BIODATABASE: Loading reference tables

Manu Chassot

30 January 2020

```
### Extend Java memory for XLConnect
#options(java.parameters = "-Xmx1024m")
options(java.parameters = "-Xmx2048m")
#options(java.parameters = "-Xmx4g" )

### Install/load devtools
if(!require(devtools)){
  install.packages("devtools", repos = "https://pbil.univ-lyon1.fr/CRAN/")
  suppressPackageStartupMessages(library(devtools,quietly = TRUE))
}

### Install/load lubripack
if(!require(lubripack)){
  install_github("espanta/lubripack")
  suppressPackageStartupMessages(library(lubripack,quietly = TRUE))
}

### Install/load libraries required for analysis
lubripack('RPostgreSQL','knitr','lubridate','XLConnect',silent = FALSE)
```

Connect to the database

```
drv <- dbDriver("PostgreSQL")
con_emotion3_local <- dbConnect(drv, user = "postgres", dbname = "emotion3", host = "localhost")
```

Read metadata from spreadsheet

```
DDD <- XLConnect::loadWorkbook("../XLS/DDD_Database.xls", create = FALSE)
```

1- Anchored Fish Aggregating Devices

```
afad <- readWorksheet(DDD, sheet = "AFAD")

### Save in temp to allow for insertion in the database
write.table(afad, file = "/tmp/afad.csv", row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.afad
send.references.tables.afad <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.afad FROM '/tmp/afad.csv' WITH DELIMITER E'\
```

2- Amino-acids list

```
amino_acids <- readWorksheet(DDD, "AMINO_ACIDS")

### Save in temp to allow for insertion in the database
write.table(amino_acids, file = "/tmp/amino_acids.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.amino_acids
send.references.tables.amino_acids <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.amino_acids FROM '/tmp/amino_acids.cs
```

3- Analysis groups

```
analysis <- readWorksheet(DDD, "ANALYSIS")

### Save in temp to allow for insertion in the database
analyis_groups <- unique(analysis[, c("analysis_group", "desc_analysis_group")])
```

```r
analyis_groups <- analyis_groups[order(analyis_groups$analysis_group), ]
write.table(analyis_groups, file = "/tmp/analysis_groups.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.analysis_groups
send.references.tables.analyis_groups <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.analysis_groups FROM '/tmp/analysi
```

4- Analysis laboratories

```r
### Extract spreadsheet ANALYSIS_LAB
analysis_lab <- readWorksheet(DDD, "ANALYSIS_LAB")

### Save in temp to allow for insertion in the database
write.table(analysis_lab, file = "/tmp/analysis_lab.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.analysis_lab
send.references.tables.analyis_lab <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.analysis_lab FROM '/tmp/analysis_lab.
```

5- Matching between groups of analysis and analysis types

```r
analysis_matching_groups <- unique(analysis[, c("analysis_group", "analysis")])
names(analysis_matching_groups)[2] <- "analysis_type"

### Save in temp to allow for insertion in the database
write.table(analysis_matching_groups, file = "/tmp/analysis_matching_groups.csv",
    row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.analysis_matching_groups
send.references.tables.analyis_matching_groups <- dbSendQuery(con_emotion3_local,
    paste0("COPY references_tables.analysis_matching_groups FROM '/tmp/analysis_matching_groups.csv' WITH DELIMITER E'\\t' CSV HEADER"))
```

6- Analysis modes

```
analysis_modes <- readWorksheet(DDD, "ANALYSIS_MODE")

### Save in temp to allow for insertion in the database
write.table(analysis_modes, file = "/tmp/analysis_modes.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.analysis_modes
send.references.tables.analyis_modes <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.analysis_modes FROM '/tmp/analysis_
```

7- Analysis replicates

```
analysis_replicate <- readWorksheet(DDD, "ANALYSIS_REPLICATE")

### Save in temp to allow for insertion in the database
write.table(analysis_replicate, file = "/tmp/analysis_replicate.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.analysis_replicate
send.references.tables.analyis_replicate <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.analysis_replicate FROM '/tmp/a
```

8- Analysis sample description

```
analysis_sample_description <- readWorksheet(DDD, "ANALYSIS_SAMP_DESCRIPTION")

### Save in temp to allow for insertion in the database
write.table(analysis_sample_description, file = "/tmp/analysis_sample_description.csv",
    row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.analysis_sample_description
send.references.tables.analyis_sample_description <- dbSendQuery(con_emotion3_local,
    paste0("COPY references_tables.analysis_sample_description FROM '/tmp/analysis_sample_description.csv' WITH DELIMITER E'\\t' CSV HEADE
```

9- Analysis types

```
analysis_types <- unique(analysis[, c("analysis", "desc_analysis")])
### Extract each component of analysis$desc_analysis from cut at '\n' and get
### first element
analysis_types$desc_analysis <- unlist(lapply(strsplit(analysis_types$desc_analysis,
    split = "\n"), function(l) l[[1]]))

### Save in temp to allow for insertion in the database
write.table(analysis_types, file = "/tmp/analysis_types.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.analysis_types
send.references.tables.analyis_types <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.analysis_types FROM '/tmp/analysis_
```

10- Atresia stages

```
atresia <- readWorksheet(DDD, "ATRESIA")

### Save in temp to allow for insertion in the database
write.table(atresia, file = "/tmp/atresia.csv", row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.atresia
send.references.tables.atresia <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.atresia FROM '/tmp/atresia.csv' WITH DELI
```

11- Certified reference materials

```
crm <- readWorksheet(DDD, "CRM")

### Save in temp to allow for insertion in the database
write.table(crm, file = "/tmp/crm.csv", row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.crm
```

```
send.references.tables.crm <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.crm FROM '/tmp/crm.csv' WITH DELIMITER E'\\t'
```

12- Derivatization modes

```
derivatization_mode <- readWorksheet(DDD, "DERIVATIZATION_MODE")

### Save in temp to allow for insertion in the database
write.table(derivatization_mode, file = "/tmp/derivatization_mode.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.vessel_storage
send.references.tables.derivatization_mode <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.derivatization_mode FROM '/tm
```

13- Drying modes

```
drying_mode <- readWorksheet(DDD, "DRYING_MODE")

### Save in temp to allow for insertion in the database
write.table(drying_mode, file = "/tmp/drying_mode.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.atresia
send.references.tables.drying_mode <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.drying_mode FROM '/tmp/drying_mode.cs
```

14- Extraction modes

```
extraction_mode <- readWorksheet(DDD, "EXTRACTION_MODE")

### Save in temp to allow for insertion in the database
write.table(extraction_mode, file = "/tmp/extraction_mode.csv", row.names = FALSE,
    sep = "\t", na = "")
```

```
### INSERT references_tables.extraction_mode
send.references.tables.extraction_mode <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.extraction_mode FROM '/tmp/extrac
```

15- Fatty acids list

```
fatty_acids <- readWorksheet(DDD, "FATTY_ACIDS")

### Save in temp to allow for insertion in the database
write.table(fatty_acids, file = "/tmp/fatty_acids.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.fatty_acids
send.references.tables.fatty_acids <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.fatty_acids FROM '/tmp/fatty_acids.cs
```

16- Fishing modes

```
fishing_mode <- readWorksheet(DDD, "FISHING_MODE")

### Save in temp to allow for insertion in the database
write.table(fishing_mode, file = "/tmp/fishing_mode.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.fishing_mode
send.references.tables.fishing_mode <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.fishing_mode FROM '/tmp/fishing_mode
```

17- Fishing gears

```
gear <- readWorksheet(DDD, "GEAR")

### Save in temp to allow for insertion in the database
write.table(gear, file = "/tmp/gear.csv", row.names = FALSE, sep = "\t", na = "")
```

```
### INSERT references_tables.gear
send.references.tables.gear <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.gear FROM '/tmp/gear.csv' WITH DELIMITER E'\
```

18- Grinding modes

```
grinding_mode <- readWorksheet(DDD, "GRINDING_MODE")

### Save in temp to allow for insertion in the database
write.table(grinding_mode, file = "/tmp/grinding_mode.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.grinding_mode
send.references.tables.grinding_mode <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.grinding_mode FROM '/tmp/grinding_m
```

19- Landing sites

```
landing <- readWorksheet(DDD, "LANDING")

### Save in temp to allow for insertion in the database
write.table(landing, file = "/tmp/landing.csv", row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.landing
send.references.tables.landing <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.landing FROM '/tmp/landing.csv' WITH DELI
```

20- Macroscopic maturity stages [visual exam]

```
macro_maturity <- readWorksheet(DDD, "MACRO_MATURITY")

### Save in temp to allow for insertion in the database
write.table(macro_maturity, file = "/tmp/macro_maturity.csv", row.names = FALSE,
    sep = "\t", na = "")
```

```
### INSERT references_tables.macro_maturity
send.references.tables.macro_maturity <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.macro_maturity FROM '/tmp/macro_ma
```

21- Microscopic maturity stages [histology]

```
micro_maturity_stage <- readWorksheet(DDD, "MICRO_MATURITY")

### Save in temp to allow for insertion in the database
write.table(micro_maturity_stage, file = "/tmp/micro_maturity_stage.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.micro_maturity
send.references.tables.micro_maturity_stage <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.micro_maturity_stage FROM '/
```

22- Minerals

```
minerals <- readWorksheet(DDD, "MINERALS")

### Save in temp to allow for insertion in the database
write.table(minerals, file = "/tmp/minerals.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.micro_maturity
send.references.tables.minerals <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.minerals FROM '/tmp/minerals.csv' WITH D
```

23- Oceans

```
ocean <- readWorksheet(DDD, "OCEAN")

### Save in temp to allow for insertion in the database
write.table(ocean, file = "/tmp/ocean.csv", row.names = FALSE, sep = "\t", na = "")
```

```
### INSERT references_tables.ocean
send.references.tables.ocean <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.ocean FROM '/tmp/ocean.csv' WITH DELIMITER
```

24- Operators

```
### Extract spreadsheet
operator <- readWorksheet(DDD, "OPERATOR")

### Remove second affiliation Temporary: Each affiliation should be added in the
### table
operator <- operator[, c("1_operator", "Affiliation1")]

### Save in temp to allow for insertion in the database
write.table(operator, file = "/tmp/operator.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.operator
send.references.tables.operator <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.operator FROM '/tmp/operator.csv' WITH D
```

25- Organic contaminants

```
organic_contaminants <- readWorksheet(DDD, "ORGANIC_CONTAMINANTS")

### Save in temp to allow for insertion in the database
write.table(organic_contaminants, file = "/tmp/organic_contaminants.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.vessel_storage
send.references.tables.organic_contaminants <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.organic_contaminants FROM '/
```

26- Otolith measurement types

```
otolith_measurement_type <- readWorksheet(DDD, "OTOLITHS")

### Save in temp to allow for insertion in the database
write.table(otolith_measurement_type, file = "/tmp/otolith_measurement_type.csv",
    row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.otolith_measurement_type
send.references.tables.otolith_measurement_type <- dbSendQuery(con_emotion3_local,
    paste0("COPY references_tables.otolith_measurement_type FROM '/tmp/otolith_measurement_type.csv' WITH DELIMITER E'\\t' CSV HEADER"))
```

27- Packaging types

```
packaging <- readWorksheet(DDD, "PACKAGING")

### Save in temp to allow for insertion in the database
write.table(packaging, file = "/tmp/packaging.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.packaging
send.references.tables.packaging <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.packaging FROM '/tmp/packaging.csv' WIT
```

28- Post-ovulatory follicles classification

```
pof <- readWorksheet(DDD, "POF")

### Save in temp to allow for insertion in the database
write.table(pof, file = "/tmp/pof.csv", row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.pof
send.references.tables.pof <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.pof FROM '/tmp/pof.csv' WITH DELIMITER E'\\t'
```

29- Prey groups

```
prey_groups <- readWorksheet(DDD, "PREY_GROUPS")

### Save in temp to allow for insertion in the database
write.table(prey_groups, file = "/tmp/prey_groups.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.prey_groups
send.references.tables.prey_groups <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.prey_groups FROM '/tmp/prey_groups.cs
```

30- Processing replicates

```
processing_replicates <- readWorksheet(DDD, "PROCESSING_REPLICATE")

### Save in temp to allow for insertion in the database
write.table(processing_replicates, file = "/tmp/processing_replicates.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.processing_replicate
send.references.tables.processing_replicates <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.processing_replicates FROM
```

31- Projects

```
project <- readWorksheet(DDD, "PROJECT")

### Save in temp to allow for insertion in the database
write.table(project, file = "/tmp/project.csv", row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.project
send.references.tables.project <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.project FROM '/tmp/project.csv' WITH DELI
```

32- Sample positions

```
sample_position <- readWorksheet(DDD, "SAMPLE_POSITION")

### Save in temp to allow for insertion in the database
write.table(sample_position, file = "/tmp/sample_position.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.sample_position
send.references.tables.sample_position <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.sample_position FROM '/tmp/sample
```

33- Sampling platforms

```
sampling_platform <- readWorksheet(DDD, "SAMPLING_PLATFORM")

### Save in temp to allow for insertion in the database
write.table(sampling_platform, file = "/tmp/sampling_platform.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.sampling_platform
send.references.tables.sampling_platform <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.sampling_platform FROM '/tmp/sa
```

34- Sex classification

```
sex <- readWorksheet(DDD, "SEX")

### Save in temp to allow for insertion in the database
write.table(sex, file = "/tmp/sex.csv", row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.sex
send.references.tables.sex <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.sex FROM '/tmp/sex.csv' WITH DELIMITER E'\\t'
```

35- Species list

```
### Caution: Do not include columns 'SFA.ID.ppt' & 'SFA_life_history_table'
species <- readWorksheet(DDD, "SPECIES")[, 1:17]

### Save in temp to allow for insertion in the database
write.table(species, file = "/tmp/species.csv", row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.species
send.references.tables.species <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.species FROM '/tmp/species.csv' WITH DELI
```

36- Stomach fullness classification

```
stomach_fullness <- readWorksheet(DDD, "STOMACH_FULLNESS")

### Save in temp to allow for insertion in the database
write.table(stomach_fullness, file = "/tmp/stomach_fullness.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.stomach_fullness
send.references.tables.stomach_fullness <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.stomach_fullness FROM '/tmp/stom
```

37- Storage modes

```
storage_mode <- readWorksheet(DDD, "STORAGE_MODE")

### Save in temp to allow for insertion in the database
write.table(storage_mode, file = "/tmp/storage_mode.csv", row.names = FALSE, sep = "\t",
    na = "")

### INSERT references_tables.storage_mode
send.references.tables.storage_mode <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.storage_mode FROM '/tmp/storage_mode
```

38- Tissues

```
tissue <- readWorksheet(DDD, "TISSUE")

### Save in temp to allow for insertion in the database
write.table(tissue, file = "/tmp/tissue.csv", row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.tissue
send.references.tables.tissue <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.tissue FROM '/tmp/tissue.csv' WITH DELIMIT
```

39- Vessels

```
vessel <- readWorksheet(DDD, "VESSEL")

### Save in temp to allow for insertion in the database
write.table(vessel, file = "/tmp/vessel.csv", row.names = FALSE, sep = "\t", na = "")

### INSERT references_tables.vessel
send.references.tables.vessel <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.vessel FROM '/tmp/vessel.csv' WITH DELIMIT
```

40- Vessel storage classification

```
### Extract spreadsheet
vessel_storage <- readWorksheet(DDD, "VESSEL_STORAGE")

### Save in temp to allow for insertion in the database
write.table(vessel_storage, file = "/tmp/vessel_storage.csv", row.names = FALSE,
    sep = "\t", na = "")

### INSERT references_tables.vessel_storage
send.references.tables.vessel_storage <- dbSendQuery(con_emotion3_local, paste0("COPY references_tables.vessel_storage FROM '/tmp/vessel_s
```