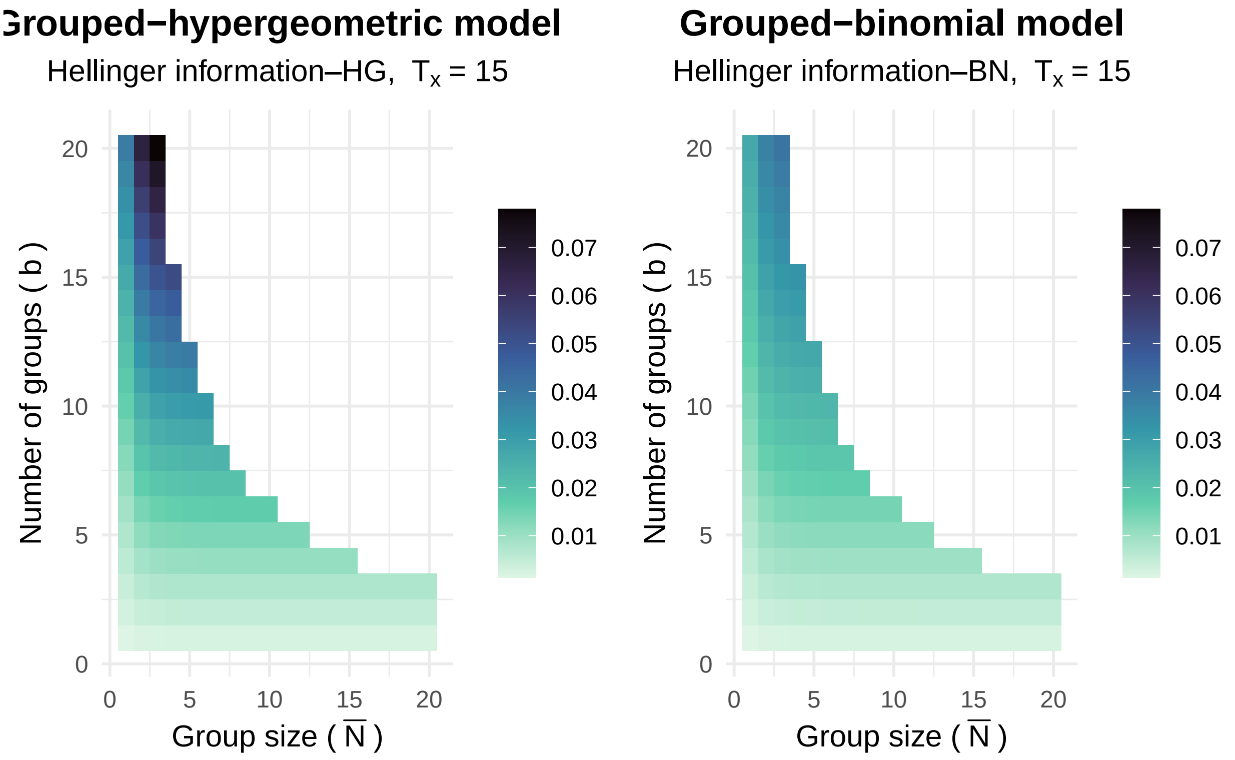
We created heatmap for both cases.

```r
min_value <- min(combinations_HG_Tx_15$HI_HG_Tx_15, combinations_BN_Tx_15$HI_BN_Tx_15, na.rm = TRUE)
max_value <- max(combinations_HG_Tx_15$HI_HG_Tx_15, combinations_BN_Tx_15$HI_BN_Tx_15, na.rm = TRUE)

Tx_15_HG <- ggplot(combinations_HG_Tx_15, aes(x = barN, y = b, fill = HI_HG_Tx_15)) +
  geom_tile() +
  scale_fill_viridis_c(option = "mako", direction = -1,name = NULL, n.breaks = 10,limits = c(min_value,
                       guide = guide_colourbar(barheight = unit(6, "cm"))) +
  labs(x = expression("Group size (" ~ bar(N) ~ ")"),
       y = expression("Number of groups (" ~ b ~ ")"),
       title = "Grouped-hypergeometric model",
       subtitle = expression("Hellinger information-HG, " ~ T[x] ~ "= 15")) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5))


Tx_15_BN <- ggplot(combinations_BN_Tx_15, aes(x = barN, y = b, fill = HI_BN_Tx_15)) +
  geom_tile() +
  scale_fill_viridis_c(option = "mako", direction = -1,name = NULL, n.breaks = 10,limits = c(min_value,
                       guide = guide_colourbar(barheight = unit(6, "cm"))) +
  labs(x = expression("Group size (" ~ bar(N) ~ ")"),
       y = expression("Number of groups (" ~ b ~ ")"),
       title = "Grouped-binomial model",
       subtitle = expression("Hellinger information-BN, " ~ T[x] ~ "= 15")) +
  theme_minimal(base_size = 14) +        # Clean theme
  theme(plot.title = element_text(hjust = 0.5, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5))

ggarrange(Tx_15_HG,Tx_15_BN,ncol = 2,nrow = 1)
```



8 Session Info & References

```
sessionInfo()
#> R version 4.4.0 (2024-04-24 ucrt)
#> Platform: x86_64-w64-mingw32/x64
#> Running under: Windows 11 x64 (build 26100)
#>
#> Matrix products: default
#>
#>
#> locale:
#> [1] LC_COLLATE=English_Australia.utf8  LC_CTYPE=C
#> [3] LC_MONETARY=English_Australia.utf8 LC_NUMERIC=C
#> [5] LC_TIME=English_Australia.utf8
#>
#> time zone: Australia/Sydney
#> tzcode source: internal
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#> [1] metR_0.18.1     ggpubr_0.6.0    ggplot2_3.5.1   groupedHG_0.1.0
#>
#> loaded via a namespace (and not attached):
#>  [1] utf8_1.2.4        generics_0.1.3    tidyr_1.3.1       rstatix_0.7.2
#>  [5] digest_0.6.35     magrittr_2.0.3    evaluate_1.0.1    grid_4.4.0
#>  [9] pkgload_1.3.4     fastmap_1.2.0     sessioninfo_1.2.2 pkgbuild_1.4.4
#> [13] backports_1.5.0   tinytex_0.57      urlchecker_1.0.1  promises_1.3.0
#> [17] purrr_1.0.2       fansi_1.0.6       viridisLite_0.4.2 scales_1.3.0
#> [21] shiny_1.8.1.1     abind_1.4-5       cli_3.6.2         rlang_1.1.4
#> [25] ellipsis_0.3.2    cowplot_1.1.3     munsell_0.5.1     remotes_2.5.0
#> [29] withr_3.0.2       cachem_1.1.0      yaml_2.3.8        devtools_2.4.5
#> [33] tools_4.4.0       memoise_2.0.1     checkmate_2.3.1   ggsignif_0.6.4
#> [37] dplyr_1.1.4       colorspace_2.1-0  httpuv_1.6.15     broom_1.0.6
#> [41] mime_0.12         vctrs_0.6.5       R6_2.5.1          lifecycle_1.0.4
#> [45] htmlwidgets_1.6.4 fs_1.6.4          car_3.1-2         usethis_3.1.0
#> [49] miniUI_0.1.1.1    pkgconfig_2.0.3   later_1.3.2       pillar_1.9.0
#> [53] gtable_0.3.5      profvis_0.4.0     Rcpp_1.0.12       glue_1.7.0
#> [57] data.table_1.15.4 highr_0.11        xfun_0.52         tibble_3.2.1
#> [61] tidyselect_1.2.1  rstudioapi_0.16.0 knitr_1.47        xtable_1.8-4
#> [65] farver_2.1.2      htmltools_0.5.8.1 rmarkdown_2.27    carData_3.0-5
#> [69] labeling_0.4.3    compiler_4.4.0
```

9 References

- Barnes, B., Mahdi, P., Das, S., & Clark, R. (2025). *Hypergeometric and Binomial Group Sampling with Sensitivity and Specificity*. Submitted to *Communications in Statistics – Theory and Methods*.
- Arnold, M. E., Cook, A., & Davies, R. (2005). A modelling approach to estimate the sensitivity of pooled faecal samples for isolation of Salmonella in pigs. *Journal of the Royal Society Interface*, 2(4), 365–372.
- Theobald, C. M., & Davie, A. M. (2014). Group testing, the pooled hypergeometric distribution, and estimating the number of defectives in small populations. *Communications in Statistics – Theory and Methods*, 43(14), 3019–3026.