ReSANGloW:

A Reproducible Spatial Analysis for Charting Nitrogen Dynamics in Global Wheat Production

Arturo Torres

LIST, ERIN

Wednesday, October 18, 2023

Goal

- Following the indications, the goal of this reproducible code is to (IIASA-BNR, 2023, assignment):

Goal

- Following the indications, the goal of this reproducible code is to (IIASA-BNR, 2023, assignment):
  - "Combine various datasets to generate indictors of nitrogen loss to the environment associated with wheat production at various spatial scales"

Goal

- Following the indications, the goal of this reproducible code is to (IIASA-BNR, 2023, assignment):
  - "Combine various datasets to generate indictors of nitrogen loss to the environment associated with wheat production at various spatial scales"
  - "Provide graphical representations and conduct simple comparisons across a few countries"

Goal

- Following the indications, the goal of this reproducible code is to (IIASA-BNR, 2023, assignment):
  - “Combine various datasets to generate indictors of nitrogen loss to the environment associated with wheat production at various spatial scales”
  - “Provide graphical representations and conduct simple comparisons across a few countries”
  - “Provide a reproducible code associated to these tasks.”

Task 1

- Using SPAM raster data [Wood-Sichra et al., 2016], a new raster at the same resolution, containing wheat production volume (in million tons Mt) is produced.

Task 1

- Using SPAM raster data [Wood-Sichra et al., 2016], a new raster at the same resolution, containing wheat production volume (in million tons Mt) is produced.
- Global scale in a raster format (5 arcminute spatial resolution) estimates of yield in Kg/Ha, physical area in Ha and harvested area in Ha for the year 2005 are available.

Reading SPAM data

```
spam_data = list("yield" = rast("data/SPAM_2005_v3.2/SPAM2005V3r2_global_Y_TA_WHEA_A.tif"),
                 "harvested_area" = rast("data/SPAM_2005_v3.2/SPAM2005V3r2_global_H_TA_WHEA_A.tif"),
                 "physical_area" = rast("data/SPAM_2005_v3.2/SPAM2005V3r2_global_A_TA_WHEA_A.tif"))
```

```
str(spam_data)
```

```
List of 3
 $ yield         :S4 class 'SpatRaster' [package "terra"]
 $ harvested_area:S4 class 'SpatRaster' [package "terra"]
 $ physical_area :S4 class 'SpatRaster' [package "terra"]
```

```
spam_data[['yield']]
```

```
class       : SpatRaster
dimensions  : 1853, 4320, 1  (nrow, ncol, nlyr)
resolution  : 0.08333333, 0.08333333  (x, y)
extent      : -180, 180, -64.41667, 90  (xmin, xmax, ymin, ymax)
coord. ref. : lon/lat WGS 84 (EPSG:4326)
source      : SPAM2005V3r2_global_Y_TA_WHEA_A.tif
name        : SPAM2005V3r2_global_Y_TA_WHEA_A
min value   :                               0
max value   :                           19429
```

Calculate Wheat Production

- Calculate wheat production by multiplying the raster layers for yield (in Kg/Ha) and harvested area (in Ha) using the * operator:

Calculate Wheat Production

- Calculate wheat production by multiplying the raster layers for yield (in Kg/Ha) and harvested area (in Ha) using the * operator:
  - wheat_production = spam_data[["yield"]] * spam_data[["harvested_area"]]

Calculate Wheat Production

- Calculate wheat production by multiplying the raster layers for yield (in Kg/Ha) and harvested area (in Ha) using the * operator:
  - wheat_production = spam_data[["yield"]] * spam_data[["harvested_area"]]
- Convert Units: The resulting values are in Kg, so it is needed to convert them to million tons (Mt). Assuming 1 ton is equal to 1,000 Kg, it is possible to use the following:

Calculate Wheat Production

- Calculate wheat production by multiplying the raster layers for yield (in Kg/Ha) and harvested area (in Ha) using the * operator:
  - wheat_production = spam_data[["yield"]] * spam_data[["harvested_area"]]
- Convert Units: The resulting values are in Kg, so it is needed to convert them to million tons (Mt). Assuming 1 ton is equal to 1,000 Kg, it is possible to use the following:
  - wheat_production_Mt = wheat_production / (1e3 * 1e6)

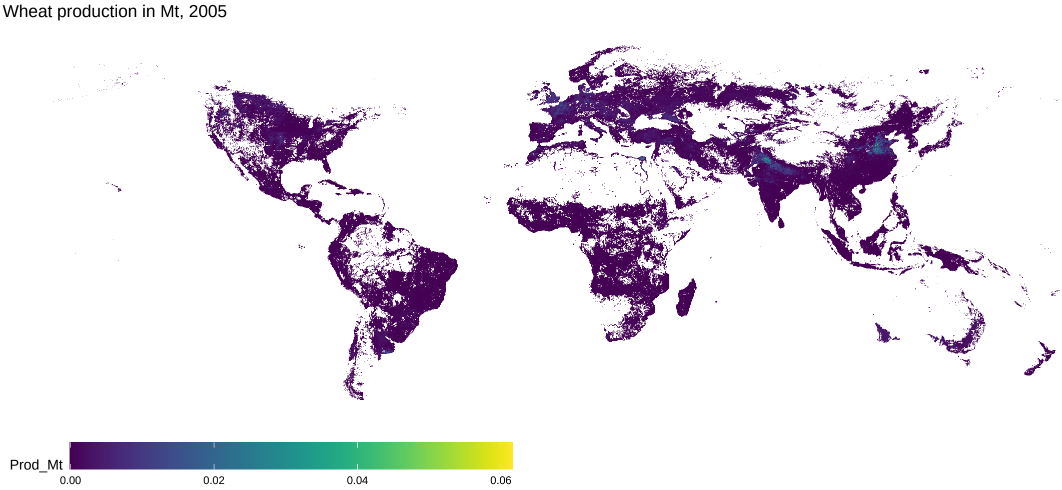
Calculate Wheat Production

- A global map is created and the raster is exported in a geotif format:

```
wheat_production = spam_data[["yield"]] * spam_data[["harvested_area"]]
wheat_production_Mt <- wheat_production / (1e9)
```

```
library(raster)

writeRaster(wheat_production_Mt, filename = "./output/wheat_production_Mt.tif",
            overwrite=TRUE, gdal = c("COMPRESS=DEFLATE", "TFW=YES"))
```

Wheat Production in Mt in 2005



Task 2

- Using the newly created raster and the GAUL shapefile of administrative borders, the production is aggregated to country level and exported to a csv file.

```
library(raster)
library(sf)
library(rgdal)
library(rgeos)


gaul_data_sf <- st_read("data/GAUL/g2015_2005_2.shp")
```

```
Reading layer `g2015_2005_2' from data source
  `/Users/torres/Documents/02_working/3-Production/05_models/32_iiasa/global_n/data/GAUL/g2015_2005_2.shp'
  using driver `ESRI Shapefile'
Simple feature collection with 38189 features and 12 fields
Geometry type: MULTIPOLYGON
Dimension:     XY
Bounding box:  xmin: -180 ymin: -89.9 xmax: 180 ymax: 83.62742
Geodetic CRS:  WGS 84
```

```
gaul_data_sp <- readOGR(dsn = "./data/GAUL", layer = "g2015_2005_2")
```

```
OGR data source with driver: ESRI Shapefile
Source: "/Users/torres/Documents/02_working/3-Production/05_models/32_iiasa/global_n/data/GAUL", layer: "g2015_2005_2"
with 38189 features
It has 12 fields
```

```
gaul_lev0 <- levels(factor(gaul_data_sp@data[,"ADM0_NAME"]))

ids <- lapply(gaul_lev0, function(x) which(gaul_data_sp@data[,"ADM0_NAME"] == x))
```

Create a Data Frame and Rename Columns

- Rename the columns to indicate the country name and the aggregated wheat production in million tons.

Create a Data Frame and Rename Columns

- Rename the columns to indicate the country name and the aggregated wheat production in million tons.
- The country_production_df contains a data frame with each country's name and its aggregated wheat production in million tons. This data frame can be used for further analysis or visualization, and export to a csv file.

Wheat Production: Top 10 Most Productive Countries in 2005

```
load("./output/country_aggregated_production_lst.RData")

country_production_df = data.frame(do.call(what = rbind, args = country_aggregated_production_lst))

country_production_df$ID = gaul_lev0
```

```
colnames(country_production_df) <- c( "Country", "Wheat_Production_Mt")

top10_order = country_production_df[order(country_production_df$Wheat_Production_Mt, decreasing = TRUE), ]

top10_order[1:10,]
```

```
                     Country Wheat_Production_Mt
52                     China            99.26530
117                    India            69.55773
260 United States of America            55.12679
206       Russian Federation            46.02241
85                    France            37.27257
46                    Canada            25.26367
93                   Germany            23.81824
189                 Pakistan            20.89733
251                   Turkey            20.83224
16                 Australia            19.30162
```

```
write.csv(country_production_df, file = "output/country_production_wheat.csv")
```

Task 3

- To create a raster map of the nitrogen (N) output in harvested wheat yield, assuming that 2% of the harvested wheat yield consists of the N element, it is used again the "raster" package. Here are the steps to achieve this:

```r
library(raster)

wheat_production_raster = raster("output/wheat_production_Mt.tif")

nitrogen_output_raster = wheat_production_raster * 0.02
```

Task 3

- To create a raster map of the nitrogen (N) output in harvested wheat yield, assuming that 2% of the harvested wheat yield consists of the N element, it is used again the "raster" package. Here are the steps to achieve this:
  - Read the "wheat_production" raster created earlier in Task 1.

```
library(raster)

wheat_production_raster = raster("output/wheat_production_Mt.tif")

nitrogen_output_raster = wheat_production_raster * 0.02
```

Task 3

- To create a raster map of the nitrogen (N) output in harvested wheat yield, assuming that 2% of the harvested wheat yield consists of the N element, it is used again the "raster" package. Here are the steps to achieve this:
  - Read the "wheat_production" raster created earlier in Task 1.
  - Calculate Nitrogen Output Raster: A new raster that represents the nitrogen (N) output in harvested wheat yield is created. Assuming 2% of the harvested yield is N, the following formula is used:

```
library(raster)

wheat_production_raster = raster("output/wheat_production_Mt.tif")

nitrogen_output_raster = wheat_production_raster * 0.02
```

Task 3

- To create a raster map of the nitrogen (N) output in harvested wheat yield, assuming that 2% of the harvested wheat yield consists of the N element, it is used again the "raster" package. Here are the steps to achieve this:
  - Read the "wheat_production" raster created earlier in Task 1.
  - Calculate Nitrogen Output Raster: A new raster that represents the nitrogen (N) output in harvested wheat yield is created. Assuming 2% of the harvested yield is N, the following formula is used:
    - nitrogen_output_raster = wheat_production_raster * 0.02

```
library(raster)

wheat_production_raster = raster("output/wheat_production_Mt.tif")

nitrogen_output_raster = wheat_production_raster * 0.02
```

Task 3

- To create a raster map of the nitrogen (N) output in harvested wheat yield, assuming that 2% of the harvested wheat yield consists of the N element, it is used again the "raster" package. Here are the steps to achieve this:
  - Read the "wheat_production" raster created earlier in Task 1.
  - Calculate Nitrogen Output Raster: A new raster that represents the nitrogen (N) output in harvested wheat yield is created. Assuming 2% of the harvested yield is N, the following formula is used:
    - nitrogen_output_raster = wheat_production_raster * 0.02
  - Plot the Nitrogen Output Raster: Visualize the "nitrogen_output_raster" using the plot function from the "raster" package.

```
library(raster)

wheat_production_raster = raster("output/wheat_production_Mt.tif")

nitrogen_output_raster = wheat_production_raster * 0.02
```

Task 3

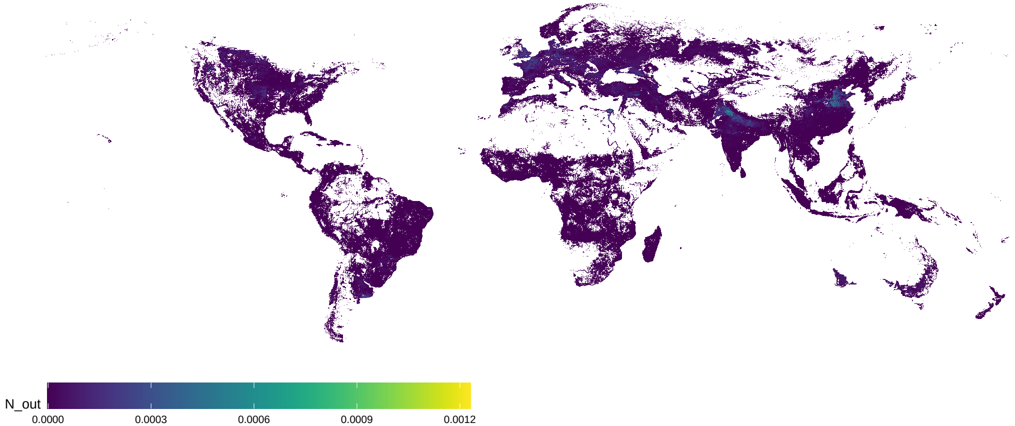
- To create a raster map of the nitrogen (N) output in harvested wheat yield, assuming that 2% of the harvested wheat yield consists of the N element, it is used again the "raster" package. Here are the steps to achieve this:
  - Read the "wheat_production" raster created earlier in Task 1.
  - Calculate Nitrogen Output Raster: A new raster that represents the nitrogen (N) output in harvested wheat yield is created. Assuming 2% of the harvested yield is N, the following formula is used:
    - nitrogen_output_raster = wheat_production_raster * 0.02
  - Plot the Nitrogen Output Raster: Visualize the "nitrogen_output_raster" using the plot function from the "raster" package.
  - Export the Nitrogen Output Raster: To export the raster map of nitrogen output, the writeRaster function is used and saved in GeoTiff format.

```
library(raster)

wheat_production_raster = raster("output/wheat_production_Mt.tif")

nitrogen_output_raster = wheat_production_raster * 0.02
```

Global Nitrogen Output

- A raster map that represents the nitrogen (N) output in the harvested wheat yield has been created, based on the assumption that 2% of the yield consists of the N element. This map will show the distribution of nitrogen output in million tons (Mt) across the Globe.

Global Nitrogen Output in Harvested Wheat Yield in Mt in 2005




```r
raster::writeRaster(nitrogen_output_raster, filename = "output/nitrogen_output.tif", format = "GTiff", overwrite = TRUE)
```

Task 4

- Using the dataset of country-level nitrogen use efficiency (NUE) of wheat from [Zhang et al., 2015], and steps from previous tasks:

Task 4

- Using the dataset of country-level nitrogen use efficiency (NUE) of wheat from [Zhang et al., 2015], and steps from previous tasks:

Task 4

- Using the dataset of country-level nitrogen use efficiency (NUE) of wheat from [Zhang et al., 2015], and steps from previous tasks:
  - a. For the 10 biggest wheat producers the country-level values of N output in harvested wheat, as well as related total N inputs and N losses (i.e., surplus) is estimated, and exported the dataset as a csv file

Task 4

- Using the dataset of country-level nitrogen use efficiency (NUE) of wheat from [Zhang et al., 2015], and steps from previous tasks:
  - a. For the 10 biggest wheat producers the country-level values of N output in harvested wheat, as well as related total N inputs and N losses (i.e., surplus) is estimated, and exported the dataset as a csv file

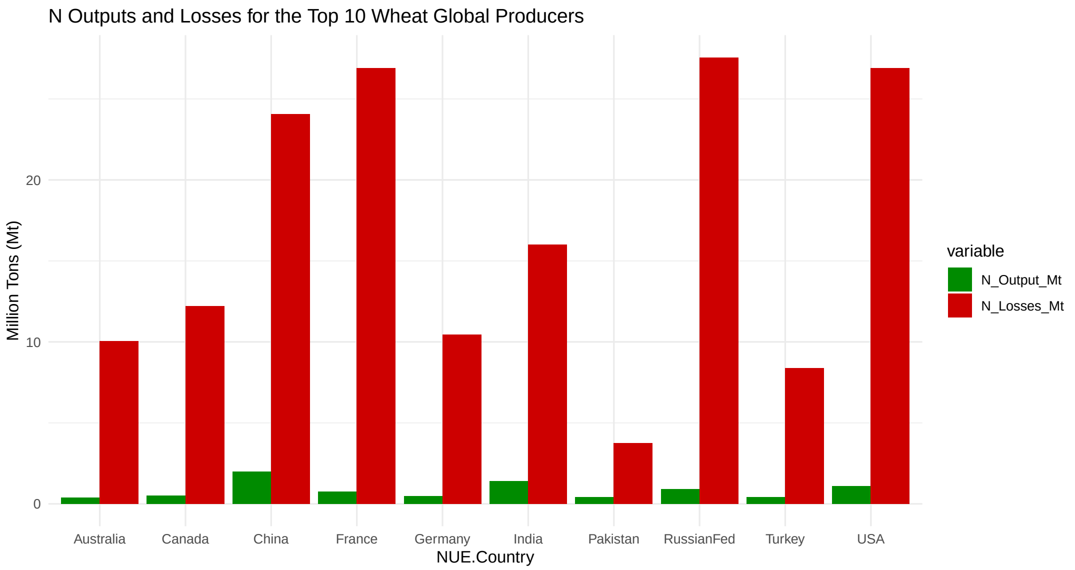
Task 4

- Using the dataset of country-level nitrogen use efficiency (NUE) of wheat from [Zhang et al., 2015], and steps from previous tasks:
  - a. For the 10 biggest wheat producers the country-level values of N output in harvested wheat, as well as related total N inputs and N losses (i.e., surplus) is estimated, and exported the dataset as a csv file
  - b. The N outputs and losses for these 10 countries are summarized in one figure (plot exported as pdf file)

a. Estimated and Exported Dataset for the Top 10 Countries by N Outputs and Losses

NUE.Country	NUE.NUE	Production_Mt	N_Output_Mt	N_Inputs_Mt	N_Losses_Mt
China	0.26	99.27	1.99	26.04	24.06
India	0.25	69.56	1.39	17.38	15.99
USA	0.51	55.13	1.10	28.00	26.90
RussianFed	0.62	46.02	0.92	28.48	27.56
France	0.74	37.27	0.75	27.65	26.91
Canada	0.50	25.26	0.51	12.72	12.21
Germany	0.46	23.82	0.48	10.92	10.45
Pakistan	0.20	20.90	0.42	4.18	3.76
Turkey	0.42	20.83	0.42	8.80	8.39
Australia	0.54	19.30	0.39	10.44	10.05

b. Visualization of the N Outputs and Losses



c. Main Patterns of N Lossess across Countries

- The main patterns of N losses across countries, in relation to production volume and NUE (including any singular feature) are explained in the following paragraph:

c. Main Patterns of N Lossess across Countries

- The main patterns of N losses across countries, in relation to production volume and NUE (including any singular feature) are explained in the following paragraph:
  - In the analysis of the top 10 wheat-producing countries, varying patterns of nitrogen losses is observed. Some countries with high wheat production volumes and relatively low NUE, such as China, showed significant nitrogen losses, suggesting inefficiencies in nitrogen utilization. In contrast, countries with relative higher NUE, like Australia, exhibited lower losses despite high production. Additionally, a few countries, like France, displayed unexpected patterns of high losses compared to their NUE, potentially indicating other factors influencing nitrogen loss, such as agricultural practices or environmental conditions.

Task 5

In the following paragraph is explained how an analysis like the one performed in previous tasks could translate to the models within BNR's modeling suite (https://iiasa.github.io/iBIOM/en/main/), including potential limitations.

- An analysis of nitrogen output, inputs, and losses in wheat production can inform IIASA BNR's modeling suite by providing critical data inputs for assessing the environmental impacts of agricultural practices. These data help in calibrating and validating models related to land use, nutrient management, and climate change mitigation, enhancing the suite's accuracy in predicting the effects of different agricultural scenarios on nitrogen cycling and environmental sustainability. Limitations, however, may arise from the simplifications made e.g. in assuming a fixed 2% nitrogen content in harvested yield, as actual values can vary. Additionally, model outcomes depend on the quality and comprehensiveness of input data, which can pose challenges in areas with limited data availability.

Task 6

Issues

- In the Task 2, an implementation of the country-level aggregation step for computing the Global wheat production per country in parallel was needed due to the high computational burden reached and avoid RAM issues. Therefore, parallel pre-computed RData objects were created to be loaded in this step to render this reproducible presentation in a reasonable short time (less than three minutes), i.e.

Task 6

Issues

- In the Task 2, an implementation of the country-level aggregation step for computing the Global wheat production per country in parallel was needed due to the high computational burden reached and avoid RAM issues. Therefore, parallel pre-computed RData objects were created to be loaded in this step to render this reproducible presentation in a reasonable short time (less than three minutes), i.e.
  - "./output/sp_join_lst.RData"

Task 6

Issues

- In the Task 2, an implementation of the country-level aggregation step for computing the Global wheat production per country in parallel was needed due to the high computational burden reached and avoid RAM issues. Therefore, parallel pre-computed RData objects were created to be loaded in this step to render this reproducible presentation in a reasonable short time (less than three minutes), i.e.
  - "./output/sp_join_lst.RData"
  - "./output/raster_lst.RData"

Task 6

Issues

- In the Task 2, an implementation of the country-level aggregation step for computing the Global wheat production per country in parallel was needed due to the high computational burden reached and avoid RAM issues. Therefore, parallel pre-computed RData objects were created to be loaded in this step to render this reproducible presentation in a reasonable short time (less than three minutes), i.e.
  - "./output/sp_join_lst.RData"
  - "./output/raster_lst.RData"
  - "./output/country_aggregated_production_lst.RData"

Task 6

Issues

- In the Task 2, an implementation of the country-level aggregation step for computing the Global wheat production per country in parallel was needed due to the high computational burden reached and avoid RAM issues. Therefore, parallel pre-computed RData objects were created to be loaded in this step to render this reproducible presentation in a reasonable short time (less than three minutes), i.e.
  - "./output/sp_join_lst.RData"
  - "./output/raster_lst.RData"
  - "./output/country_aggregated_production_lst.RData"
- Similarly in Task 3 the actual computation of the Global country production was coded in parallel and saved as an RData object to be loaded subsequently, i.e.

Task 6

Issues

- In the Task 2, an implementation of the country-level aggregation step for computing the Global wheat production per country in parallel was needed due to the high computational burden reached and avoid RAM issues. Therefore, parallel pre-computed RData objects were created to be loaded in this step to render this reproducible presentation in a reasonable short time (less than three minutes), i.e.
  - "./output/sp_join_lst.RData"
  - "./output/raster_lst.RData"
  - "./output/country_aggregated_production_lst.RData"
- Similarly in Task 3 the actual computation of the Global country production was coded in parallel and saved as an RData object to be loaded subsequently, i.e.
  - "./output/country_aggregated_N_lst.RData"

Task 6

Assumptions

- The computations were done by using the Geodetic CRS: WGS 84.

Task 6

Assumptions

- The computations were done by using the Geodetic CRS: WGS 84.
- The three input maps from the SPAM model for the year 2005 are global scale in raster format (5 arcminute spatial resolution):

Task 6

Assumptions

- The computations were done by using the Geodetic CRS: WGS 84.
- The three input maps from the SPAM model for the year 2005 are global scale in raster format (5 arcminute spatial resolution):
  - Estimates of yield in Kg/Ha,

Task 6

Assumptions

- The computations were done by using the Geodetic CRS: WGS 84.
- The three input maps from the SPAM model for the year 2005 are global scale in raster format (5 arcminute spatial resolution):
  - Estimates of yield in Kg/Ha,
  - Physical area in Ha,

Task 6

Assumptions

- The computations were done by using the Geodetic CRS: WGS 84.
- The three input maps from the SPAM model for the year 2005 are global scale in raster format (5 arcminute spatial resolution):
  - Estimates of yield in Kg/Ha,
  - Physical area in Ha,
  - Harvested area in Ha.

Task 6

Assumptions

- The computations were done by using the Geodetic CRS: WGS 84.
- The three input maps from the SPAM model for the year 2005 are global scale in raster format (5 arcminute spatial resolution):
  - Estimates of yield in Kg/Ha,
  - Physical area in Ha,
  - Harvested area in Ha.
- 1 ton is equal to 1,000 Kg, so 1 million ton is equal to 1,000,000,000 Kg, i.e. 1 million ton = 1e9 Kg.

References

Ulrike Wood-Sichra, Alison B Joglekar, and Liangzhi You. Spatial Production Allocation Model (SPAM) 2005: Technical Documentation. 2016.

Xin Zhang, Eric A. Davidson, Denise L. Mauzerall, Timothy D. Searchinger, Patrice Dumas, and Ye Shen. Managing nitrogen for sustainable development. *Nature*, 528 (7580):51–59, December 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature15743. URL https://www.nature.com/articles/nature15743.

Session Info

```
Time difference of 2.93719 mins

─ Session info ───────────────────────────────────────────────────────────────
 setting  value
 version  R version 4.2.0 (2022-04-22)
 os       macOS 13.4.1
 system   aarch64, darwin20
 ui       X11
 language (EN)
 collate  en_US.UTF-8
 ctype    en_US.UTF-8
 tz       Europe/Luxembourg
 date     2023-10-18
 pandoc   3.1.8 @ /opt/homebrew/bin/ (via rmarkdown)

─ Packages ───────────────────────────────────────────────────────────────────
 package     * version date (UTC) lib source
 class         7.3-21  2023-01-23 [2] CRAN (R 4.2.0)
 classInt      0.4-9   2023-02-28 [1] CRAN (R 4.2.0)
 cli           3.6.0   2023-01-09 [1] CRAN (R 4.2.0)
 codetools     0.2-19  2023-02-01 [2] CRAN (R 4.2.0)
 colorspace    2.1-0   2023-01-23 [1] CRAN (R 4.2.0)
 DBI           1.1.3   2022-06-18 [1] CRAN (R 4.2.0)
 digest        0.6.31  2022-12-11 [1] CRAN (R 4.2.0)
 doParallel  * 1.0.17  2022-02-07 [1] CRAN (R 4.2.0)
 dplyr         1.1.3   2023-09-03 [1] CRAN (R 4.2.0)
 e1071         1.7-13  2023-02-01 [1] CRAN (R 4.2.0)
 evaluate      0.20    2023-01-17 [1] CRAN (R 4.2.0)
 fansi         1.0.4   2023-01-22 [1] CRAN (R 4.2.0)
 farver        2.1.1   2022-07-06 [1] CRAN (R 4.2.0)
 fastmap       1.1.1   2023-02-24 [1] CRAN (R 4.2.0)
 foreach     * 1.5.2   2022-02-02 [1] CRAN (R 4.2.0)
 generics      0.1.3   2022-07-05 [1] CRAN (R 4.2.0)
 ggplot2     * 3.4.1   2023-02-10 [1] CRAN (R 4.2.0)
```