Taxonomic harmonization: workflows

Emilio Berti

6/15/2021

Contents

Number of unique species in BioTIME 1
Comparison Bogota - Torino 4
Comparison Torino - GBIF 9

Number of unique species in BioTIME

BioTIME taken as raw file had 44,326 unique taxa. After passing it through `rgnparser` (*gn_parse_tidy()*), 4,734 taxa (11%) were duplicates. Of the remaining 39,592 taxa, 6,692 did not have *Genus species* nomenclature and were removed. Importantly, the remaining 32,900 taxa did not consist exclusively of *Genus species* taxa, but it was not uncommon to have common names and taxonomic keywords such as *Family fam.* We proceeded with the three workflow (Bogota, Torino, and GBIF only) with the remaining 32,900 taxa that had at least two words in their names, a necessary condition for the taxa to be identified at the species level.

```
library(tidyverse)

biotime <- read_csv("~/Documents/biotime_common.csv")
message("BioTIME raw number of unique taxa: ",
        length(unique(biotime$BioTIME)))
biotime %>%
  mutate(species_level = modify(parsed, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  distinct(parsed, .keep_all = TRUE) %>%
  filter(species_level) %>%
  select(-species_level)
```

```
## # A tibble: 32,900 x 5
##    BioTIME              parsed               class   phylum     common
##    <chr>                <chr>                <chr>   <chr>      <chr>
##  1 Abagrotis apposita   Abagrotis apposita   Insecta Arthropoda <NA>
##  2 Abagrotis baueri     Abagrotis baueri     Insecta Arthropoda <NA>
##  3 Abagrotis erratica   Abagrotis erratica   Insecta Arthropoda <NA>
##  4 Abagrotis forbesi    Abagrotis forbesi    Insecta Arthropoda <NA>
##  5 Abagrotis glenni     Abagrotis glenni     Insecta Arthropoda <NA>
##  6 Abagrotis nefascia   Abagrotis nefascia   Insecta Arthropoda <NA>
##  7 Abagrotis placida    Abagrotis placida    Insecta Arthropoda <NA>
##  8 Abagrotis pulchrata  Abagrotis pulchrata  Insecta Arthropoda <NA>
##  9 Abagrotis reedi     Abagrotis reedi     Insecta Arthropoda <NA>
## 10 Abagrotis scopeops  Abagrotis scopeops  Insecta Arthropoda <NA>
## # ... with 32,890 more rows
```

```
message("Parsed unique number of taxa: ",
        length(unique(biotime$parsed)))
diff_parsed <- length(unique(biotime$BioTIME)) - length(unique(biotime$parsed))
```

```
# workflow 1 -------
plants <- read_csv("~/Documents/bogota_lcvp.csv")
fishes <- read_csv("~/Documents/bogota_fishbase.csv")
birds <- read_csv("~/Documents/bogota_ebird.csv")
gbif <- read_csv("~/Documents/bogota_gbif.csv")
wf1 <- biotime %>%
  select(-class, -phylum, -BioTIME) %>%
  distinct_all() %>%
  left_join(plants %>% distinct_all()) %>%
  left_join(fishes %>% distinct_all()) %>%
  left_join(birds %>% distinct_all()) %>%
  left_join(gbif %>% distinct_all())
#remove GBIF if another db found something
wf1 <- wf1 %>%
  mutate(remove_gbif = pmap(list(lcvp, fishbase, ebird),
                            function(x, y, z) {
                              valid <- !is.na(c(x, y, z))
                              if (any(valid))
                                TRUE
                              else
                                FALSE
                            }) %>% unlist() %>% as.logical()) %>%
  mutate(gbif = modify2(gbif, remove_gbif, function(x, y) {
    if (y)
      NA
    else
      x
  })) %>%
  select(-remove_gbif)
wf1 <- wf1 %>%
  mutate(conflict = pmap(list(lcvp, fishbase, ebird),
                         function(x, y, z) {
                           valid <- !is.na(c(x, y, z))
                           if (sum(valid) > 1)
                             TRUE
                           else
                             FALSE
                         }) %>% unlist() %>% as.logical()) %>%
  filter(!conflict) %>%
  select(-conflict)
wf1 <- wf1 %>%
  mutate(species_level = modify(parsed, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  distinct(parsed, .keep_all = TRUE) %>%
  filter(species_level) %>%
  select(-species_level)
wf1 <- wf1 %>%
  select(-common) %>%
  pivot_longer(cols = 2:5,
               names_to = "step",
               values_to = "matched") %>%
  filter(!is.na(matched)) %>%
  select(-step) %>%
  mutate(species_level = modify(matched, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  distinct(parsed, .keep_all = TRUE) %>%
  filter(species_level) %>%
  select(-species_level)
```

```
# workflow 2 ----------
plants <- read_csv("~/Documents/torino_lcvp.csv")
fishes <- read_csv("~/Documents/torino_fishbase.csv")
birds <- read_csv("~/Documents/torino_ebird.csv")
gbif <- read_csv("~/Documents/torino_gbif.csv") %>%
  mutate(species_level = modify(gbif, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  filter(species_level) %>%
  select(-species_level) %>%
  mutate(species_level = modify(parsed, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  filter(species_level) %>%
  select(-species_level)
wf2 <- biotime %>%
  select(-class, -phylum, -BioTIME) %>%
  distinct_all() %>%
  left_join(plants %>% distinct_all()) %>%
  left_join(fishes %>% distinct_all()) %>%
  left_join(birds %>% distinct_all()) %>%
  left_join(gbif %>% distinct_all()) %>%
  mutate(species_level = modify(parsed, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  distinct(parsed, .keep_all = TRUE) %>%
  filter(species_level) %>%
  select(-species_level)
wf2 <- wf2 %>%
  select(-common) %>%
  pivot_longer(cols = 2:5,
               names_to = "step",
               values_to = "matched") %>%
  filter(!is.na(matched)) %>%
  select(-step) %>%
  mutate(species_level = modify(matched, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  distinct(parsed, .keep_all = TRUE) %>%
  filter(species_level) %>%
  select(-species_level)
```

```
gbif <- read_csv("~/Documents/bogota_gbif.csv") %>%
  distinct(parsed, .keep_all = TRUE) %>%
  mutate(species_level = modify(parsed, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  filter(species_level) %>%
  select(-species_level) %>%
  mutate(species_level = modify(gbif, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  distinct(parsed, .keep_all = TRUE) %>%
  filter(species_level) %>%
  select(-species_level)
```

Comparison Bogota - Torino

Bogota workflow found 636 more than Torino, with Torino finding only 1 species more than Bogota.

```
wf1 %>%
  full_join(wf2, by = "parsed", suffix = c("_bogota", "_torino")) %>%
  mutate(matched_bogota = ifelse(is.na(matched_bogota), "NA", matched_bogota),
         matched_torino = ifelse(is.na(matched_torino), "NA", matched_torino)) %>%
  filter(matched_bogota != matched_torino) %>%
  pivot_longer(cols = 2:3, names_to = "workflow", values_to = "matches") %>%
  filter(matches == "NA") %>%
  group_by(workflow) %>%
  tally(name = "missing but found in the other workflow")
```

```
## # A tibble: 2 x 2
##   workflow       `missing but found in the other workflow`
##   <chr>                                              <int>
## 1 matched_bogota                                         1
## 2 matched_torino                                       636
```

If we inspect where these mis-matches come from, we find that they are mostly in birds, fishes, and plants. As these categories are identified by GBIF, Torino will pass them to the appropriate taxa-specific reference. In Bogota, instead, they are all passed against all taxa-specific references and, if not found, to GBIF. As such, the majority of these mis-matches comes from Bogota using GBIF taxonomy for taxa that should have been identified by taxa-specific references or left umatched. Bogota is likely mixing taxonomies, and there isn't much it can be done about it. The only thing is to remove species names from GBIF when they should have been obtained from a taxa-specific source.

```
wf1 %>%
  full_join(wf2, by = "parsed", suffix = c("_bogota", "_torino")) %>%
  left_join(biotime %>% select(parsed, common)) %>%
  mutate(matched_bogota = ifelse(is.na(matched_bogota), "NA", matched_bogota),
         matched_torino = ifelse(is.na(matched_torino), "NA", matched_torino)) %>%
  filter(matched_bogota != matched_torino) %>%
  pivot_longer(cols = 2:3, names_to = "workflow", values_to = "matches") %>%
  filter(matches == "NA") %>%
  distinct_all() %>%
  group_by(workflow, common) %>%
  tally(name = "missing but found in the other workflow")
```

```
## # A tibble: 5 x 3
## # Groups:   workflow [2]
##   workflow       common          `missing but found in the other workflow`
##   <chr>          <chr>                                               <int>
## 1 matched_bogota <NA>                                                    1
## 2 matched_torino birds                                                 235
## 3 matched_torino fishes                                                178
## 4 matched_torino vascular plants                                       207
## 5 matched_torino <NA>                                                   16
```

If we exclude issues with mixing taxonomies in Bogota, the difference between the two workflows is minimal, namely 27 species. By inspecting these, it is evident that most of them are vascular plants, which are not identified by GBIF as such and hence not passed to `LCVP` in Torino. In fact, the parsed species names are not found in GBIF, which explains the differences between Bogota and Torino. Overall, GBIF correctly identified the higher taxonomic group of 11,899 taxa out of a total of 11,926 (99.77%).

```
wf1 %>%
  full_join(wf2, by = "parsed", suffix = c("_bogota", "_torino")) %>%
  left_join(biotime %>% select(parsed, common)) %>%
  mutate(matched_bogota = ifelse(is.na(matched_bogota), "NA", matched_bogota),
         matched_torino = ifelse(is.na(matched_torino), "NA", matched_torino)) %>%
  filter(matched_bogota != matched_torino) %>%
  filter(is.na(common))
```

```
## # A tibble: 27 x 4
##    parsed            matched_bogota                       matched_torino common
##    <chr>             <chr>                                <chr>          <chr>
##  1 Aglaia ridleyi    Aglaia oligophylla Miq.              NA             <NA>
##  2 Aglaia rufa       Aglaia rufibarbis Ridl.              NA             <NA>
##  3 Arenaria lychnid~ Eremogone capillaris (Poir.) Fenzl   NA             <NA>
##  4 Arenaria stricta  Sabulina macra (A.Nelson & J.F.Macbr~ NA             <NA>
##  5 Arthrophyllum di~ Polyscias biformis (Philipson) Lowry~ NA             <NA>
##  6 Atylus tridens    Isopogon tridens (Meisn.) F.Muell.   Atylus tridens <NA>
##  7 Crepis longipes   Youngia longipes (Hemsl.) Babc. & St~ Crepis longip~ <NA>
##  8 Eugenia filiform~ Eugenia confusa DC.                  NA             <NA>
##  9 Eugenia rugosa    Eugenia patens Poir.                 NA             <NA>
## 10 Eulalia aurea     Eulalia aurea (Bory) Kunth           NA             <NA>
## # ... with 17 more rows
```

For Bogota, we removed taxa-specific names found in GBIF using the same step as in Torino.

```
plants <- read_csv("~/Documents/bogota_lcvp.csv")
fishes <- read_csv("~/Documents/bogota_fishbase.csv")
birds <- read_csv("~/Documents/bogota_ebird.csv")
gbif <- read_csv("~/Documents/bogota_gbif.csv")
wf1 <- biotime %>%
  select(-class, -phylum, -BioTIME) %>%
  distinct_all() %>%
  left_join(plants %>% distinct_all()) %>%
  left_join(fishes %>% distinct_all()) %>%
  left_join(birds %>% distinct_all()) %>%
  left_join(gbif %>% distinct_all())
#remove GBIF if another db found something
wf1 <- wf1 %>%
  mutate(remove_gbif = pmap(list(lcvp, fishbase, ebird),
                            function(x, y, z) {
                              valid <- any(!is.na(c(x, y, z)))
                              if (any(valid))
                                TRUE
                              else
                                FALSE
                            }) %>% unlist() %>% as.logical()) %>%
  mutate(gbif = modify2(gbif, remove_gbif, function(x, y) {
    if (y)
      NA
    else
      x
  })) %>%
  select(-remove_gbif)
wf1 <- wf1 %>%
  mutate(conflict = pmap(list(lcvp, fishbase, ebird),
                         function(x, y, z) {
                           valid <- !is.na(c(x, y, z))
                           if (sum(valid) > 1)
                             TRUE
                           else
                             FALSE
```

```r
                         }) %>% unlist() %>% as.logical()) %>%
  filter(!conflict) %>%
  select(-conflict)
wf1 <- wf1 %>%
  mutate(species_level = modify(parsed, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  distinct(parsed, .keep_all = TRUE) %>%
  filter(species_level) %>%
  select(-species_level)
# new step
wf1 <- wf1 %>% mutate(gbif = modify2(common, gbif, function(x, y) {
  if (x %in% c("vascular plants", "birds", "fishes"))
    NA
  else
    y
}))
# as usual
wf1 <- wf1 %>%
  select(-common) %>%
  pivot_longer(cols = 2:5,
               names_to = "step",
               values_to = "matched") %>%
  filter(!is.na(matched)) %>%
  select(-step) %>%
  mutate(species_level = modify(matched, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  distinct(parsed, .keep_all = TRUE) %>%
  filter(species_level) %>%
  select(-species_level)

# reload GBIF only
gbif <- read_csv("~/Documents/bogota_gbif.csv") %>%
  distinct(parsed, .keep_all = TRUE) %>%
  mutate(species_level = modify(parsed, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  filter(species_level) %>%
  select(-species_level) %>%
  mutate(species_level = modify(gbif, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  distinct(parsed, .keep_all = TRUE) %>%
  filter(species_level) %>%
  select(-species_level)
```

Bogota identified 30,628 species of the total 32,900. Torino identified 30,613. The differences are minimal and the two workflows can be used interchangebly (**if cleaning Bogota after to avoid mixing taxonomies**). In the next table, the difference is only in the number of species matched in *NA*; these species names refer, however, to (mostly) plants, which are incorrectly classified by GBIF in the first step of Torino. Overall, if one is interested in a marginal increase in accuracy, Bogota may be recommended, while if one is interested in computational speed, Torino would be preferred.

```
wf1 %>%
  left_join(biotime) %>%
  select(-class, -phylum) %>%
  mutate(common = ifelse(is.na(common), "NA", common)) %>%
  group_by(common) %>%
  tally(name = "Bogota matched") %>%
  mutate(`Bogota cumulative` = cumsum(`Bogota matched`)) %>%
  left_join(
    wf2 %>%
      left_join(biotime) %>%
      select(-class, -phylum) %>%
      mutate(common = ifelse(is.na(common), "NA", common)) %>%
      group_by(common) %>%
      tally(name = "Torino matched") %>%
      mutate(`Torino cumulative` = cumsum(`Torino matched`))
  ) %>%
  mutate(total = 32900)
```

```
## # A tibble: 5 x 6
##   common    `Bogota matched` `Bogota cumulati~ `Torino matched` `Torino cumulat~
##   <chr>                <int>             <int>            <int>            <int>
## 1 birds                  877               877              877              877
## 2 fishes                5413              6290             5413             6290
## 3 mammals                289              6579              289             6579
## 4 NA                   19504             26083            19489            26068
## 5 vascular~             4545             30628             4545            30613
## # ... with 1 more variable: total <dbl>
```

```
message("Number of unique taxa in Bogota: ",
        length(unique(wf1$matched)))
diff_bogota <- length(unique(wf1$parsed)) - length(unique(wf1$matched))
message("Number of unique taxa in Torino: ",
        length(unique(wf2$matched)))
diff_torino <- length(unique(wf2$parsed)) - length(unique(wf2$matched))
```

As the two workflow are mostly identical, we will focus for simplicity on Torino from now on. In Torino, around 77% of birds, 96% of fishes, and 95% of vascular plants were correctly identified by using `rebird`, `rfishbase`, and `lcvplants` R packages, respectively. For the other taxa, we used `rgbif` and identified 93%

of the species names.

```
biotime %>%
  mutate(species_level = modify(parsed, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  filter(species_level) %>%
  select(-species_level) %>%
  select(parsed, common) %>%
  left_join(wf2) %>%
  mutate(matched = ifelse(is.na(matched), "Non matched", "Matched")) %>%
  group_by(common, matched) %>%
  tally() %>%
  pivot_wider(names_from = "matched", values_from = n) %>%
  mutate(frac = round(Matched / (Matched + `Non matched`), 2))
```

```
## # A tibble: 5 x 4
## # Groups:   common [5]
##   common          Matched `Non matched`  frac
##   <chr>             <int>         <int> <dbl>
## 1 birds               877           267  0.77
## 2 fishes             5413           253  0.96
## 3 mammals             289             5  0.98
## 4 vascular plants    4545           257  0.95
## 5 <NA>              19489          1514  0.93
```

Comparison Torino - GBIF

We compare now how using only GBIF differ from the Torino workflow. As Bogota is very similar to Torino, there will not be many differences if using it instead of Torino here. However, as Bogota is more complex to work with, as results need to be properly cleaned and it takes longer time, I focused here on Torino only.

Torino and GBIF only differ in 1,837 species names, 624 of which were species beloning to plants, birds, or fishes for which an accepted name was not found in the taxa-specific references.

```
message("Number of unique taxa in GBIF only: ",
        length(unique(gbif$gbif)))
gbif <- gbif %>%
  mutate(species_level = modify(parsed, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  filter(species_level) %>%
  select(-species_level) %>%
  mutate(species_level = modify(gbif, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
```

```r
      TRUE
  }) %>% as.logical()) %>%
  filter(species_level) %>%
  select(-species_level)
diff_gbif <- length(unique(gbif$parsed)) - length(unique(gbif$gbif))
naive <- gbif %>%
  left_join(wf2) %>%
  distinct(parsed, .keep_all = TRUE) %>%
  mutate(species_level = modify(parsed, function(x) {
    len <- str_split(x, " ", simplify = TRUE) %>% length()
    if (len == 1)
      FALSE
    else
      TRUE
  }) %>% as.logical()) %>%
  filter(species_level) %>%
  select(-species_level) %>%
  transmute(parsed,
            gbif = ifelse(is.na(gbif), "NA", gbif),
            torino = ifelse(is.na(matched), "NA", matched)) %>%
  mutate(torino = modify(torino, function(x) {
    paste(str_split(x, " ", simplify = TRUE)[1:2], collapse = " ")
  }),
  torino = gsub("NA NA", "NA", torino))
naive %>%
  filter(gbif != torino) %>%
  distinct_all()
```

```
## # A tibble: 1,837 x 3
##    parsed                  gbif                     torino
##    <chr>                   <chr>                    <chr>
##  1 Acacia melanoceras      Acacia melanoceras       Vachellia melanoceras
##  2 Acanthis cannabina      Acanthis cannabina       NA
##  3 Acanthochaenus lutkeni  Acanthochaenus lutkeni   Acanthochaenus luetkenii
##  4 Acanthopagrus schlegeli Acanthopagrus schlegeli  Acanthopagrus schlegelii
##  5 Acanthurus marginatus   Acanthurus marginatus    Ctenochaetus marginatus
##  6 Acanthurus nigros       Acanthurus nigros        NA
##  7 Acanthurus tennenti     Acanthurus tennenti      Acanthurus tennentii
##  8 Acentronura dendritica  Acentronura dendritica   Amphelikturus dendriticus
##  9 Achyrocline saturioides Achyrocline satureioides NA
## 10 Acipenser oxyrhynchus   Acipenser oxyrhynchus    Acipenser oxyrinchus
## # ... with 1,827 more rows
```

```r
naive %>%
  filter(gbif != torino) %>%
  filter(torino == "NA")
```

```
## # A tibble: 624 x 3
##    parsed                   gbif                     torino
##    <chr>                    <chr>                    <chr>
##  1 Acanthis cannabina       Acanthis cannabina       NA
##  2 Acanthurus nigros        Acanthurus nigros        NA
##  3 Achyrocline saturioides  Achyrocline satureioides NA
##  4 Aconitum delphiniifolium Aconitum delphiniifolium NA
##  5 Actinostemon comunis     Actinostemon communis    NA
##  6 Actitis macularia        Actitis macularia        NA
##  7 Adelosebastes lutens     Adelosebastes latens     NA
##  8 Agalinus purpurea        Agalinis purpurea        NA
##  9 Aglaia barberi           Aglaia barberi           NA
## 10 Ahliesaurs berryi        Ahliesaurus berryi       NA
## # ... with 614 more rows
```

GBIF only harmonized species list had 30,688 unique species names (*Genus species*), whereas Torino had 29,827 (difference = 861 species).

```
naive %>%
  filter(gbif != "NA") %>%
  pull(gbif) %>%
  unique() %>%
  length()
```

```
## [1] 30688
```

```
naive %>%
  filter(torino != "NA") %>%
  pull(torino) %>%
  unique() %>%
  length()
```

```
## [1] 29827
```

Part of the difference is accounted by taxa-specific references identifying synonyms that are considered unique species in GBIF; in total, 688 parsed species names were identified as synonyms in Torino (repeated in total 1,409 times), whereas none was found in GBIF.

```
naive %>%
  filter(gbif != "NA") %>%
  distinct_all() %>%
  select(-torino) %>%
  pull(parsed) %>%
  table() %>%
  table()
```

```
## .
##     1
## 31172
```

```
naive %>%
  filter(torino != "NA") %>%
  select(-gbif) %>%
  distinct_all() %>%
  pull(torino) %>%
  table() %>%
  table()
```

```
## .
##     1     2     3     4
## 29139   657    29     2
```

In summary, of the 44,326 original unique species names, around 11% were the same species with synthatic differences in the way they were written, resolved by passing the species names to `rgnparser`. An additional 15% were removed from harmonization due to not having binomial names, as we were interested in resolving names of taxa at the species level. Both workflows we ran, identified around 92% of the remaining species names, with marginal differences between the harmonized taxonomies. Using only GBIF to harmonize the list of species names resulted in the highest number of matched. This workflow, however, ignored synonym matching accounted for in the other two workflows, with potential consequences on downstream analyses such as species richness and species turnover across sites. Of the original raw names in BioTIME, however, only 81% of the species names already corrected for spelling and syntax were matched, due to the presence of many taxa with taxonomic information only for taxonomic ranks higher than the species level. Despite this relatively low proportion, both our suggested workflows managed to harmonized around 98% of the taxa names that referred to a species (i.e. *Genus species*). Importantly, as we used taxa-specific references when available, the harmonized taxonomy is in line with current taxonomic hypotheses. For taxa that did not have specific references, using GBIF might have resulted in overestimating the number of unique species (see above); however, as there are currently no tools in R to access taxa-specific references, this could not have been solved otherwise, which stress the importance of developing such tools in the future.

```
tibble(steps = c("original",
                 "gnparser",
                 "gnparser + only binomial names",
                 "gnparser + bogota",
                 "gnparser + torino",
                 "gnparser + GBIF"),
       `n unique taxa` = c(44326,
                           44326 - diff_parsed,
                           32900,
                           32900 - diff_bogota,
                           32900 - diff_torino,
                           32900 - diff_gbif),
       `difference from raw` = c(0,
                                 diff_parsed,
                                 diff_parsed + 6692,
                                 diff_parsed + diff_bogota,
                                 diff_parsed + diff_torino,
                                 diff_parsed + diff_gbif),
       `difference from gnparser` = c(NA,
                                      0,
                                      6692,
                                      diff_bogota,
                                      diff_torino,
                                      diff_gbif))
```

```
## # A tibble: 6 x 4
##   steps                `n unique taxa` `difference from r~ `difference from gnp~
##   <chr>                          <dbl>               <dbl>                 <dbl>
## 1 original                       44326                   0                    NA
## 2 gnparser                       39592                4734                     0
## 3 gnparser + only bin~           32900               11426                  6692
## 4 gnparser + bogota              32164                5470                   736
## 5 gnparser + torino              32166                5468                   734
## 6 gnparser + GBIF                32416                5218                   484
```

Here, we have shown three taxonomic harmonization workflow, two coherent with our guidelines and a more naive approach that uses only GBIF to harmonize species names. We presented this example not to udnerstate the utility of GBIF in taxonomic harmonization (for instance, its usage in the first step of Torino had very high accuracy), but rather because this naive approach may be particularly attractive to macroecologists that just started working with taxonomic harmonization. Our aim was to provide example workflows that followed our guidelines and start creating a roadmap that places taxonomic tools into their proper places.