CVrefDB package vignette

Gilles San Martin

18 February 2023 - 03:58

Contents

1 Introduction **2**

2 Simple example **2**

2.1 Load a fasta reference database and its taxonomy 3

2.2 Cross-validation with blast 4

2.3 Assign taxonomy based on blast outputs 5

2.4 Exploit the results 6

2.4.1 % of correct identifications at each taxonomic level and for each method 6

2.4.2 Relationship between % of correct assignments and Identity or Consensus score 7

2.4.3 Compare with another approach : leaked CV 9

2.4.4 Scores per taxon 11

1 Introduction

Cross Validation (CV) of DNA barcodes reference databases (refDB) using blast.

The main aim of this package is to repeatedly blast sequences of a DNA barcode reference database against itself to evaluate the quality of the taxonomic assignments at various taxonomic levels and using different strategies.

There are two main functions :

- `CV_blastn()` : This function will repeatedly extract a random sample of sequences from a DNA barcode reference database in fasta format, then blast these query sequences against the remainder of the database (k-fold cross-validation) or the whole database (leaked cross-validation) and return the output of the top blast hits with the true and predicted taxonomies and blastn statistics (bit score, identity,...).
- `assign_taxonomy()` : This function will assign a taxonomy to sequences based on their top blastn hits. It can use different methods for the assignment based on the best bit score, the consensus score and it allows optionally to specify the minimum identity, length and E-value to take into consideration for each taxonomic level. The output contains a taxonomic assignment for each taxonomic level (species, genus, family,...) and identity and consensus scores that can be used to decide to dismiss untrustworthy assignments

2 Simple example

```
# load the packages necessary to run the examples
library(CVrefDB)
library(Biostrings) # from Bioconductor
library(dplyr)
library(ggplot2)
library(pander) # just to print tables
```

2.1 Load a fasta reference database and its taxonomy

Retrieve an example reference database (fasta file) and its corresponding taxonomic information (in a separate tsv file) from the package data examples. This example contains ~1000 sequences for the ITS2 barcode of plants from the order “Rosales” restricted to a portion of the gene amplified by a given primer used in metabarcoding.

```
fasta_path <- system.file("extdata/ITS2_Rosales_Restricted.fasta",
                          package = "CVrefDB")
taxonomy_path <- system.file("extdata/ITS2_Rosales_Restricted.tsv",
                             package = "CVrefDB")

# Read the fasta and tsv files
fasta <- Biostrings::readDNAStringSet(fasta_path)
reftaxo <- read.table(taxonomy_path, sep = "\t", header = TRUE)

# inspect the content of these files :
fasta
#> DNAStringSet object of length 932:
#>       width seq                                                                                         names
#>   [1]   338 TGCAGAATCCCGTGAACCATCGAGTCTTTGAACGCAAGTTGCGCC...TTCAACGCGACCCCAGGTCAGGCGGGGTTACCCGCTGAATTTAA AB762287.1
#>   [2]   163 TGCAGAATCCCGTGAACCATCGAGTTTTTGAACGCAAGTTGCGCC...GATGGGACGGATGATGGCCTCCCGTGTGCCCGGTCACGCGGTTG AB894161.1
#>   [3]   241 TGCAGAATCCCGTGAACCATCGAGTCTTTGAACGCAAGTTGCGCC...GTCCAAAAATCGATTCCCCCGTCACGTCGTCTTGGCAACAGGTA AF165415.1
#>   [4]   350 TGCAGAATCCCGTGAACCATCGAGTCTTTGAACGCAAGTTGCGCC...TTCAACGCGACCCCAGGTCAGGCGGGATTACCCGCTGAATTTAA AF223066.1
#>   [5]   332 TGCAGAATCCCGTGAACCATCGAGTCTTTGAACGCAAGTTGCGCC...TTCAACGCGACCCCAGGTCAGGCGGGGTTACCCGCTGAATTTAA AF362711.1
#>   ...   ... ...
#> [928]   337 TGCAGAATCCCGTGAACCATCGAGTCTTTGAACGCAAGTTGCGCC...TTCAACGCGACCCCAGGTCAGGCGGGGTTACCCGCTGAATTTAA MN905585.1
#> [929]   339 TGCAGAATCCCGTGAACCATCGAGTCTTTGAACGCAAGTTGCGCC...TTCAACGCGACCCCAGGTCAGGCGGGGTTACCCGCTGAATTTAA MN905594.1
#> [930]   338 TGCAGAATCCCGTGAACCATCGAGTCTTTGAACGCAAGTTGCGCC...TTCAACGCGACCCCAGGTCAGGCGGGGTCACCCGCTGAATTTAA MN905578.1
#> [931]   336 TGCAGAATCCCGTGAACCATCGAGTCTTTGAACGCAAGTTGCGCC...TTCAACGCGACCCCAGGTCAGGCGGGGTTACCCGCTGAATTTAA MN905586.1
#> [932]   332 TGCAGAATCCCGTGAACCATCGAGTCTTTGAACGCAAGTTGCGCC...TTCAACGCGACCCCAGGTCAGGCGGGGTTACCCGCTGAATTTAA MW474729.1
head(reftaxo)
#>           ID
#> 1 KM037198.1
#> 2 KP858847.1
#> 3 KM037495.1
#> 4 KP858908.1
#> 5 KP858891.1
#> 6 KR086741.1
#>                                                                                                            Taxonomy
#> 1                   k__Viridiplantae; p__Streptophyta; c__Magnoliopsida; o__Rosales; f__Rosaceae; g__Rubus; s__cf.
#> 2       k__Viridiplantae; p__Streptophyta; c__Magnoliopsida; o__Rosales; f__Urticaceae; g__Elatostema; s__balansae
#> 3           k__Viridiplantae; p__Streptophyta; c__Magnoliopsida; o__Rosales; f__Rosaceae; g__Rubus; s__perrobustus
#> 4 k__Viridiplantae; p__Streptophyta; c__Magnoliopsida; o__Rosales; f__Urticaceae; g__Poikilospermum; s__acuminatum
#> 5    k__Viridiplantae; p__Streptophyta; c__Magnoliopsida; o__Rosales; f__Urticaceae; g__Elatostema; s__hypoglaucum
#> 6     k__Viridiplantae; p__Streptophyta; c__Magnoliopsida; o__Rosales; f__Cannabaceae; g__Aphananthe; s__cuspidata
```

2.2 Cross-validation with blast

Proceed to the cross validation step with `CV_blastn` : by default it is a 10-folds cross-validation retrieving for each query sequence the 15 top blast hits.

We examine then the output for one of the query sequences corresponding to Prunus pseudocerasus (column `Species_true`). The best blast hit is Prunus avium (which is incorrect, see Column `Species` corresponding to the matching sequence) with an identity score of 99.69%, while the 7 next hits correspond to Prunus pseudocerasus (which is correct) with Identity scores ranging between 98.2% and 99.1%.

```
output_10FoldCV <- CV_blastn(fasta_db = fasta, taxo = reftaxo, seed = 12, verbose = FALSE)

dim(output_10FoldCV)
#> [1] 14062    24

# examine the results for one queried sequence
output_10FoldCV %>%
    # drop some columns so that the output fits and A4 page...
    select(-c(Label, Fold, E_value, Kingdom_true, Phylum_true, Class_true,
              Order_true, Family_true,
              Species_orig_true,Kingdom,Phylum,Class,Order,Family, Species_orig)) %>%
    filter(TaxID_query == "KF241102.1")
#>    TaxID_query Genus_true         Species_true TaxID_blast Bit_score Length Identity  Genus                Species
#> 1   KF241102.1     Prunus Prunus pseudocerasus  FJ899097.1       606    332   99.699 Prunus           Prunus avium
#> 2   KF241102.1     Prunus Prunus pseudocerasus  KF241127.1       595    331   99.094 Prunus  Prunus pseudocerasus
#> 3   KF241102.1     Prunus Prunus pseudocerasus  KF241110.1       590    331   98.792 Prunus  Prunus pseudocerasus
#> 4   KF241102.1     Prunus Prunus pseudocerasus  KF241130.1       590    331   98.792 Prunus  Prunus pseudocerasus
#> 5   KF241102.1     Prunus Prunus pseudocerasus  KF241119.1       590    331   98.792 Prunus  Prunus pseudocerasus
#> 6   KF241102.1     Prunus Prunus pseudocerasus  KF241103.1       590    331   98.792 Prunus  Prunus pseudocerasus
#> 7   KF241102.1     Prunus Prunus pseudocerasus  KF241125.1       584    331   98.489 Prunus  Prunus pseudocerasus
#> 8   KF241102.1     Prunus Prunus pseudocerasus  KF241111.1       584    331   98.489 Prunus  Prunus pseudocerasus
#> 9   KF241102.1     Prunus Prunus pseudocerasus  KF241116.1       579    331   98.187 Prunus  Prunus pseudocerasus
#> 10  KF241102.1     Prunus Prunus pseudocerasus  GQ179666.1       556    331   96.979 Prunus         Prunus cornuta
#> 11  KF241102.1     Prunus Prunus pseudocerasus  FJ606753.1       553    333   96.697 Prunus             Prunus sp.
#> 12  KF241102.1     Prunus Prunus pseudocerasus  KC862377.1       551    331   96.677 Prunus      Prunus lusitanica
#> 13  KF241102.1     Prunus Prunus pseudocerasus  FJ572043.1       544    331   96.375 Prunus          Prunus dulcis
#> 14  KF241102.1     Prunus Prunus pseudocerasus  MN215998.1       379    341   87.390 Sorbus    Sorbus tianschanica
#> 15  KF241102.1     Prunus Prunus pseudocerasus  KC896708.1       375    338   87.278  Rubus Rubus cochinchinensis
```

2.3 Assign taxonomy based on blast outputs

Then, we assign the taxonomy with `assign_taxonomy()` which will use the previous results to assign the taxnonomy to each query sequence with 4 different methods by default based on the Bit score and/or on the consensus score (% of hits among the top 10 by default agreeing on a given taxonomy). We look again at the output of the same query sequence.

```
assigned_long <-
assign_taxonomy(output_10FoldCV, taxo = reftaxo,
                Order = NA # Ignore the order level
                )
assigned_long %>%
    filter(ID == "KF241102.1")
#> # A tibble: 12 x 10
#>    Method     Tax_level Level ID         Taxon                 Bit Length Ident  Cons Taxon_true
#>    <chr>      <fct>     <fct> <chr>      <chr>               <dbl>  <int> <dbl> <dbl> <chr>
#>  1 TopHit     Family    f     KF241102.1 Rosaceae              606    332  99.7 100   Rosaceae
#>  2 TopHit     Genus     g     KF241102.1 Prunus                606    332  99.7 100   Prunus
#>  3 TopHit     Species   s     KF241102.1 Prunus avium          606    332  99.7  10   Prunus pseudocerasus
#>  4 TopHitPlus Family    f     KF241102.1 Rosaceae              606    332  99.7 100   Rosaceae
#>  5 TopHitPlus Genus     g     KF241102.1 Prunus                606    332  99.7 100   Prunus
#>  6 TopHitPlus Species   s     KF241102.1 Prunus avium          606    332  99.7  10   Prunus pseudocerasus
#>  7 TopN       Family    f     KF241102.1 Rosaceae              606    332  99.7 100   Rosaceae
#>  8 TopN       Genus     g     KF241102.1 Prunus                606    332  99.7 100   Prunus
#>  9 TopN       Species   s     KF241102.1 Prunus pseudocerasus  595    331  99.1  80   Prunus pseudocerasus
#> 10 TopNPlus   Family    f     KF241102.1 Rosaceae              606    332  99.7 100   Rosaceae
#> 11 TopNPlus   Genus     g     KF241102.1 Prunus                606    332  99.7 100   Prunus
#> 12 TopNPlus   Species   s     KF241102.1 Prunus pseudocerasus  595    331  99.1  88.9 Prunus pseudocerasus
```

This output table is sometimes easier to read when we have only one row for each query sequence and each method and that the various taxonomic levels are displayed in column. This can be done with the function `pivot_assign_taxonomy()` (based on `tidyr::pivot_wider()`).

```
assigned_wide <- pivot_assign_taxonomy(assigned_long)

assigned_wide %>%
    select(!starts_with("f_")) %>%
    select(!matches("Bit|Length")) %>%
    filter(ID == "KF241102.1")
#> # A tibble: 4 x 10
#>   Method     ID         g_Taxon_true g_Taxon g_Ident g_Cons s_Taxon_true         s_Taxon              s_Ident s_Cons
#>   <chr>      <chr>      <chr>        <chr>     <dbl>  <dbl> <chr>                <chr>                  <dbl>  <dbl>
#> 1 TopHit     KF241102.1 Prunus       Prunus     99.7    100 Prunus pseudocerasus Prunus avium            99.7   10
#> 2 TopHitPlus KF241102.1 Prunus       Prunus     99.7    100 Prunus pseudocerasus Prunus avium            99.7   10
#> 3 TopN       KF241102.1 Prunus       Prunus     99.7    100 Prunus pseudocerasus Prunus pseudocerasus    99.1   80
#> 4 TopNPlus   KF241102.1 Prunus       Prunus     99.7    100 Prunus pseudocerasus Prunus pseudocerasus    99.1   88.9
```

At the species level, the methods `TopHit` and `TopHitPlus` both select (wrongly) "Prunus avium" because they are based on the best blast hit (best Bit Score) while `TopN` and `TopNPlus` methods select "Prunus pseudocerasus" because they are mainly based on the consensus score (`s_Cons`). With `TopN`, the species level consensus score is 70% because 7 out of the 10 best blast hits correspond to this species. For `TopNplus`, this consensus score is higher (87.5%) because this method will first filter out the hits which do not fulfill some criteria. For example, here the Identity score must be at least 97% for the species level identifications. As a consequence, the 9th and 10th best hits are discarded and the consensus score is 7/8.

2.4 Exploit the results

2.4.1 % of correct identifications at each taxonomic level and for each method

You can then explore the results in many different ways.

We can for example compute the % of correct identifications for each taxonomic level and each method (NB the missing assignments are not counted here as correct identification but this could be subject to discussion)

```
accuracy <-
assigned_long %>%
    mutate(CorrectID = Taxon == Taxon_true & !is.na(Taxon)) %>%
    group_by(Method, Tax_level) %>%
    summarize(CorrectID = sum(CorrectID)*100/n()) %>%
    arrange(Tax_level)
accuracy
#> # A tibble: 12 x 3
#> # Groups:   Method [4]
#>    Method     Tax_level CorrectID
#>    <chr>      <fct>         <dbl>
#>  1 TopHit     Family         99.8
#>  2 TopHitPlus Family         99.8
#>  3 TopN       Family         98.3
#>  4 TopNPlus   Family         98.6
#>  5 TopHit     Genus          91.8
#>  6 TopHitPlus Genus          91.8
#>  7 TopN       Genus          84.2
#>  8 TopNPlus   Genus          87.8
#>  9 TopHit     Species        15.9
#> 10 TopHitPlus Species        16.8
#> 11 TopN       Species        13.2
#> 12 TopNPlus   Species        16.5

# x11(width = 10/2.54, height = 5/2.54)
plot_accuracy <-
accuracy %>%
    ggplot(aes(x = Tax_level, y = CorrectID, fill = Method)) +
    geom_col(position = position_dodge()) +
    coord_flip() +
    scale_fill_manual(values = (c("red4", "orangered", "orange", "gold"))) +
    xlab("") + ylab("% correct taxonomic assignment")+
    theme_bw(8)
plot_accuracy
```

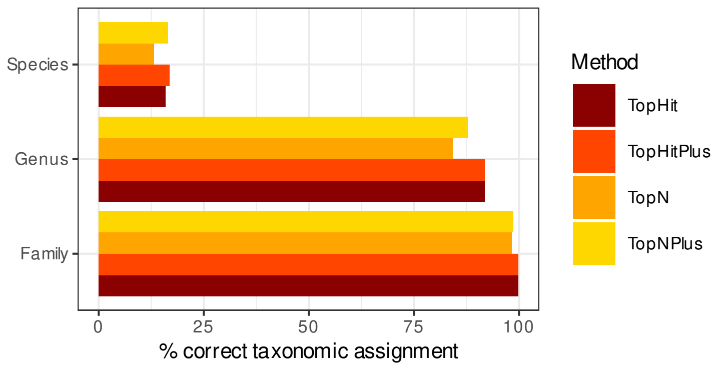

Figure 1:

2.4.2 Relationship between % of correct assignments and Identity or Consensus score

We can explore also the relationship between the identity or consensus score with the proportion of correct identification (with a binomial GLM). This might be useful to determine thresholds under which the identification should not be trusted.

```
# x11(width = 17/2.54, height = 6.5/2.54)
plot_identity <-
assigned_long %>%
    mutate(CorrectID = Taxon == Taxon_true & !is.na(Taxon)) %>%
    filter(Method == "TopHitPlus") %>%
    ggplot(aes(y = as.numeric(CorrectID), x = Ident)) +
    facet_wrap(~Tax_level, nrow = 1) +
    geom_point(alpha = 0.1, size = 0.75,
               position = position_jitter(width = 0, height = 0.1)) +
    stat_smooth(method = "glm", se = FALSE,  method.args = list(family = binomial))+
    ylab("Proportion of \ncorrect identification") +
    xlab("Identity score (%)") +
    theme_bw(10)

plot_consensus <-
assigned_long %>%
    mutate(CorrectID = Taxon == Taxon_true & !is.na(Taxon)) %>%
    filter(Method == "TopHitPlus") %>%
    ggplot(aes(y = as.numeric(CorrectID), x = Cons)) +
    facet_wrap(~Tax_level, nrow = 1) +
    geom_point(alpha = 0.1, size = 0.75,
               position = position_jitter(width = 0, height = 0.1)) +
    stat_smooth(method = "glm", se = FALSE,  method.args = list(family = binomial))+
    ylab("Proportion of \ncorrect identification") +
    xlab("Consensus score (%)") +
    theme_bw(10)

plot_identity
plot_consensus
```

The plots show for example that at the genus level the proportion of correct identification drops very quickly when the % of identity is < 95-97%. The situation is rather different at species level though : even when the % of Identity is very close to 100%, the proportion of correct identification remains very low. High consensus scores (>20-30%) are a good indication that the species level identification is correct, however this concerns only a very small proportion of the data.

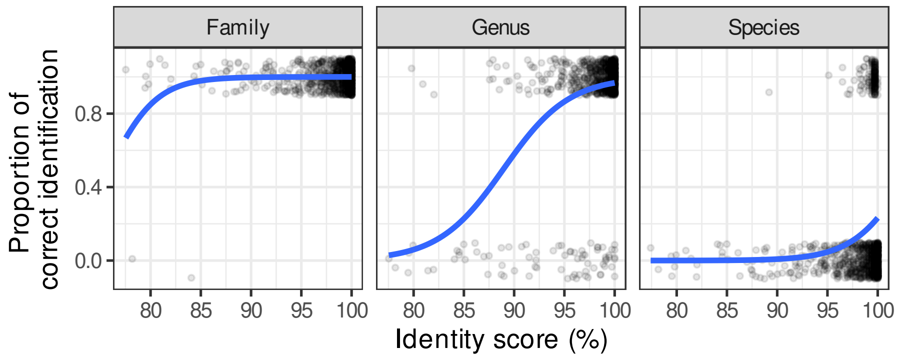

Figure 2:

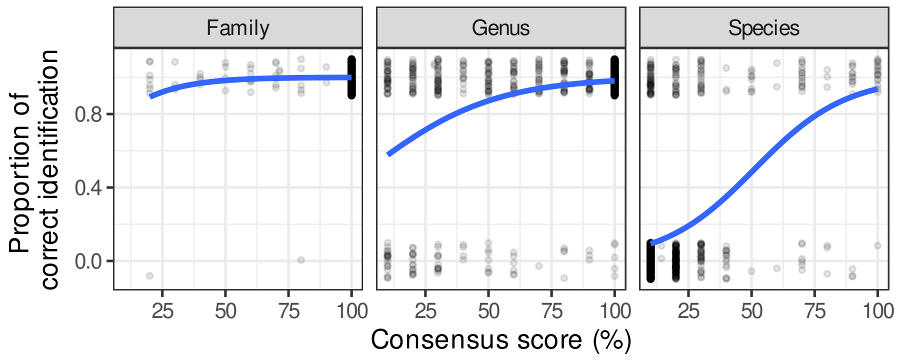

Figure 3:

We can also examine how the correct/incorrect identification are distributed in a 2 dimensional space formed by the identity and consensus scores. This approach does not help much however in our small study case. For example, at species level, when the consensus score is 20 or 30% we can still have both wrong and correct identification even when the Identity is close to 100%

```
# x11(width = 11/2.54, height = 7/2.54)
assigned_long %>%
    mutate(CorrectID = Taxon == Taxon_true & !is.na(Taxon)) %>%
    filter(Method == "TopHitPlus") %>%
    ggplot(aes(y = Ident, x = Cons)) +
    facet_grid(CorrectID~Tax_level,
               labeller = labeller(.rows = label_both)) +
    geom_point(alpha = 0.1, size = 0.75,
             position = position_jitter(width = 1, height = 0)) +
    ylab("Identity score (%)") +
    xlab("Consensus score (%)") +
    theme_bw(10)
```

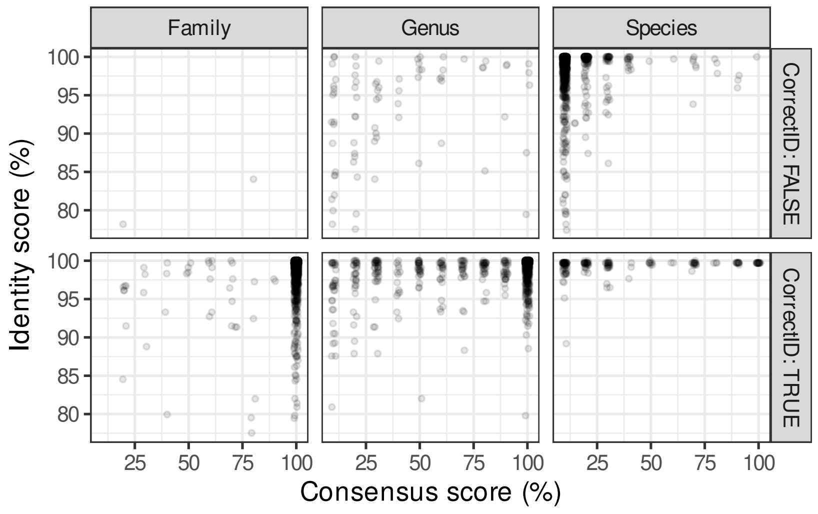

Figure 4:

2.4.3 Compare with another approach : leaked CV

Note however that all these results are using 10 folds cross validation. This means that when we blast a query sequence, this sequence has been removed from the reference database. So, if a species is only represented by one sequence in the database (e.g. because of the absence of intraspecific variation or because the species is rare or the database incomplete) blast can never match the right taxon at the species level. This means that these estimates are probably pessimistic relative to the true accuracy.

We can easily compute the same statistics but with a "leaked cross-validation" approach in which the blasted query sequences remain in the reference database. We just need to change the default option `k=10` (for 10 folds CV) to `k=1` :

```
output_leakedCV <- CV_blastn(fasta_db = fasta, taxo = reftaxo,
                             k = 1, seed = 12, verbose = FALSE)
assigned_long_leakedCV <- assign_taxonomy(output_leakedCV, taxo = reftaxo, Order = NA)

# recompute accuracy
accuracy_leakedCV <-
assigned_long_leakedCV %>%
    mutate(CorrectID = Taxon == Taxon_true & !is.na(Taxon)) %>%
    group_by(Method, Tax_level) %>%
    summarize(CorrectID = sum(CorrectID)*100/n()) %>%
    arrange(Tax_level)

# add logical column CorrectID and keep only the TopHitPlus results in a temporary object
tmp <-
assigned_long_leakedCV %>%
    mutate(CorrectID = Taxon == Taxon_true & !is.na(Taxon)) %>%
    filter(Method == "TopHitPlus")
```

The accuracy is better at the species level but we still have at least 20% of sequences that are not correctly identified at species level. There is again a clear advantage for the TopHit and TopHitPlus methods.

```
# recycle previous graph by just changing the dataset with ggplot %+% operator
plot_accuracy %+% accuracy_leakedCV
```

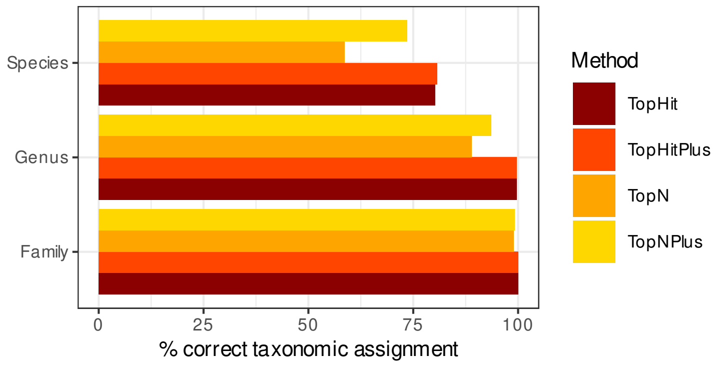

Figure 5:

The next graphs show that almost all hits have 100% identity even though at species level only 80% of the identifications are correct. This means that in this case the identity is not a good indicator of the trustworthiness of the identification (or we can consider that we need at least 100%).

We can also see that at the species level, we have relatively few errors when the consensus score is > 20%, however, this concerns only a very small part of the sequences as most of the results have consensus scores <= 20% including correct and incorrect identifications.

```
plot_identity %+% tmp
```

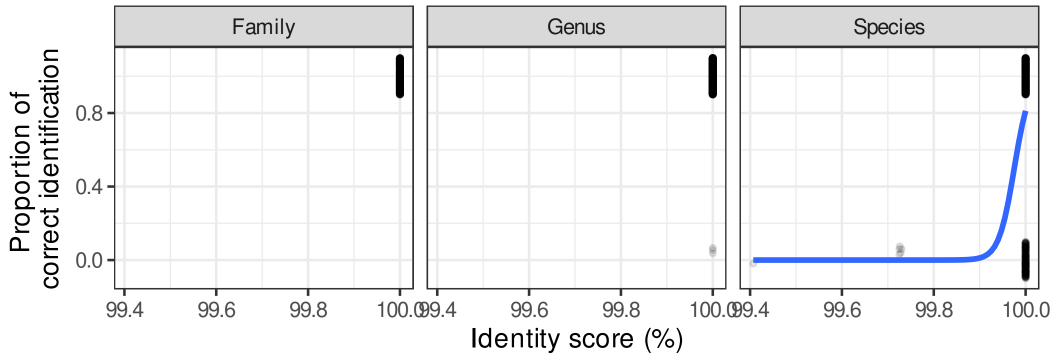

Figure 6:

```
plot_consensus %+% tmp
```

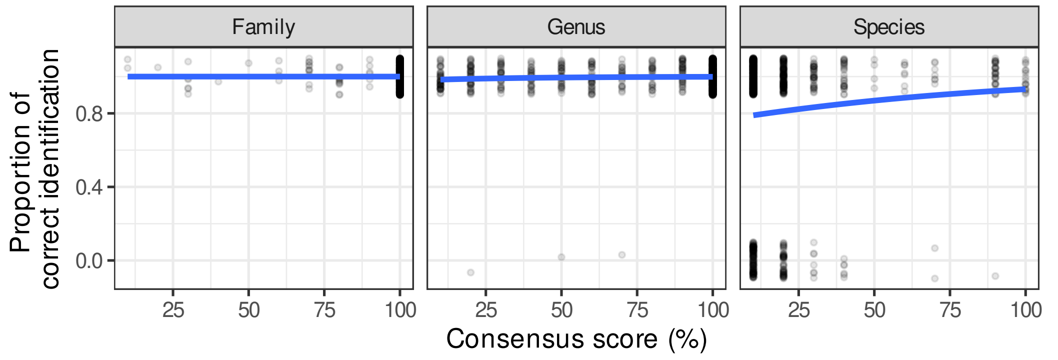

Figure 7:

2.4.4 Scores per taxon

We might also want to explore how each family or each genus performs separately. The function `score_per_taxon()` allows you to explore these questions in one way among many other ways.

For example with the following code we can see that among the 29 sequences (`N`) in the family `Elaeagnaceae` (column `Grouping_taxon`), 100%, 100% and 82.7% are correctly predicted (column `Pct`) at the family, genus and species level respectively (column `Tax_level`). In contrast for the Rosaceae family the assignments are correct in 100%, 93.9% and 7.6% of the cases at family, genus and species level.

```
scores <-
score_per_taxon(assigned_long %>% filter(Method == "TopHitPlus"),
                grouping_tax_level = "Family",
                predicted_NA_wrong = TRUE
                )
scores %>%
    arrange(Tax_level, desc(Pct)) %>%
    pander::pander() # just to print a nice table
```

Method	Tax_level	Level	Grouping_taxon	Pct	N
TopHitPlus	Family	f	Elaeagnaceae	100.000	29
TopHitPlus	Family	f	Rhamnaceae	100.000	10
TopHitPlus	Family	f	Rosaceae	100.000	460
TopHitPlus	Family	f	Ulmaceae	100.000	14
TopHitPlus	Family	f	Urticaceae	100.000	248
TopHitPlus	Family	f	Moraceae	99.383	162
TopHitPlus	Family	f	Cannabaceae	88.889	9
TopHitPlus	Genus	g	Elaeagnaceae	100.000	29
TopHitPlus	Genus	g	Moraceae	97.531	162
TopHitPlus	Genus	g	Rosaceae	93.913	460
TopHitPlus	Genus	g	Ulmaceae	92.857	14
TopHitPlus	Genus	g	Cannabaceae	88.889	9
TopHitPlus	Genus	g	Urticaceae	85.081	248
TopHitPlus	Genus	g	Rhamnaceae	50.000	10
TopHitPlus	Species	s	Elaeagnaceae	82.759	29
TopHitPlus	Species	s	Ulmaceae	57.143	14
TopHitPlus	Species	s	Cannabaceae	55.556	9
TopHitPlus	Species	s	Rhamnaceae	30.000	10
TopHitPlus	Species	s	Urticaceae	27.823	248
TopHitPlus	Species	s	Moraceae	8.025	162
TopHitPlus	Species	s	Rosaceae	7.609	460

Graphical represnetation of this table :

```
# x11(width = 8/2.54, height = 5/2.54)
scores %>%
    filter(Tax_level != "Family") %>%
    arrange(Tax_level, desc(Pct)) %>%
    mutate(Grouping_taxon = factor(Grouping_taxon,
                                   levels = rev(unique((Grouping_taxon))))) %>%
    ggplot(aes(y = Pct, x = Grouping_taxon)) +
    facet_wrap(~Tax_level) +
    geom_col(position = position_dodge()) +
    coord_flip() +
    xlab("") + ylab("% correct taxonomic assignment")+
    theme_bw(10)
```

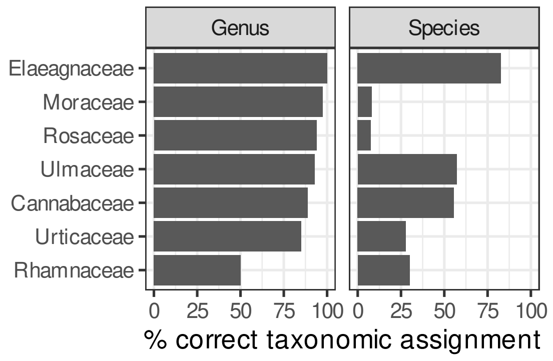

Figure 8:

We can look at similar results but using the leaked CV assignments. The genus level assignments are always very good but the family Rosaceae performs clearly worse than the others at the species level with only 66.3% of correct assignment even when the true sequence is in the reference database…

```
scores <-
score_per_taxon(assigned_long_leakedCV %>% filter(Method == "TopHitPlus"),
                 grouping_tax_level = "Family",
                 predicted_NA_wrong = TRUE
                 )
scores %>%
    arrange(Tax_level, desc(Pct)) %>%
    pander::pander()
```

Method	Tax_level	Level	Grouping_taxon	Pct	N
TopHitPlus	Family	f	Cannabaceae	100.000	9
TopHitPlus	Family	f	Elaeagnaceae	100.000	29
TopHitPlus	Family	f	Moraceae	100.000	162
TopHitPlus	Family	f	Rhamnaceae	100.000	10
TopHitPlus	Family	f	Rosaceae	100.000	460
TopHitPlus	Family	f	Ulmaceae	100.000	14
TopHitPlus	Family	f	Urticaceae	100.000	248
TopHitPlus	Genus	g	Cannabaceae	100.000	9
TopHitPlus	Genus	g	Elaeagnaceae	100.000	29
TopHitPlus	Genus	g	Moraceae	100.000	162
TopHitPlus	Genus	g	Rhamnaceae	100.000	10
TopHitPlus	Genus	g	Ulmaceae	100.000	14
TopHitPlus	Genus	g	Rosaceae	99.783	460
TopHitPlus	Genus	g	Urticaceae	99.194	248
TopHitPlus	Species	s	Cannabaceae	100.000	9
TopHitPlus	Species	s	Rhamnaceae	100.000	10
TopHitPlus	Species	s	Elaeagnaceae	96.552	29
TopHitPlus	Species	s	Urticaceae	94.355	248
TopHitPlus	Species	s	Moraceae	93.827	162
TopHitPlus	Species	s	Ulmaceae	92.857	14
TopHitPlus	Species	s	Rosaceae	66.522	460

We can try to look at what is happening in the Rosaceae family by computing the % of correct assignments for each genus at species level. We display only the genus with less than 100% of correct predictions with leaked CV. We can see for example that the difficult genus Rubus with 262 sequences is correctly predicted only 46.9% of the time and this is certainly decreasing the global performance of the family Rosaceae.

```
scores <-
score_per_taxon(assigned_long_leakedCV %>% filter(Method == "TopHitPlus"),
                 grouping_tax_level = "Genus",
                 predicted_NA_wrong = TRUE
                 )
scores %>%
    arrange(Tax_level, desc(Pct)) %>%
    filter(Tax_level == "Species" & Pct < 100) %>%
    pander::pander()
```

Method	Tax_level	Level	Grouping_taxon	Pct	N
TopHitPlus	Species	s	Cliffortia	96.296	27
TopHitPlus	Species	s	Hippophae	96.154	26
TopHitPlus	Species	s	Elatostema	95.614	114
TopHitPlus	Species	s	Boehmeria	95.455	22
TopHitPlus	Species	s	Debregeasia	92.857	14
TopHitPlus	Species	s	Ficus	92.857	140
TopHitPlus	Species	s	Nanocnide	92.308	13
TopHitPlus	Species	s	Prunus	89.474	19
TopHitPlus	Species	s	Spiraea	87.500	8
TopHitPlus	Species	s	Rosa	83.333	6
TopHitPlus	Species	s	Rhaphiolepis	80.000	15
TopHitPlus	Species	s	Parietaria	76.923	13
TopHitPlus	Species	s	Poikilospermum	75.000	4
TopHitPlus	Species	s	Pourthiaea	75.000	4
TopHitPlus	Species	s	Prinsepia	75.000	4
TopHitPlus	Species	s	Vauquelinia	75.000	4
TopHitPlus	Species	s	Zelkova	75.000	4
TopHitPlus	Species	s	Amelanchier	69.231	13
TopHitPlus	Species	s	Phippsiomeles	66.667	3
TopHitPlus	Species	s	Pipturus	66.667	3
TopHitPlus	Species	s	Rubus	47.328	262
TopHitPlus	Species	s	Sarcochlamys	0.000	1