**ROTINA 001**

**SELEÇÃO DE PALAVRAS-CHAVES PELA METODOLOGIA PECO**

**Leonardo da Silva Tomadon**

**Programa de Pós-Graduação em Engenharia e Tecnologia Ambiental (PPGETA)**

**Universidade Federal do Paraná (UFPR)**

**1.** Formular a busca ingênua, que seria o conjunto de palavras-chave formuladas a partir de um conhecimento prévio no assunto dentro da estrutura PECO, pare entender melhor, veja grames et al. 2019.

Exemplo de string grames et al. 2019:

(((woodpecker\* OR sapsucker\* OR Veniliorn\* OR Picoid\* OR Dendropic\* OR Melanerp\* OR Sphyrapic\*) AND (fire\* OR burn\* OR wildfire\*)) AND (((nest\* OR reproduct\* OR breed\* OR fledg\*) AND (succe\* OR fail\* OR surviv\*)) OR (surviv\* OR mortalit\* OR death\*) OR (‘food availab\*’ OR forag\* OR provision\*) OR ( emigrat\* OR immigrat\* Ordispers\*)) AND (occup\* OR occur\* OR presen\* OR coloniz\* OR colonis\* OR abundan\* OR ‘population size’ OR ‘habitat suitability’ OR ‘habitat selection’ OR persist\*))

**2.** Ao formular seu conjunto de palavras chave (string), o próximo passo é fazer a primeira busca nas bases de dados, seja ela qual for, scopus, web of science.

**3.** Após essa busca inicial, é necessário fazer o download dos arquivos, para a pasta de trabalha, recomenda-se o formato bibtex para os arquivos baixados.

**4.** O próximo passo é rodar o script no software R. Basta entrar com os arquivos baixados e rodar esse script.

#instalando pacote  
install.packages("remotes")  
install.packages("dplyr")  
install.packages("readr")  
install.packages("igraph")  
install.packages("ggraph")  
install.packages("RefManageR")

install.packages(“litsearchr”)

install. packages(“bibtex”)

#chamar a instalação do litsearchr pelo github  
install\_github("elizagrames/litsearchr", ref="main")

#carregando pacotee  
library(remotes)  
library(readr)  
library(ggraph)  
library(igraph)  
library(dplyr)  
library(litsearchr)  
library(RefManageR)  
library(bibtex)

## definindo a sua pasta de trabalho  
setwd("H:/Tomadon/18 -Palotina/PAPER\_VALORACAO/BUSCA\_JAQUE")

# Carregar o arquivo exportado

naive\_results <- import\_results(file= c("G:/Tomadon/18 -Palotina/PAPER\_VALORACAO/BUSCA\_JAQUE/scopus.bib")

naive\_results\_wos <- import\_results(file= c("savedrecs.bib", "scopus.bib"))

## remover duplicados   
dedupe1 <- litsearchr::remove\_duplicates(naive\_results\_wos, field = "title", method = "exact")  
View(dedupe1)

#nome das colunas  
colnames(dedupe1)

#contagem de termos nao dispniveis das seções  
sum(is.na(dedupe1[, "abstract"]))  
sum(is.na(dedupe1[, "keywords"]))  
sum(is.na(dedupe1[, "title"]))

## extração de termos  
rakedkeywords <-   
 litsearchr::extract\_terms(dedupe1,  
 text = paste(dedupe1$title, dedupe1$abstract),  
 method = "fakerake",  
 min\_freq = 1,  
 ngrams = TRUE,  
 min\_n = 1,  
 max\_n = 5,  
 language = "English")

## atribui a extração de termos a uma nova variável  
terms2 <- rakedkeywords

### metodo raked  
dfm2 <- create\_dfm(elements = dedupe1$abstract,   
 features = terms2)  
  
## rede de interação dos termos  
naivegraph <- litsearchr::create\_network(search\_dfm = dfm2, min\_studies = 5, min\_occ = 2)

#computa força das interaçoes  
strengths <- strength(naivegraph)  
  
#transforma força das interaçoes em uma planilha organizada  
term\_strengths <- data.frame(term=names(strengths), strength=strengths, row.names=NULL) %>%  
 mutate(rank=rank(strength, ties.method="min")) %>% arrange(strength)  
  
# gera imagem da força das palavras em funçao do ranking  
cutoff\_fig <- ggplot(term\_strengths, aes(x=rank, y=strength, label=term)) +  
 geom\_line() +  
 geom\_point() +  
 geom\_text(data=filter(term\_strengths, rank>5), hjust="right", nudge\_y=20, check\_overlap=TRUE)  
  
# Seleciona linha de corte por metodo acumulativo  
cutoff <- find\_cutoff(naivegraph,   
 method="cumulative",   
 percent = 0.8)  
  
# plot da linha de corte por metodo acumulativo  
cutoff\_fig + geom\_hline(yintercept=cutoff, linetype="dashed")  
  
# Isola as palavras com base nos criterios de corte  
reduced <- litsearchr::reduce\_graph(naivegraph, cutoff\_strength = cutoff[1])  
  
# Isola as palavras   
selected\_terms <- litsearchr::get\_keywords(reduced)  
  
# Salvar os dados em um planilha  
write.csv(selected\_terms, "palavras\_extraidas.csv")

**5.** O resultado será tabela .csv chamada de palavras\_extraidas. Dentro deste arquivo, você vai encontrar uma seleção de palavras chave, selecionadas a partir da matriz de co-ocorrência. O objetivo é reformular a string da busca ingênua, com as novas palavras-chave, adicionando ou retirando palavras, ou refazendo uma totalmente nova.

**6.** Com a nova string formulada a partir da metodologia, será feita uma nova busca na base de dados escolhidas., chamada de busca avançada. O objetivo é que essa nova busca seja mais direcionada, e que resulte em uma amostra maior que a busca ingênua.

**Referência**

Grames, E. M., Stillman A. N., Morgan, W. T., Elphick, C. S. An automated approach to identifying search terms for systematic reviews using keyword co‐occurrence networks. Methods in Ecology and Evolution, 2019;10:1645–1654. DOI: 10.1111/2041-210X.13268