```
# ------------------------------------------------------------------------------
# Function 'spatseg'
#
# Author: Seong-Yun Hong <hong.seongyun@gmail.com>
# ------------------------------------------------------------------------------
spatseg <- function(env, method = "all", useC = TRUE, negative.rm = FALSE,
  tol = .Machine$double.eps) {

  validObject(env)
  dd <- env@data + tol; ee <- env@env + tol

  # Check if any of the original data values are 0 or less (often as a result
  # of kernel smoothing)
  negIDs <- apply(dd, 1, function(z) any(z <= 0))
  if (sum(negIDs) > 0 && negative.rm) {
    warning("rows with negative values have been removed", call. = FALSE)
    dd <- dd[-which(negIDs),]
    ee <- ee[-which(negIDs),]
  } else if (sum(negIDs) > 0 && !negative.rm) {
    warning("negative values have been replaced with 'tol'", call. = FALSE)
    index <- (dd <= 0)
    dd <- replace(dd, index, tol)
    ee <- replace(ee, index, tol)
  }

  negIDs <- apply(ee, 1, function(z) any(z <= 0))
  if (sum(negIDs) > 0 && negative.rm) {
    warning("rows with negative values removed", call. = FALSE)
    dd <- dd[-which(negIDs),]
    ee <- ee[-which(negIDs),]
  } else if (sum(negIDs) > 0 && !negative.rm) {
    warning("negative values replaced with 'tol'", call. = FALSE)
    index <- (ee <= 0)
    dd <- replace(dd, index, tol)
    ee <- replace(ee, index, tol)
  }

  method <- match.arg(method, c("exposure", "information", "diversity",
                                "dissimilarity", "all"), several.ok = TRUE)
  if ("all" %in% method)
    method <- c("exposure", "information", "diversity", "dissimilarity")

  P <- matrix(0, nrow = 0, ncol = 0)
  H <- numeric(); R <- numeric(); D <- numeric()

  if (useC) {
    m <- ncol(dd)
    method <- c("exposure" %in% method, "information" %in% method,
                "diversity" %in% method, "dissimilarity" %in% method)
    tmp <- .Call("spseg", as.vector(dd), as.vector(ee),
                          as.integer(m), as.integer(method))
    results <- list(); n <- m^2
    if (!is.na(tmp[1])) {
      results$p <- matrix(tmp[1:n], ncol = m, byrow = TRUE)
      rownames(results$p) <- colnames(results$p) <- colnames(dd)
    }
    if (!is.na(tmp[n+1]))
      results$h <- tmp[n+1]
    if (!is.na(tmp[n+2]))
      results$r <- tmp[n+2]
    if (!is.na(tmp[n+3]))
      results$d <- tmp[n+3]
  }

  else {
    # Number of population groups
    m <- ncol(dd)
    # Total population in the study area
    ptsSum <- sum(dd)
    # Population of all groups at each data point
    ptsRowSum <- apply(dd, 1, sum)
    # Total population of each subgroup
    ptsColSum <- apply(dd, 2, sum)
    # Proportion of each subgroup
    ptsProp <- ptsColSum / ptsSum
    # Population proportion of each group at each local environment
    envProp <- t(apply(ee, 1, function(z) z/sum(z)))

    if ("exposure" %in% method) {
      P <- matrix(0, nrow = m, ncol = m)
      rownames(P) <- colnames(P) <- colnames(dd)
      for (i in 1:m) {
        A <- dd[, i] / ptsColSum[i]
        for (j in 1:m) {
          P[i, j] <- sum(A * envProp[, j])
        }
      }
    }

    if ("information" %in% method) {
      Ep <- apply(envProp, 1, function(z) sum(z * log(z, base = m))) * -1
      E <- sum(ptsProp * log(ptsProp, base = m)) * -1
      H <- 1 - (sum(ptsRowSum * Ep) / (ptsSum * E))
    }

    if ("diversity" %in% method) {
      Ip <- apply(envProp, 1, function(z) sum(z * (1 - z)))
      I <- sum(ptsProp * (1 - ptsProp))
      R <- 1 - sum((ptsRowSum * Ip) / (ptsSum * I))
    }

    if ("dissimilarity" %in% method) {
      I <- sum(ptsProp * (1 - ptsProp))
      constant <- ptsRowSum / (2 * ptsSum * I)
      Dp <- t(apply(envProp, 1, function(z) abs(z - ptsProp)))
      D <- sum(apply(Dp, 2, function(z) sum(z * constant)))
    }
    results <- list(p = P, h = H, r = R, d = D)
  }

  SegSpatial(results$d, results$r, results$h, results$p,
             env@coords, env@data, env@env, env@proj4string)
}
```