

## 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, para 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* OR dispers*)) 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 trabalho, 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("elizagrimes/litsearchr", ref="main")
```

```
#carregando pacotes
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(naive_results)

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

## extração de termos
rakedkeywords <-
  litsearchr::extract_terms(naive_results,
    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 interações
strengths <- strength(naivegraph)

#transforma força das interações 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 função 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