VM0048: Leakage Belt Delineation

Winrock Intl

2025-05-16

Table of Contents

[Project Area & Jurisdictions 1](#_Toc198303084)

[Derive Leakage Area Belt 2](#_Toc198303085)

[Derive Leakage Area Masks 3](#_Toc198303086)

[Roads Mask 3](#_Toc198303087)

[Habitat Mask 5](#_Toc198303088)

[Slope Mask 6](#_Toc198303089)

[Visual Check 7](#_Toc198303090)

[Apply Leakage Area Masks 8](#_Toc198303091)

[Tally Features 10](#_Toc198303092)

## Project Area & Jurisdictions

country = sf::read\_sf("./data/AOI/liberia\_boundary\_national.shp") |>sf::st\_transform(32629)  
counties = sf::st\_read("./data/AOI/places\_poly\_county.shp") |>sf::st\_transform(32629)

Reading layer `places\_poly\_county' from data source   
 `/Users/seamus/repos/rspb-redd-risk-new/data/AOI/places\_poly\_county.shp'   
 using driver `ESRI Shapefile'  
Simple feature collection with 16 features and 3 fields  
Geometry type: POLYGON  
Dimension: XY  
Bounding box: xmin: -11.50675 ymin: 4.353908 xmax: -7.367323 ymax: 8.551925  
Geodetic CRS: WGS 84

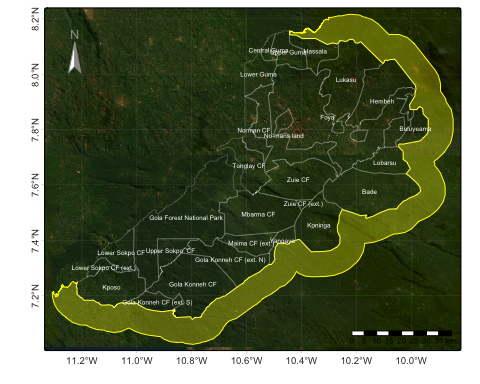
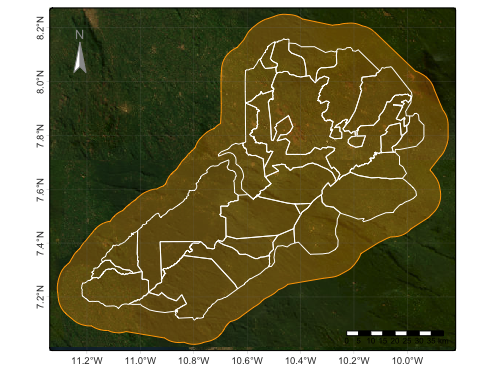
jurisdiction = counties |>dplyr::filter(name=="Grand Cape Mount County"|name=="Gharpolu County")  
jurisdiction$name = 'Grand Cape Mount & Gharpolu Counties'  
  
aoi = sf::read\_sf("/Users/seamus/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/AOI/Archive/ProjectArea\_CF-Expansion\_051525/updated\_areas.shp") |>  
 sf::st\_make\_valid() |>  
 sf::st\_transform("EPSG:32629") |>   
 sf::st\_cast("MULTIPOLYGON") |> sf::st\_as\_sf() |> dplyr::select("Name")  
  
aoi$area\_ha = round(as.numeric(sf::st\_area(aoi) \* 0.0001, 4))  
aoi |> sf::st\_drop\_geometry() |> janitor::adorn\_totals()

Name area\_ha  
 Lukasu 46997  
 Mbarma CF 44323  
 Lower Sokpo CF 14069  
 Upper Sokpo CF 10923  
 Zuie CF 36236  
 Gola Forest National Park 89042  
 Tonglay CF 19595  
 Norman CF 12349  
 Foya 104885  
 Kposo 30763  
 Gola Konneh CF 53102  
 Gola Konneh CF (ext. S) 10897  
 Gola Konneh CF (ext. N) 28590  
 Lower Sokpo CF (ext.) 4771  
 Maima CF (ext.) 13916  
 Zuie CF (ext.) 1495  
 Yangaya 5834  
 Koninga 31741  
 Lobarsu 16548  
 Bade 51367  
 No-mans land 39447  
 Lower Guma 14219  
 Central Guma 3850  
 Upper Guma 13771  
 Hassala 14570  
 Buluyeama 10021  
 Hembeh 46184  
 Total 769505

# Derive Leakage Area Belt

aoi\_union = sf::st\_transform(aoi, 32629) |> sf::st\_union() |> sf::st\_make\_valid()  
leakage\_buffer = sf::st\_buffer(aoi\_union, dist = 5500, endCapStyle="ROUND") |>  
 sf::st\_as\_sf()  
leakage\_buffer = concaveman::concaveman(leakage\_buffer, concavity=5) |>   
 sf::st\_zm() |> sf::st\_make\_valid()  
leakage\_belt\_whole = sf::st\_buffer(leakage\_buffer, dist = 4500, endCapStyle="ROUND") |>  
 sf::st\_make\_valid()   
leakage\_belt = sf::st\_difference(leakage\_belt\_whole, st\_union(st\_combine(aoi\_union)))  
leakage\_belt$area\_ha = round(as.numeric(sf::st\_area(leakage\_belt) \* 0.0001, 4))   
leakage\_belt = sf::st\_intersection(country, leakage\_belt)  
  
tmap::tmap\_mode("plot")  
tmap::tm\_shape(leakage\_belt\_whole) +   
 tmap::tm\_polygons(col="orange",fill="orange",fill\_alpha=0.3,lwd=1)+  
 #tmap::tm\_add\_legend(type="lines",col="orange",labels="Leakage Belt (10km)") +  
 tmap::tm\_shape(aoi) + tmap::tm\_borders(lwd=1, col="white") +  
 tmap::tm\_basemap("Esri.WorldImagery")  
  
tmap::tm\_shape(leakage\_belt) +   
 tm\_polygons(col="yellow",fill="yellow",fill\_alpha=0.3)+  
 #tmap::tm\_add\_legend(type="lines",col="yellow",labels="Leakage Belt (10km)") +   
 tmap::tm\_shape(aoi) + tmap::tm\_borders(lwd=0.2, col="white") +   
 tmap::tm\_text(text="Name", size=0.3, col="white") +  
 tmap::tm\_basemap("Esri.WorldImagery")  
  
# save locally  
#sf::wt\_write(leakage\_belt, "OneDrive.../20087 - RSPB Gola Feasibility/Deliverables/  
# Spatial Data/LEAKAGE/Archive/LeakageBelt\_10k-Radius\_UnFiltered.zip")   
#sf::st\_write(leakage\_belt\_whole, "OneDrive.../20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/LEAKAGE/Archive/LeakageBelt\_10k-Radius\_UnFiltered-Whole.shp")

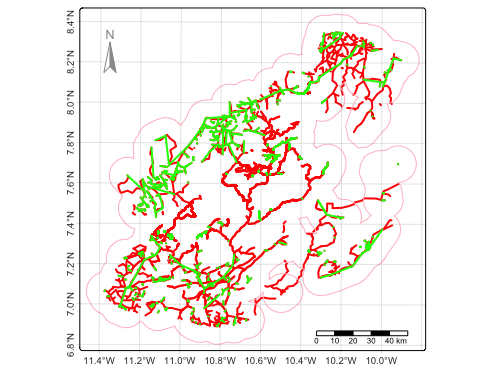
Simple feature collection with 1 feature and 4 fields  
Geometry type: MULTIPOLYGON  
Dimension: XY  
Bounding box: xmin: 245117.9 ymin: 776808.3 xmax: 406590.1 ymax: 908414.4  
Projected CRS: WGS 84 / UTM zone 29N  
 Name Shape\_Leng Shape\_Area geometry  
1 Leakage Belt 10km Radius 138549.7 469973056 MULTIPOLYGON (((245306.7 79...  
 area\_ha  
1 298838



# Derive Leakage Area Masks

## Roads Mask

roads\_one = sf::st\_read("~/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/ROADS/Archive/roads\_simplified\_one.shp")  
roads\_two = sf::st\_read("~/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/ROADS/Archive/roads\_simplified\_two.shp")  
  
# we have simplify mask shapefiles and split them up to shorten computing   
# time & avoid crashing. See option for "harsh" simplification on line 163  
roads\_one\_simplified = roads\_one |> sf::st\_make\_valid() |> sf::st\_cast("MULTILINESTRING") |>   
 rmapshaper::ms\_simplify(keep=0.5)  
roads\_two\_simplified = roads\_two |> sf::st\_make\_valid() |> sf::st\_cast("MULTILINESTRING") |>   
 rmapshaper::ms\_simplify(keep=0.5)  
  
# bigger file needs more simplificaiotn  
roads\_one\_simplified\_harsh = rmapshaper::ms\_simplify(  
 roads\_one\_simplified, keep=0.01)   
  
# now apply buffer operation, but note this takes time. Its   
# advised processing inputs as much as possible before running  
roads\_one\_buffer = sf::st\_buffer(  
 roads\_one\_simplified\_harsh,   
 dist = 10000,   
 nQuadSegs = 5,  
 endCapStyle="ROUND",   
 joinStyle = "ROUND",  
 mitreLimit = 1,  
 singleSide = FALSE  
 )  
  
roads\_two\_buffer = sf::st\_buffer(  
 roads\_two\_simplified,   
 dist = 10000,   
 nQuadSegs = 5,  
 endCapStyle="ROUND",   
 joinStyle = "ROUND",  
 mitreLimit = 1,  
 singleSide = FALSE  
 )  
  
# Combine, dissolve and cast to single feature  
roads\_mask = sf::st\_combine(roads\_one\_buffer, roads\_two\_buffer) |>  
 sf::st\_union() |> sf::st\_cast("POLYGON")  
  
# Visual check  
tmap::tmap\_mode("plot")  
tmap::tm\_shape(roads\_mask) + tmap::tm\_borders(lwd=0) +  
 tmap::tm\_shape(roads\_one\_simplified\_harsh) + tmap::tm\_lines(lwd=2, col="red") +  
 tmap::tm\_shape(roads\_two\_simplified) + tmap::tm\_lines(lwd=2, col="green") +  
 tmap::tm\_shape(roads\_mask) + tmap::tm\_borders(lwd=1, col="pink") +   
 tmap::tm\_graticules(lines=T,labels.rot=c(0,90),lwd=0.2) +  
 tmap::tm\_scale\_bar(position = c("RIGHT", "BOTTOM"), text.size = .5) +   
 tmap::tm\_compass(color.dark = "gray60", text.color = "gray60", position = c("left", "top"))  
  
# Save output to MASKS folder and purge memory  
#sf::st\_write(roads\_mask, "/Users/seamus/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/MASKS/LeakageMask\_Roads\_10km-Buffer\_051625.shp", delete\_dsn=T)



## Habitat Mask

# import inputs  
wetlands = terra::rast("/Users/seamus/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/HABITAT/Wetlands/GLWD\_EPSG32629.tif")  
protected\_areas = sf::st\_read("/Users/seamus/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/HABITAT/Protected Areas/Archive/WDPA\_Mar2025\_Public\_32629\_GOLA.shp")

leakage\_belt\_crop = sf::st\_transform(sf::st\_as\_sf(leakage\_belt\_whole), 32629) |> terra::vect()  
wetlands = terra::crop(wetlands, leakage\_belt\_crop, mask=T)  
  
# tidy labeling  
code\_dict\_2 <- data.frame(  
 id = c(1, 4, 7, 10, 12, 14, 15, 18, 20, 21, 26, 31),  
 label = c(  
 "Freshwater lake", # 1  
 "Large river", # 4  
 "Small streams", # 7  
 "Riverine, regularly flooded, forested", # 10  
 "Riverine, seasonally flooded, forested", # 12  
 "Riverine, seasonally saturated, forested", # 14  
 "Riverine, seasonally saturated, non-forested", # 15  
 "Palustrine, seasonally saturated, forested", # 18  
 "Ephemeral, forested", # 20  
 "Ephemeral, non-forested", # 21  
 "Tropical peatland, forested", # 26  
 "Other coastal wetland" # 31  
 ))  
  
levels(wetlands) <- code\_dict\_2  
wetlands[wetlands == 0] <- NA  
  
# derive wetland mask  
wetlands\_mask <- wetlands  
wetland\_classes <- c(1, 4, 7, 10, 12, 14, 15, 18, 20, 21, 26, 31)  
terra::values(wetlands\_mask) <- ifelse(terra::values(wetlands) %in% wetland\_classes, 1, NA)  
  
# save locally for faster computing  
#raster::writeRaster(wetlands\_mask, "/Users/seamus/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/MASK/LeakageMask\_Wetland-GLWD\_051625.tif", overwrite=T)  
#sf::st\_write(protected\_areas, "/Users/seamus/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/MASK/LeakageMask\_ProtectedAreas\_WDPA\_051625.shp", delete\_dsn=T)

## Slope Mask

# skipping these operations here (est. time 12 mins)  
DEM = terra::rast("/Users/seamus/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/DEM/DEM\_EPSG32629.tif")   
  
# derive slope percentage from degree   
slope\_degrees = terra::terrain(DEM, v="slope", unit="degrees")  
slope\_percent = tan(slope\_degrees \* (pi / 180)) \* 100  
slope\_percent = terra::clamp(slope\_percent, 0, 100)   
slope\_invalid = slope\_percent > 10  
slope\_invalid[slope\_invalid == 0] <- NA  
slope\_mask = terra::as.polygons(slope\_invalid, dissolve=T)|>sf::st\_as\_sf()|>sf::st\_union()  
  
# save locally & reload  
sf::st\_write(slope\_mask, "/Users/seamus/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/MASK/LeakageMask\_Slope10%-Invalid\_051625.zip", delete\_dsn=T)  
slope\_mask = sf::st\_read("/Users/seamus/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/MASK/Archive/LeakageMask\_Slope10%-Invalid\_051625/slope\_poly\_simplified.shp")

## Visual Check

# Visual check  
tmap::tmap\_mode("plot")  
tmap::tm\_shape(roads\_mask) + tmap::tm\_borders(lwd=0) +  
 tmap::tm\_shape(wetlands) + tm\_raster(col.legend = tm\_legend("Wetlands (GLWD")) +  
 tmap::tm\_shape(leakage\_belt) + tm\_polygons(col="yellow",fill="yellow",fill\_alpha=0.4, lwd=1.5)+   
 tmap::tm\_shape(protected\_areas) + tm\_polygons(fill="ORIG\_NAME", fill.legend = tm\_legend("Protected Areas (WDPA)")) +  
 tmap::tm\_shape(aoi) + tmap::tm\_borders(lwd=1.5, col="red") +   
 tmap::tm\_text(text="Name", size=0.3, col="black") +  
 tmap::tm\_shape(roads\_mask) + tmap::tm\_borders(col="purple") +  
 tmap::tm\_graticules(lines=T,labels.rot=c(0,90),lwd=0.2) +  
 tmap::tm\_scalebar(position = c("RIGHT", "BOTTOM"), text.size = .5) +   
 tmap::tm\_compass(color.dark = "gray60", text.color = "gray60", position = c("left", "top"))



# Apply Leakage Area Masks

# clip  
roads\_leakage\_mask = sf::st\_intersection(leakage\_belt, roads\_mask)  
slope\_leakage\_mask = sf::st\_intersection(leakage\_belt, slope\_mask)  
wetlands\_leakage\_mask = sf::st\_intersection(leakage\_belt, wetlands\_mask)  
protected\_areas\_leakage\_mask = sf::st\_intersection(leakage\_belt, protected\_areas)  
  
# merge. Btw actually makes better sense to keep these seperate.   
# They are easier to operate seperately due to their size and linework.  
leakage\_mask\_a = sf::st\_union(roads\_leakage, slope\_leakage\_mask)  
leakage\_mask\_b = sf::st\_union(leakage\_mask\_a, wetlands\_leakage\_mask)  
leakage\_mask\_c = sf::st\_union(leakage\_mask\_b, protected\_areas\_leakage\_mask)  
  
# save  
sf::st\_write(leakage\_mask\_c, "/Users/seamus/Library/CloudStorage/OneDrive-WinrockInternationalInstituteforAgriculturalDevelopment/20087 - RSPB Gola Feasibility/Deliverables/Spatial Data/LEAKAGE/Leakage Masks

A map of a large area

AI-generated content may be incorrect.

A map of a large area

AI-generated content may be incorrect.

## 

## Tally Features

road\_count\_whole = sf::st\_intersection(road, leakage\_belt\_whole)  
road\_count = sf::st\_intersection(road, leakage\_belt)  
road\_length\_whole = sum(sf::st\_length(road\_count\_whole)) + sum(sf::st\_length(road\_count\_whole))  
road\_length = sum(sf::st\_length(road\_count)) + sum(sf::st\_length(road\_count))  
road\_length\_whole  
road\_length  
  
waterways\_count\_whole = sf::st\_intersection(waterways, leakage\_belt\_whole)  
waterways\_count = sf::st\_intersection(waterways, leakage\_belt)  
waterways\_length\_whole = sum(sf::st\_length(waterways\_count\_whole))  
waterways\_length = sum(sf::st\_length(waterways\_count))  
waterways\_length\_whole  
waterways\_length  
  
places\_count\_whole = sf::st\_intersection(places, leakage\_belt\_whole)  
places\_count = sf::st\_intersection(places, leakage\_belt)  
places\_count\_whole  
places\_count