

Obtaining host-parasite occurrence data from Genbank

Andrew Park

```
#source("http://bioconductor.org/biocLite.R")
#library(BiocInstaller)
#biocLite("genbankr")
library(genbankr)

library(rentrez)
library(plyr)
library(dplyr)
library(magrittr)

#' finds hosts of parasite based on genbank sequences
#'
#' @param x character parasite latin binom
#' @param retmax numeric number of parasite uid records to obtain
#' @param parallel should we go ahead and run this on multiple cores?
#' @param cores number of cores to utilize
#'
#' @returns df a data.frame consisting of columns para.name, host.name, uid, acc.num

findHostsbySeq <- function(x, retmax = 1000, parallel = FALSE,
  cores = NA) {
  require(plyr)
  require(dplyr)
  require(magrittr)
  para.uid <- entrez_search(db = "nucleotide", term = x, retmax = retmax)$ids
  para.acc <- host <- vector()

  if (parallel) {
    require(doMC)
    if (is.na(cores)) {
      stop("Please identify the number of cores you wish to use")
    }
    cl <- doMC::registerDoMC(cores)
    df <- ldply(para.uid, .parallel = TRUE, .paropts = list(.export = para.uid),
      function(x) {
        parAcc <- entrez_fetch(db = "sequences", id = x,
          rettype = "acc")
        para.acc <- gsub("\n", "", x = parAcc)
        if (para.acc == "") {
          host <- NA
        } else {
          tmp <- try(readGenBank(GBAccession(para.acc),
            ret.seq = F, partial = T, verbose = F)$sources$elementMetadata$listData$host)
          if (is.null(tmp) | inherits(tmp, "try-error")) {
            host <- NA
          } else {
            host <- tmp
          }
        }
      })
  }
```

```

        return(c(host.name = host, uid = x, acc.num = para.acc))
    })
    df$para.name <- x
    df %<>% filter(host.name != "" & !is.na(host.name))
}

if (parallel == FALSE) {

  for (i in para.uid) {
    parAcc <- entrez_fetch(db = "sequences", id = i,
      rettype = "acc")
    para.acc[i] <- gsub("\n", "", x = parAcc)
    if (para.acc[i] == "") {
      host[i] <- NA
    } else {
      tmp <- try(readGenBank(GBAccession(para.acc[i]),
        ret.seq = F, partial = T, verbose = F)@sources@elementMetadata@listData$host)
      if (is.null(tmp) | inherits(tmp, "try-error")) {
        host[i] <- NA
      } else {
        host[i] <- tmp
      }
    }
  }
  df <- data.frame(para.name = x, host.name = host, uid = para.uid,
    acc.num = para.acc)
  df %<>% filter(host.name != "" & !is.na(host.name) &
    host.name != x)
}
return(df)
}

A.ovale <- findHostsbySeq("Amblyomma ovale",
  retmax=80, parallel=TRUE, cores=2)

Y.pestis <- findHostsbySeq("Yersinia pestis")

```