

```

---
title: "Untitled"
author: "Nathan Brouwer"
date: "9/21/2021"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```

A complete bioinformatics workflow in R

By: Nathan L. Brouwer

"Worked example: Building a phylogeny in R"

Introduction

- Understand ~~and able~~ how to use R. Markdown
- Create a Phylogenetic tree for the shroom family.

Vocab

Accession number - unique number assigned by a particular database as an additional means to find the article

Fasta File: text-based format for representing either nucleotide sequences or amino acid sequence.

Pairwise Alignment: identify regions of similarity between ~~Files~~ Files

MSA (Multiple sequence Alignment): Maximizes the similarity between Files.

distance Matrix: Summarize

reproducible work flow: R code that can ~~compute~~ be adaptable to compute.

~~not in R~~ double colon :: get the ~~package~~ function from that package

Package :: function { /n: ~~creates~~ creates a new line; new line character

```
## Software Preliminaries
```

```
### Download necessary packages
```

```
### Load packages into memory
```

```
`{r, message= F, warning=F}  
# github packages  
library(compbio4all) -> devtools::install_github("brouweren/compbio4all")  
# CRAN packages -> install.packages("CRAN")  
library(rentrez) -> install.packages("rentrez")  
library(seqinr)  
library(ape)  
  
# Bioconductor packages  
library(msa)  
library(Biostrings)  
`{r
```

```
## XXXXXXXXing macromolecular sequences
```

```
`{r}  
# Human shroom 3 (H. sapiens)  
hShroom3 <- entrez_fetch(db = "protein",  
                        id = "NP_065910",  
                        rettype = "fasta")  
`{r
```

cat function respects the new line character.

```
`{r}  
cat(hShroom3)  
`{r
```

- humans have 3 different shroom genes.

```

---
title: "Untitled"
author: "Nathan Brouwer"
date: "9/21/2021"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```

A complete bioinformatics workflow in R

By: Nathan L. Brouwer

"Worked example: Building a phylogeny in R"

Introduction

- Understand ~~and~~ how to use R. Markdown
- Create a Phylogenetic tree for the shroom family.

Vocab

Accession number - unique number assigned by a particular database as an additional means to find the article

Fasta File: text-based format for representing either nucleotide sequences or amino acid sequence.

Pairwise Alignment: identify regions of similarity between ~~Files~~ Files

MSA (Multiple sequence Alignment): Maximizes the similarity between Files.

distance Matrix: Summarize

reproducible work flow: R code that can ~~compute~~ be adaptable to compute.

~~get the function from that package~~ get the function from that package

Package function { /n: creates a new line; new line character

```
## Software Preliminaries
```

```
### Download necessary packages
```

```
### Load packages into memory
```

```
`{r, message= F, warning=F}  
# github packages  
library(compbio4all) -> devtools::install_github("brouweren/compbio4all")  
# CRAN packages -> install.packages("CRAN")  
library(rentrez) -> install.packages("rentrez")  
library(seqinr)  
library(ape)  
  
# Bioconductor packages  
library(msa)  
library(Biostrings)  
`{r
```

```
## XXXXXXXXing macromolecular sequences
```

```
`{r}  
# Human shroom 3 (H. sapiens)  
hShroom3 <- entrez_fetch(db = "protein",  
                        id = "NP_065910",  
                        rettype = "fasta")  
`{r
```

cat function respects the new line character.

```
`{r}  
cat(hShroom3)  
`{r
```

- humans have 3 different shroom genes.

type of fasta
same code that assigns the ~~data~~ data to its appropriate variable.

```
```{r}
Mouse shroom 3a (M. musculus)
mShroom3a <- entrez_fetch(db = "protein",
 id = "AAF13269",
 rettype = "fasta")
```

```
Human shroom 2 (H. sapiens)
hShroom2 <- entrez_fetch(db = "protein",
 id = "CAA58534",
 rettype = "fasta")
```

```
Sea-urchin shroom
sShroom <- entrez_fetch(db = "protein",
 id = "XP_783573",
 rettype = "fasta")
...`
```

# nchar(number of character) % ~~does~~ <sup>include</sup> ~~characters~~ /n a

```
```{r}
nchar(hShroom3) // this is row sequence that is why /n is
nchar(mShroom3a) counted as character.
nchar(sShroom)
nchar(hShroom2)
```

``` L> number of amino of acids in the human shroom 2.

```
Prepping macromolecular sequences
```

```
```{r}
fasta_cleaner
```

```{r}
fasta_cleaner <- function(fasta_object, parse = TRUE){

  fasta_object <- sub("^(>)(.*?) (\\n) (.*?) (\\n\\n)", "\\4", fasta_object)
  fasta_object <- gsub("\\n", "", fasta_object)

  if(parse == TRUE){
    fasta_object <- stringr::str_split(fasta_object,
                                       pattern = "",
                                       simplify = FALSE)
  }

  return(fasta_object[[1]])
}
```
```

```
```{r}
hShroom3 <- fasta_cleaner(hShroom3, parse = F)
mShroom3a <- fasta_cleaner(mShroom3a, parse = F)
hShroom2 <- fasta_cleaner(hShroom2, parse = F)
sShroom <- fasta_cleaner(sShroom, parse = F)
```
```

```
```{r}
hShroom3
```
```

*Aligning*

```
XXXXXXXXing sequences
```

*- Pairwise Alignment.*

```
```{r}
# add necessary function
align.h3.vs.m3a <- Biostrings::PairwiseAlignment(
  hShroom3,
  mShroom3a)
...

```

tab completion

↓

PairwiseAlignment()

compares lines up the sequences of human shroom3 + mice.

