

Módulo 4: Palaeoverse

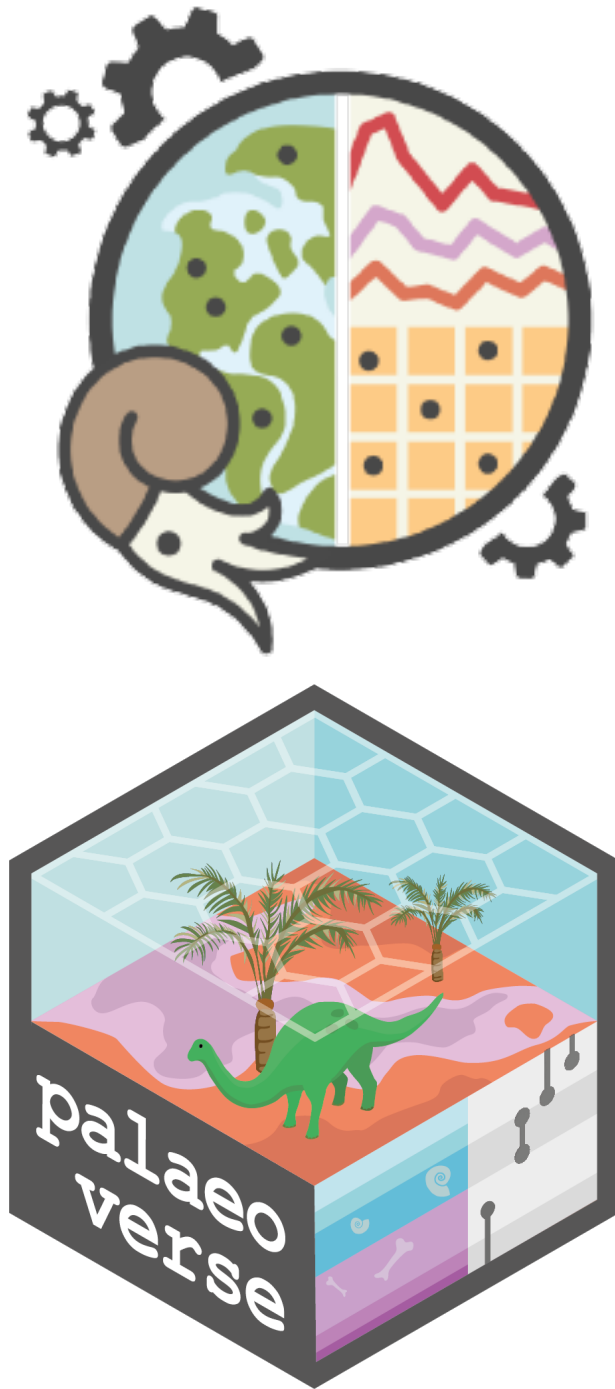
David Caro, GIPHIN

2025-06-14

Table of contents

Palaeoverse	2
Paquetes relacionados	3
Instalación	3
Cargar paquetes	4
Diversidad de Tetrápodos	4
Importar y revisar los datos	4
Explorando las funciones de palaeoverse	5
Datos temporales	5
Datos espaciales	8
Diversidad	8
Macrostrat	11
Contexto Geológico: La Formación Hell Creek	11
Extrayendo los Datos (Fetching)	12
Visualizar	13
Phylopic	14
Funcionamiento	14
Elegir una imagen	15

Palaeoverse



palaeoverse es un paquete de R que busca **facilitar el análisis de datos paleontológicos**.

Proporciona herramientas para:

- Estandarizar datos fósiles
- Analizar riqueza taxonómica y patrones de muestreo
- Realizar simulaciones paleontológicas
- Visualizar y conectar datos de bases abiertas

El paquete es parte de un ecosistema más amplio que incluye paquetes complementarios como `sepkosky`, `phylopic` y `macrostrat`.

Paquetes relacionados

Paquete	¿Qué hace?
<code>palaeoverse</code>	Funciones para estandarizar, analizar y visualizar datos fósiles
<code>sepkosky</code>	Accede a la base de datos de Sepkoski con información de géneros marinos fósiles
<code>phylopic</code>	Permite añadir siluetas de organismos fósiles a gráficos de R
<code>macrostrat</code>	Conecta con la base de datos Macrostrat (estratigrafía y litología)

Instalación

```
# Instalar desde CRAN o GitHub
install.packages("ggspatial")
install.packages("palaeoverse")
install.packages("sepkosky")
install.packages("rmacrostrat")
install.packages("rphylopic")
```

En caso de no funcionar con la instalación típica lo hacemos por GitHub:

```
install.packages("devtools")
devtools::install_github("palaeoverse/palaeoverse")
devtools::install_github("palaeoverse/rmacrostrat")
install.packages("remotes")
remotes::install_github("palaeoverse/rphylopic")
```

Cargar paquetes

```
library(palaeoverse)
library(rphylopic)
library(rmacrostrat)
library(tidyverse)
library(ggplot2)
library(ggspatial)
```

Diversidad de Tetrápodos

Este ejercicio está inspirado por un ejemplo de [Palaeoverse tetrápodos del Carbonífero y Pérmico](#)

Importar y revisar los datos

Como siempre lo primero es revisar la estructura de los datos y su organización

```
# Importamos los datos de Tetrápodos
data(tetrapods)

str(tetrapods)
```

```
'data.frame': 5270 obs. of 32 variables:
 $ occurrence_no : num 146665 147090 147521 147523 147563 ...
 $ collection_no : num 12943 13004 13043 13044 13046 ...
 $ identified_name : chr "n. gen. Anconastes n. sp. vesperus" "Gorgonopsia indet." "n. gen
 $ identified_rank : chr "species" "suborder" "species" "species" ...
 $ accepted_name : chr "Anconastes vesperus" "Gorgonopsia" "Procolophon trigoniceps" "Ar
 $ accepted_rank : chr "species" "suborder" "species" "species" ...
 $ early_interval : chr "Missourian" "Capitanian" "Induan" "Artinskian" ...
 $ late_interval : chr "Gzhelian" NA "Olenekian" NA ...
 $ max_ma : num 306 265 252 290 279 ...
 $ min_ma : num 299 260 247 279 272 ...
 $ phylum : chr "Chordata" "Chordata" "Chordata" "Chordata" ...
 $ class : chr "Osteichthyes" "Osteichthyes" "Reptilia" "Reptilia" ...
 $ order : chr "Temnospondyli" "Cotylosauria" "NO_ORDER_SPECIFIED" "Araeoscelidia
 $ family : chr "Trematopidae" NA "Procolophonidae" "Araeoscelidae" ...
 $ genus : chr "Anconastes" NA "Procolophon" "Araeoscelis" ...
```

```

$ abund_value      : num  2 1 10 2 6 2 1 39 56 2 ...
$ abund_unit       : chr   "specimens" "specimens" "specimens" "individuals" ...
$ lng              : num  -106.3 22 26.5 -98.9 -99.2 ...
$ lat              : num   36.3 -32.4 -32.1 33.6 33.8 ...
$ collection_name  : chr   "El Cobre Canyon" "Hottentotsrivier Farm" "Donnybrook (Queenstown"
$ cc               : chr   "US" "ZA" "ZA" "US" ...
$ formation        : chr   "Cutler" "Abrahamskraal" "Katberg" "Nocona" ...
$ stratgroup       : chr   NA "Beaufort" "Beaufort" "Wichita" ...
$ member           : chr   NA NA NA NA ...
$ zone            : chr   NA "Tapinocephalus Assemblage" NA NA ...
$ lithology1       : chr   "siltstone" "sandstone" "mudstone" "conglomerate" ...
$ environment      : chr   "dry floodplain" "\"floodplain\"" "fluvial indet." "terrestrial i
$ pres_mode        : chr   "body" "body,mold/impression" "body" "body" ...
$ taxon_environment: chr   "brackish,freshwater,terrestrial" "terrestrial" "terrestrial" "te
$ motility          : chr   "actively mobile" "actively mobile" "actively mobile" "actively m
$ life_habit       : chr   "ground dwelling" NA NA NA ...
$ diet             : chr   "carnivore" "carnivore" NA NA ...

```

```
unique(tetrapods$class)
```

```
[1] "Osteichthyes" "Reptilia"      "Saurischia"
```

! Ejercicio

Crea una variable con la edad máxima y otro con la mínima

```

lim_menor <- min(tetrapods$min_ma)
lim_mayor <- max(tetrapods$max_ma)

```

Explorando las funciones de palaeoverse

Datos temporales

Palaeoverse tiene muchas funciones útiles para trabajar con datos temporales como `time_bins()`

```

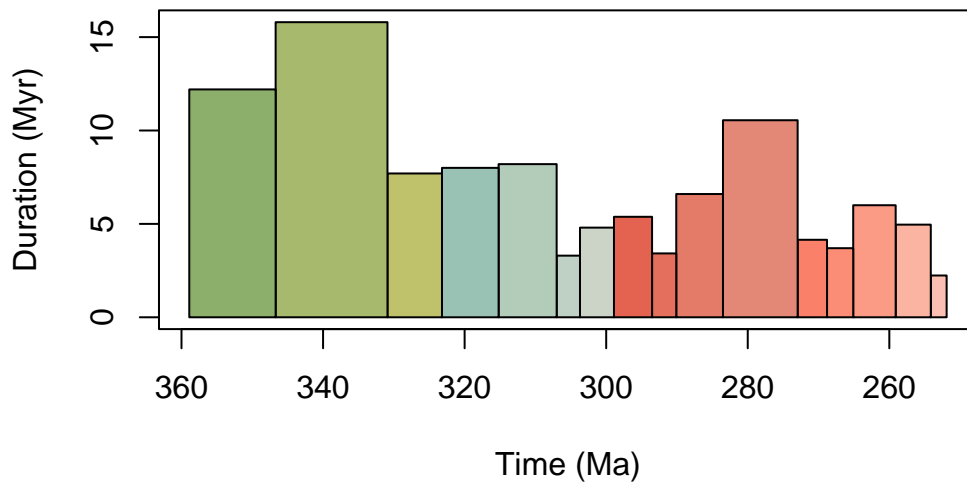
# Creamos el intervalo temporal que deseamos usando time_bins()
pisos<- time_bins(interval = c("Carboniferous", "Permian"),
# Queremos que la división sea en pisos

```

```

rank = "stage",
# Que use los datos de la tabla del 2020
scale = "GTS2020",
# Visualizamos los intervalos creados
plot = TRUE)

```



```
head(pisos)
```

	bin	interval_name	rank	max_ma	mid_ma	min_ma	duration_myrr	abbr	colour	font
1	1	Tournaisian	stage	358.9	352.80	346.7	12.2	T	#8CB06C	black
2	2	Visean	stage	346.7	338.80	330.9	15.8	V	#A6B96C	black
3	3	Serpukhovian	stage	330.9	327.05	323.2	7.7	Se	#BFC26B	black
4	4	Bashkirian	stage	323.2	319.20	315.2	8.0	B	#99C2B5	black
5	5	Moscovian	stage	315.2	311.10	307.0	8.2	M	#B3CBB9	black
6	6	Kasimovian	stage	307.0	305.35	303.7	3.3	Ka	#BFD0C5	black

palaeoverse nos permite ajustar las edades numéricas según el nombre del periodo

```

# Usamos la función look_up que compara los valores y nos regresa dos columnas
tetrapods <- look_up(tetrapods, int_key = interval_key)

```

```

# Reemplazamos valores NA con los originales
# ifelse nos permite hacer la evaluación directamente para todos los elementos
tetrapods$interval_max_ma <- ifelse(is.na(tetrapods$interval_max_ma),
                                   tetrapods$max_ma, tetrapods$interval_max_ma)

tetrapods$interval_min_ma <- ifelse(is.na(tetrapods$interval_min_ma),
                                   tetrapods$min_ma, tetrapods$interval_min_ma)

# Calcular edad media
tetrapods$interval_mid_ma <- (tetrapods$min_ma + tetrapods$max_ma)/2

# str(tetrapods)

# Removemos los valores que estén por fuera del intervalo deseado
cp_tetrapods <- subset(tetrapods, min_ma > min(pisos$min_ma) & max_ma <
                      max(pisos$max_ma))

# Ocurrencias asignadas por los intervalos deseados
# Este método asigna la ocurrencia al intervalo donde está su mayor parte
maj_tetrapods <- bin_time(occdf = cp_tetrapods,
                        bins = pisos,
                        method = 'majority')

colnames(maj_tetrapods)

```

[1]	"occurrence_no"	"collection_no"	"identified_name"
[4]	"identified_rank"	"accepted_name"	"accepted_rank"
[7]	"early_interval"	"late_interval"	"max_ma"
[10]	"min_ma"	"phylum"	"class"
[13]	"order"	"family"	"genus"
[16]	"abund_value"	"abund_unit"	"lng"
[19]	"lat"	"collection_name"	"cc"
[22]	"formation"	"stratgroup"	"member"
[25]	"zone"	"lithology1"	"environment"
[28]	"pres_mode"	"taxon_environment"	"motility"
[31]	"life_habit"	"diet"	"early_stage"
[34]	"late_stage"	"interval_max_ma"	"interval_mid_ma"
[37]	"interval_min_ma"	"id"	"n_bins"
[40]	"bin_assignment"	"bin_midpoint"	"overlap_percentage"

Datos espaciales

`palaeoverse` también nos permite obtener las paleocoordenadas de los puntos usando la función `palaeorotate()`

```
# Obtener las paleocoordenadas de los puntos
maj_tetrapods <- palaeorotate(occdf = maj_tetrapods, age = "bin_midpoint",
                             # Usaremos el famoso Paleomap como referencia
                             method = "point", model = "PALEOMAP")
```

Si deseamos estudiar la distribución espacial de los datos lo mejor es hacer una grilla para así poder agrupar las ocurrencias y estudiarlas de manera discreta

```
# Creamos un intervalo espacial que es basicamente una grilla

maj_tetrapods <- bin_space(occdf = maj_tetrapods,
                          lng = 'p_lng',
                          lat = 'p_lat',
                          spacing = 100)
```

Average spacing between adjacent cells in the primary grid was set to 103.6 km.
H3 resolution: 3

! Ejercicio

Qué variables hemos añadido al data frame original y cuál es el significado de cada una de ellas?

Diversidad

Vamos a usar algunas funciones para ver la diversidad de tetrápodos

```
# Esta función filtra todos los valores únicos en cuanto a los taxones
generos_tetra <- tax_unique(maj_tetrapods,
                            genus = "genus",
                            family = "family",
                            order = "order",
                            class = "class",
                            # Queremos que se tenga hasta el nivel de género
                            resolution = "genus")
```



```
str(generos_tetra)
```

```
'data.frame': 795 obs. of 5 variables:
 $ class      : chr  "Osteichthyes" "Osteichthyes" "Reptilia" "Osteichthyes" ...
 $ order      : chr  "Dicynodontia" "Cotylosauria" NA "Temnospondyli" ...
 $ family     : chr  NA "Charassognathidae" "Nyctiphruretidae" "Zatracheidae" ...
 $ genus      : chr  "Abajudon" "Abdalodon" "Abyssomedon" "Acanthostomatops" ...
 $ unique_name: chr  "Abajudon sp." "Abdalodon sp." "Abyssomedon sp." "Acanthostomatops sp."
```

```
length(unique(maj_tetrapods$genus))
```

```
[1] 793
```

! Ejercicio

¿Por qué el largo de valores únicos de `maj_tetrapods$genus` es 2 filas mayor que el data frame?

Para calcular la diversidad tendremos en cuenta la cantidad de ocurrencias por colección

```
# Esta función nos permite extraer la cantidad de géneros por colección
coll_genera <- group_apply(occdf = maj_tetrapods,
                          group = c("collection_no"),
                          fun = tax_unique,
                          genus = "genus",
                          family = "family",
                          order = "order",
                          class = "class",
                          resolution = "genus")

# Obtenemos los nombres de géneros únicos por colección
unique_genera <- unique(coll_genera[, c("unique_name", "collection_no")])
# Calculamos el número de géneros únicos por colección
coll_taxa <- group_apply(unique_genera, group = "collection_no", fun = nrow)
# Renombramos las columnas
colnames(coll_taxa) <- c("n_taxa", "collection_no")
# Seleccionamos las columnas de edad geológica por colección
coll_info <- maj_tetrapods[, c("collection_no", "max_ma", "interval_mid_ma", "min_ma")]
# Eliminamos colecciones duplicadas
```

```
coll_info <- coll_info[!duplicated(coll_info[1]), ]
# Combinamos la información de edad con la cantidad de géneros
alpha_data <- merge(coll_info, coll_taxa, by = "collection_no")
# Revisamos del nuevo dataframe
str(alpha_data)
```

```
'data.frame':  1238 obs. of  5 variables:
 $ collection_no  : num  11149 11152 11165 11323 12943 ...
 $ max_ma         : num   290 296 299 303 306 ...
 $ interval_mid_ma: num   285 284 294 301 302 ...
 $ min_ma         : num   279 272 290 299 299 ...
 $ n_taxa         : int    1 1 9 1 1 7 6 1 1 1 ...
```

Ahora vamos a graficar usando las funciones de palaeoverse

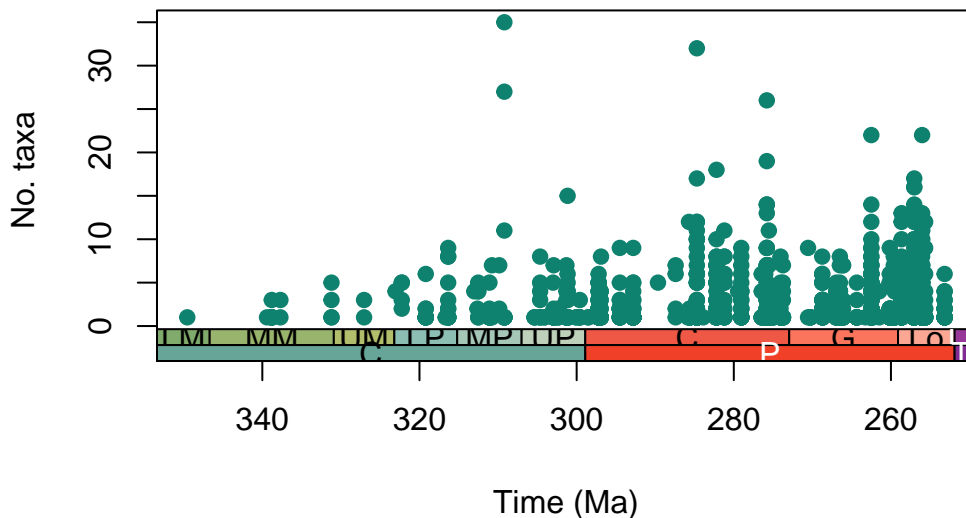
```
# Crear el gráfico de diversidad alfa (número de taxones por colección) en el tiempo
plot(alpha_data$interval_mid_ma,      # Coordenada x: edad media (en Ma) de cada colección
      alpha_data$n_taxa,              # Coordenada y: número de géneros únicos por colección
      axes = FALSE,                  # No dibujar los ejes por defecto, los agregaremos manu
      xlim = rev(range(alpha_data$interval_mid_ma, na.rm = TRUE)), # Invertir el eje x (de p
      xlab = " ",                     # Etiqueta vacía por ahora, la agregaremos más abajo
      ylab = "No. taxa",              # Etiqueta del eje y
      pch = 19,                       # Estilo de punto: sólido
      col = "#0e826f")                # Color del punto (verde azulado)

# Añadir un marco (box) alrededor del gráfico
box()

# Añadir el eje vertical (eje Y) en el lado izquierdo
axis(2)

# Añadir el eje X con nombres geológicos (épocas y periodos)
# Esta función es del paquete `deeptime` y añade intervalos geológicos automáticamente
axis_geo(side = 1, intervals = list("epochs", "periods"))

# Agregar la etiqueta del eje X, desplazada hacia abajo
title(xlab = "Time (Ma)", line = 4)
```



Macrostrat

Todos los datos usados por Macrostrat se encuentran en su base de datos <https://macrostrat.org/>.

Vamos trabajar con datos geológicos reales utilizando el paquete `macrostrat` en R. Nuestro objetivo es encontrar y mapear los afloramientos de la famosa **Formación Hell Creek**, una unidad geológica crucial para entender el límite Cretácico-Paleógeno (K/Pg) en Norteamérica. Este ejercicio está inspirado por un ejemplo de [Palaeoverse](#)

Contexto Geológico: La Formación Hell Creek

La **Formación Hell Creek** es una unidad geológica que aflora en Montana, Dakota del Norte y Dakota del Sur (Estados Unidos). Data del Cretácico Superior (Maastrichtiano) al Paleógeno temprano, y es famosa por su excepcional registro fósil, que incluye dinosaurios icónicos como *Edmontosaurus*, *Triceratops* y *Tyrannosaurus rex*, y mamíferos, justo antes de la extinción masiva del K/Pg hace aproximadamente 66 millones de años. Está compuesta por arcillolitas, lutitas y arenitas depositadas por un sistema fluvial en la costa del Mar Interior Occidental.

Extrayendo los Datos (Fetching)

Lo primero que necesitamos son los identificadores de nombres estratigráficos (`strat_name_ids`) asociados con la Formación Hell Creek. Estos IDs están vinculados a nombres estratigráficos únicos en la base de datos **Macrostrat**. Podemos buscar los IDs relevantes usando las funciones `def_*` que proporcionan información sobre los datos en la base de datos.

```
# Buscar nombres estratigráficos para "Hell Creek"
def_strat_names(strat_name = "hell creek")
```

	strat_name	strat_name_long	rank	strat_name_id	concept_id	bed	bed_id
1	Hell Creek	Hell Creek Formation	Fm	71267	8598		0
2	Hell Creek	Hell Creek Member	Mbr	71268	8598		0
3	Hell Creek	Hell Creek Formation	Fm	867	8598		0

	mbr	mbr_id	fm	fm_id	subgp	subgp_id	gp	gp_id	sgp
1		0	Hell Creek	71267		0	Montana	2565	
2	Hell Creek	71268	Lance	1090		0	Black Buttes coal	69391	
3		0	Hell Creek	867		0			0

	sgp_id	b_age	t_age	b_period	t_period	c_interval	t_units	ref_id
1	0	100.5	56	Cretaceous	Paleogene		0	19
2	0	100.5	56	Cretaceous	Paleogene		0	19
3	0	72.2	66	Cretaceous	Paleogene		13	1

Hay un total de 3 nombres correspondientes, los tres comparten el mismo `concept_id` (8598). Esto indica que estos nombres estratigráficos están unidos bajo una misma entidad geológica. Por lo tanto, podemos usar este `concept_id` para buscar todas las referencias a “Hell Creek”:

Ahora, vamos a usar los `strat_name_ids` asociados con el `concept_id` de Hell Creek para obtener los datos espaciales. Para esto, usaremos la función `get_map_outcrop()`, que nos permite obtener polígonos de afloramiento de mapas geológicos. Especificaremos que queremos la salida como un objeto `sf` (simple features). Un objeto `sf` es una colección de “características simples” (representaciones de objetos del mundo real) que incluye atributos y geometrías en un `data.frame`.

```
# Buscar nombres estratigráficos asociados con el concept_id de Hell Creek
hc_def <- def_strat_names(concept_id = 8598)
```

```
# Obtener información espacial de afloramientos asociada con cada strat_name_id de Hell Creek
hc <- get_map_outcrop(strat_name_id = hc_def$strat_name_id, sf = TRUE)
```

```
head(hc, n = 1)
```

Simple feature collection with 1 feature and 20 fields

Geometry type: MULTIPOLYGON

Dimension: XY

Bounding box: xmin: -110.0199 ymin: 45.86057 xmax: -110.0168 ymax: 45.86277

Geodetic CRS: WGS 84

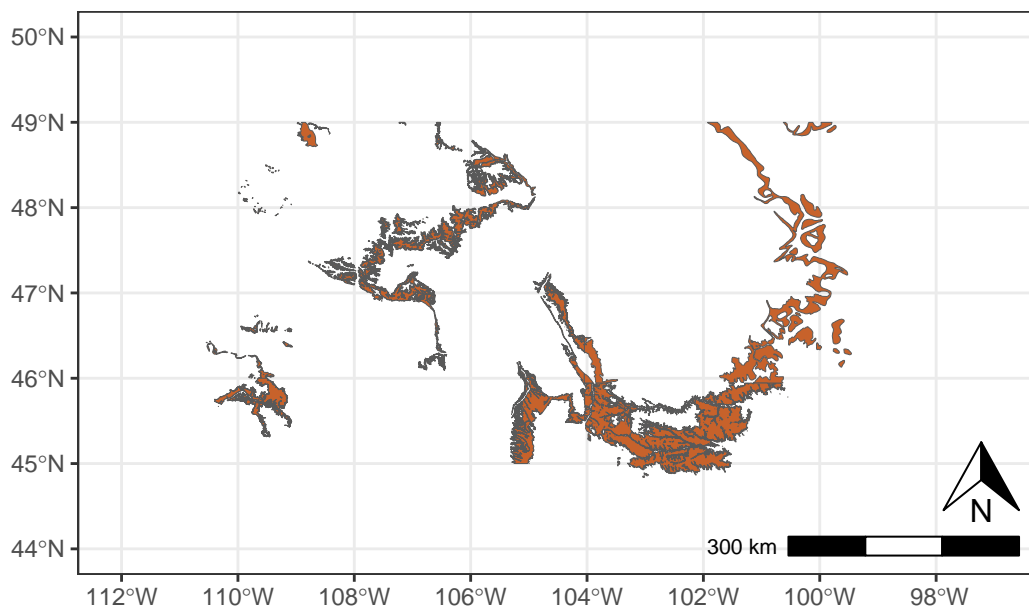
	map_id	source_id		name		strat_name		
1	2092425	25		Hell Creek Formation		Hell Creek Formation		
				lith	descrip	comments	macro_units	strat_names
1	Major::	{sandstone, mudstone}					9213 867, 71267, 71268	
	liths	t_int_id	t_int_age		t_int_name	b_int_id	b_int_age	b_int_name
1	7, 10	34		66	Late Cretaceous	34	100.5	Late Cretaceous
	color	t_age	b_age	best_int_name				geometry
1	#F2FA8C	66	72.1	Maastrichtian	MULTIPOLYGON	(((-110.0168 4...		

Visualizar

Vamos a crear un mapa de forma sencilla

```
# Plot the map
ggplot() +
  geom_sf(data = hc, fill = "#C7622B", lwd = 0) +
  coord_sf(xlim = c(-112, -97), ylim = c(44, 50)) +
  annotation_north_arrow(location = "br",
    pad_y = unit(0.75, "cm"),
    height = unit(1, "cm"), width = unit(1, "cm")) +
  annotation_scale(location = "br", width_hint = 0.3) +
  theme_bw()
```

Scale on map varies by more than 10%, scale bar may be inaccurate



! Ejercicio

Busca información de la Formación que quieras y crea un mapa con su extensión

Phylopic

El paquete **rphylopic** en R facilita la búsqueda y visualización de siluetas de organismos de la base de datos **PhyloPic**, permitiéndote incorporarlas fácilmente a tus gráficos. Puedes añadir estas siluetas como capas o puntos de datos tanto en gráficos de R base como en gráficos de **ggplot2**. Además, **rphylopic** ofrece funciones para elegir entre siluetas disponibles, transformarlas (rotar, cambiar color) y guardar las imágenes.

Funcionamiento

Cada silueta en **phylopic** tiene un **Identificador Único Universal (UUID)**. El primer paso para usar una silueta es obtener su UUID.

La función clave es `get_uuid()`, que permite buscar siluetas en **PhyloPic** usando nombres taxonómicos o filogenéticos (por ejemplo, “*Canis lupus*” o “pan-Mollusca”). Por defecto, `get_uuid()` devuelve el UUID de la primera coincidencia. Una vez que tienes el UUID, puedes usar `get_phylopic()` para obtener la silueta.

Sin embargo, a menudo existen **múltiples siluetas** para un mismo nombre. Puedes usar el argumento `n` en `get_uuid()` para obtener varios UUIDs coincidentes.

```
# Obtener un único UUID para una especie (ej. lobo)
uuid_lobo <- get_uuid(name = "Canis lupus")

# Obtener la imagen para ese UUID
img_lobo <- get_phylopic(uuid = uuid_lobo)

# Pero si existen múltiples siluetas, podemos obtener varios UUIDs:
multi_uuids_lobo <- get_uuid(name = "Canis lupus", n = 5)
print(multi_uuids_lobo)
```

```
[1] "113d2520-9f92-456f-b305-52ee3986172d"
[2] "11658f8c-e0c2-4612-85ef-bdc44acdae0b"
[3] "3492f4ca-01f0-4609-a84f-084a84bf4e95"
[4] "5a0398e3-a455-4ca6-ba86-cf3f1b25977a"
[5] "76c8fdec-d0af-47b9-b949-dc610419d832"
```

Elegir una imagen

Si hay múltiples siluetas, puede ser difícil elegir la correcta sin verlas. La función `pick_phylopic()` es muy útil porque muestra las siluetas y permite seleccionar la que deseas de forma interactiva.

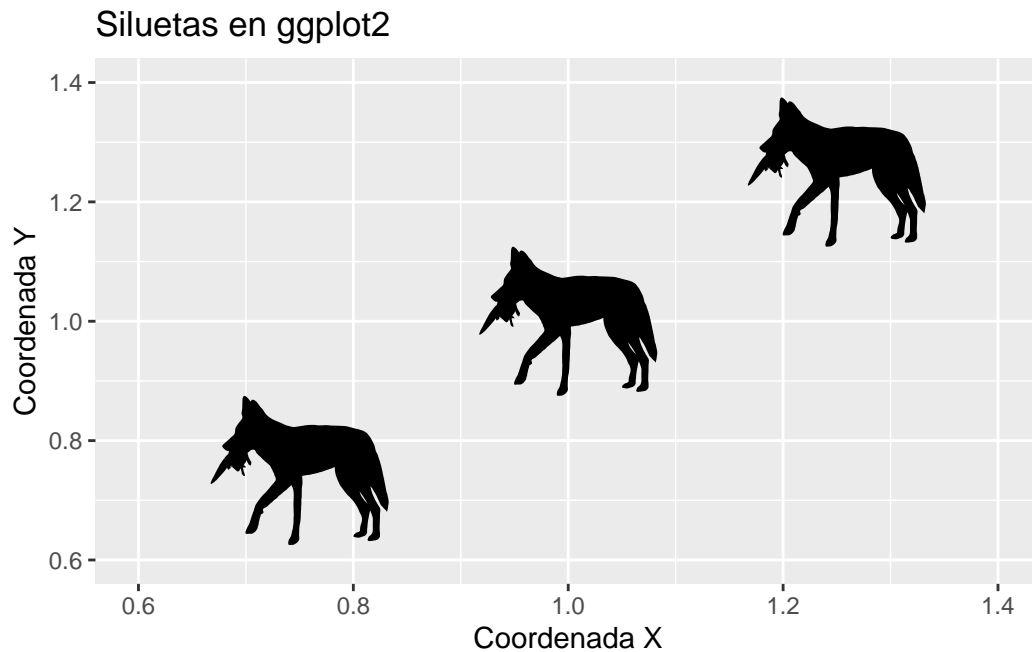
```
# Mostrar las primeras 4 siluetas de Canis lupus e interactuar para elegir
# Una ventana se abrirá para que selecciones la imagen deseada.
# Por ejemplo, si eliges la opción 1:
# img_seleccionada <- pick_phylopic(name = "Canis lupus", n = 4, view = 4)
```

```
# Crear un gráfico ggplot2 base
p <- ggplot() +
  coord_cartesian(xlim = c(0.6, 1.4), ylim = c(0.6, 1.4)) +
  labs(x = "Coordenada X", y = "Coordenada Y", title = "Siluetas en ggplot2")

# Añadir la imagen seleccionada (la que guardamos en img_seleccionada)
p <- p + add_phylopic(img = img_lobo, x = 1.25, y = 1.25, height = 0.25)

# Añadir otra silueta directamente usando el UUID
uuid_ggplot <- get_uuid(name = "Canis lupus", n = 1)
p <- p + add_phylopic(uuid = uuid_ggplot, x = 1, y = 1, height = 0.25)
```

```
# Añadir la primera silueta vinculada a un nombre directamente
p + add_phylopic(name = "Canis lupus", x = 0.75, y = 0.75, height = 0.25)
```



! Ejercicio

Busca y elige siluetas para el taxon que quieras

! Ejercicio

Crea una gráfica de dispersión usando los datos de Palmer Penguins, donde cada punto sea una silueta de pingüino.

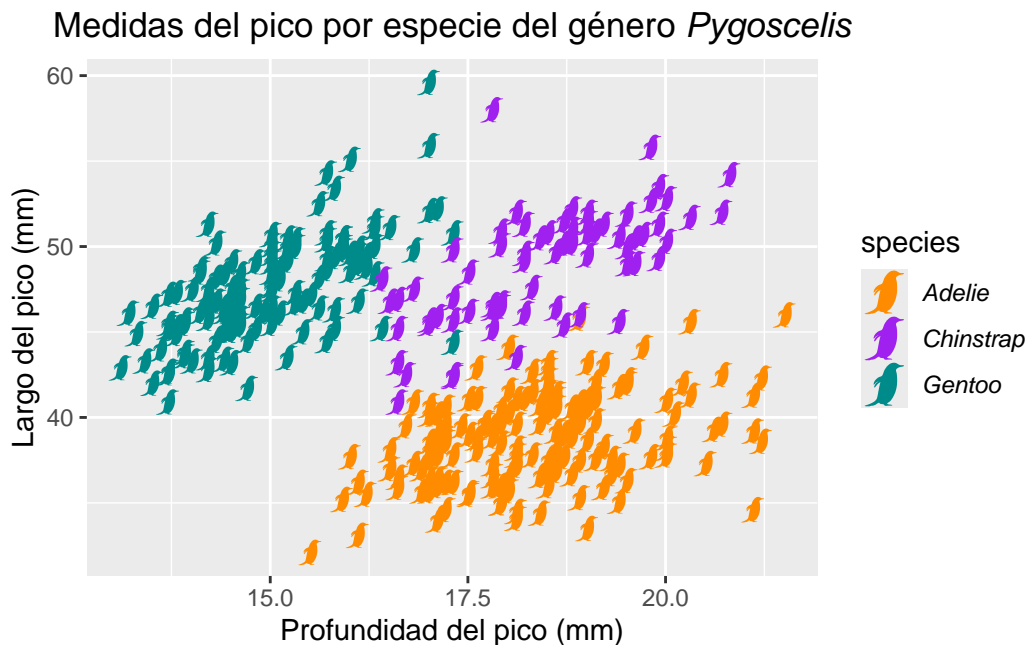
- Deben usar `geom_phylopic(img = pinguino)`

```
uuid <- get_uuid(name = "Pygoscelis", n = 3)
# Get the image for that UUID
pinguino <- get_phylopic(uuid = uuid[3])
```

```
library(palmerpenguins)
# Elegimos la silueta que deseamos
ggplot(penguins) +
```



```
# Gráfica de dispersión
geom_phylopic(img = pinguino,
              aes(x = bill_depth_mm, y = bill_length_mm, color = species),
              height = 1.5, show.legend = TRUE,
              key_glyph = phylopic_key_glyph(img = pinguino)) +
# Seleccionamos la escala de colores para que cuadre
scale_colour_manual(values = c("darkorange", "purple", "cyan4")) +
# Le damos un título
ggtitle(substitute(paste("Medidas del pico por especie del género ", italic("Pygoscelis"))))
# Cambiamos los nombres de los ejes
xlab("Profundidad del pico (mm)") +
ylab("Largo del pico (mm)") +
labs("Especies") +
# Ajustamos el título y cambiamos la leyenda cursiva
theme(plot.title = element_text(vjust = 0, hjust = 0.5), legend.text = element_text(face = 'italic'))
```



```
citation("palaeoverse")
```

To cite palaeoverse in publications, use the following citation:

Jones, L.A., Gearty, W., Allen, B.J., Eichenseer, K., Dean, C.D.,
Galván S., Kouvari, M., Godoy, P.L., Nicholl, C.S.C., Dillon, E.M.,

Flannery-Sutherland, J.T., Chiarenza, A.A. 2023. palaeoverse: A community-driven R package to support palaeobiological analysis. *Methods in Ecology and Evolution*. 14(9), 2205--2215. doi: 10.1111/2041-210X.14099

A BibTeX entry for LaTeX users is

```
@Article{,
  title = {palaeoverse: A community-driven R package to support palaeobiological analysis},
  author = {Lewis A. Jones and William Gearty and Bethany J. Allen and Kilian Eichenseer and},
  year = {2023},
  journal = {Methods in Ecology and Evolution},
  volume = {14(9)},
  pages = {2205--2215},
  doi = {10.1111/2041-210X.14099},
}
```

```
citation("rmacrostrat")
```

To cite rmacrostrat in publications, use the following citation:

Jones, L.A., Dean, C.D., Gearty, W., and Allen, B.J. 2024. rmacrostrat: An R package for accessing and retrieving data from the Macrostrat geological database. *Geosphere*, v. 20:6, 1456--1467. doi: 10.1130/GES02815.1

A BibTeX entry for LaTeX users is

```
@Article{,
  title = {rmacrostrat: An R Package for fetching geologic data from the Macrostrat database},
  author = {Lewis A. Jones and Christopher D. Dean and William Gearty and Bethany J. Allen and},
  year = {2024},
  journal = {Geosphere},
  volume = {20},
  number = {6},
  pages = {1456--1467},
  doi = {10.1130/GES02815.1},
}
```

```
citation("rphylopic")
```

To cite rphylopic in publications, use the following citation:

Gearty, W. and Jones, L.A. 2023. rphylopic: An R package for fetching, transforming, and visualising PhyloPic silhouettes. *Methods in Ecology and Evolution*, 14(11), 2700-2708. doi: 10.1111/2041-210X.14221.

A BibTeX entry for LaTeX users is

```
@Article{,
  title = {rphylopic: An R package for fetching, transforming, and visualising PhyloPic silhouettes},
  author = {William Gearty and Lewis A. Jones},
  year = {2023},
  volume = {14},
  number = {11},
  pages = {2700-2708},
  journal = {Methods in Ecology and Evolution},
  doi = {10.1111/2041-210X.14221},
}
```