Supplemental Material - *metagenomeFeatures*: An R package for working with 16S rRNA reference databases and marker-gene survey feature data.

Nate Olson

2018-05-23

Contents

1 *Paenibacillus* species resolution for 16S rRNA V12 and V4 regions.

1.0.1 Background

16S rRNA amplicon sequencing is commonly used for microbial community characterization, including differential abundance analysis. A limitation to 16S rRNA amplicon sequencing is a lack of taxonomic resolution, where organisms are only identifiable to the genus or family level. We define taxonomic resolution as the ability to differentiate between groups within a taxonomic level, for example differentiating between species within a genus. While similar to determining whether a sequence represents a novel species, here we are only interested in determining whether the 16S rRNA region of interest contains sufficient information for species-level taxonomic assignment. Taxonomic resolution varies by clade and amplicon regions. Though the extent to which taxonomic resolition varies is not well characterized.

Here we demonstrate how *metagenomeFeatures* and the `MgDb` annotation packages can be used to characterize taxonomic resolution for a specific clade and amplicon region, specifically for the *Paenibacillus* genus and V12 and V4 regions. Originally classified under the *Bacillus* genus, a novel genus was formed based on the 16S rRNA gene similarity in the 1990s. *Paenibacillus* spp. are facultative anaerobic bacteria present in a variety of environments including the soil, water, and can act as opportunistic pathogens in humans [@ouyang2008]. *Paenibacillus* spp. will play an important role in sustainable agricultural industries [@grady2016]. As such, appropriate speciation is of interest. The V12 and V4 region were used as they represent two commonly used amplicons for 16S rRNA marker-gene surveys. We will use the Greengenes 13.5 database, accessed using the *greengenes13.5MgDb* annotation package for our analysis of the *Paenibacillus* genus. The Greengenes 13.5 database is used for demonstration purposes but the other `MgDb` annotation packages can also be used; RDP 11.5 - `ribosomaldatabaseproject11.5MgDb or SILVA 128.1` –silva128.1MgDb‘.

1.0.2 Required Packages

In addition to *metagenomeFeatures* and *greengenes13.5MgDb* the *DECIPHER*, *tidyverse*, and *ggpubr* packages are also used in the following analysis. Our analysis uses the `DECIPHER` package to extract the amplicon regions, perform multiple sequence alignment, and generate a pairwise sequence distance matrix [@Wright2016-mo]. The `tidyverse` and `ggpubr` packages will be used to reformat the taxonomic and distance matrix data and generate summary figures [@tidyverse;@ggpubr].

```
library(tidyverse); packageVersion("tidyverse")
```

```
## [1] '1.2.1'
```

```
library(ggpubr); packageVersion("ggpubr")
```

```
## [1] '0.1.6'
```

```
library(DECIPHER); packageVersion("DECIPHER")
```

```
## [1] '2.8.1'
```

```
library(metagenomeFeatures); packageVersion("metagenomeFeatures")
```

```
## Warning: replacing previous import 'lazyeval::is_formula' by
## 'purrr::is_formula' when loading 'metagenomeFeatures'
```

```
## Warning: replacing previous import 'lazyeval::is_atomic' by
## 'purrr::is_atomic' when loading 'metagenomeFeatures'
```

```
## [1] '2.0.0'
```

```
library(greengenes13.5MgDb); packageVersion("greengenes13.5MgDb")
```

```
## [1] '2.0.0'
```

1.0.3 *Paenibacillus* Sequence and Taxonomy Data

We first subset the Greeengenes 13.5 database using the `mgDb_select` function. Then summarize the taxonomy data using functions from `tidyverse` package, specifically `dplyr`, `stringr` and `forcats` functions for manipulating `data.frames`, `strings`, and `factor` vectors respectively.

```
paeni_16S <- metagenomeFeatures::mgDb_select(gg13.5MgDb,
                          type = c("taxa","seq"),
                          keys = "Paenibacillus",
                          keytype = "Genus")
```

```
## Per genus count data
taxa_df <- paeni_16S$taxa %>%
    ## cleaning up species names
    mutate(Species = if_else(Species == "s__", "Unassigned", Species),
           Species = str_replace(Species, "s__","")) %>%
    group_by(Species) %>%
    summarise(Count = n()) %>%
    ungroup() %>%
    mutate(Species = fct_reorder(Species, Count))

## Count info for text
total_otus <- sum(taxa_df$Count)
unassigned_idx <- taxa_df$Species == "Unassigned"
no_species_assignment <- taxa_df$Count[unassigned_idx]
```

For the Greengenes 13.5 database, there are a total of 2912 sequences classified as 15 species in the Genus *Paenibacillus*. The number of sequences assigned to specific *Paenibacillus* species range from 199 *Paenibacillus amylolyticus* to 2 *Paenibacillus illinoisensis* (Fig. 1). Sequences only classified to the genus level, "Unassigned", is the most abundant group, 2308.

```
taxa_df %>%
ggplot() +
    geom_bar(aes(x = Species, y = Count), stat = "identity") +
    geom_text(aes(x = Species, y = Count, label = Count),
```

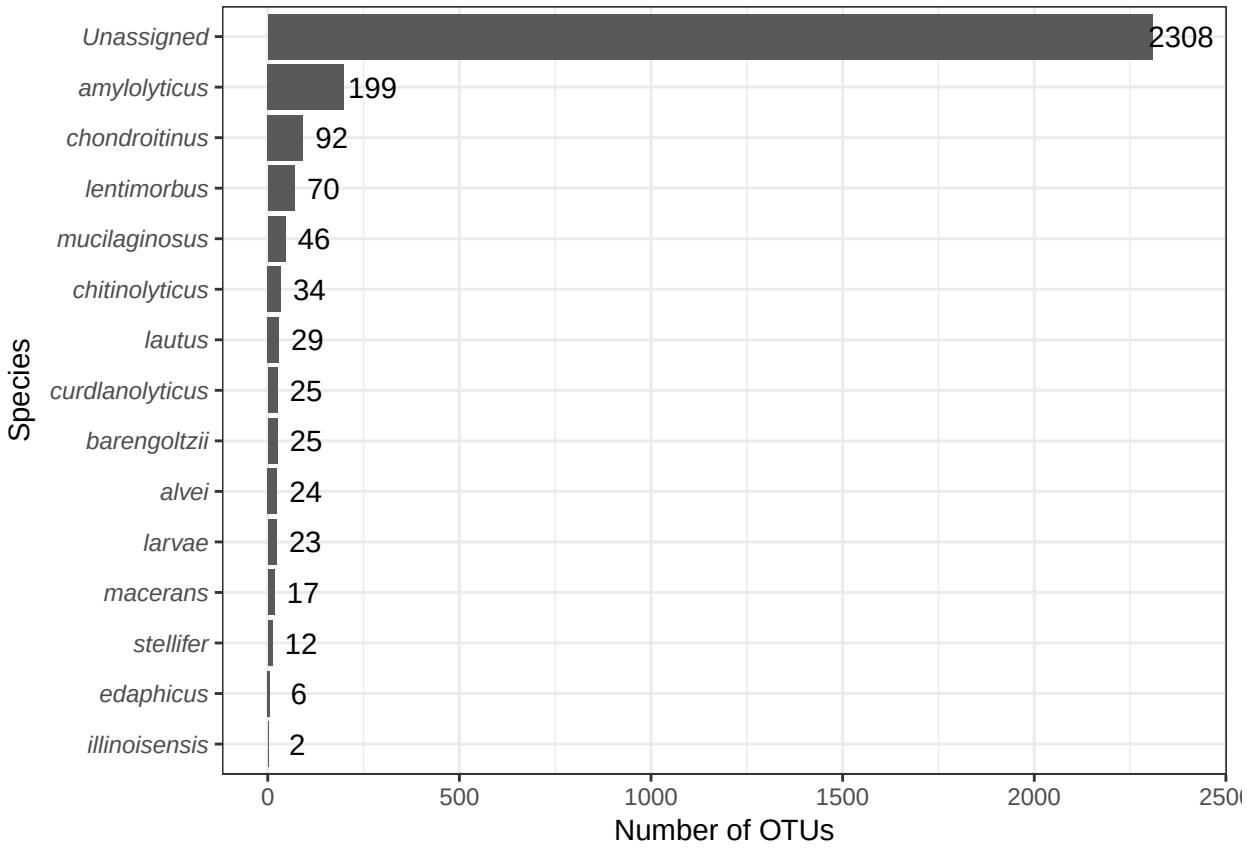

Figure 1: Number of sequences assigned to species in the genus *Paenibacillus*.

```
            nudge_y = 75) +
  labs(y = "Number of OTUs") +
  coord_flip() +
  theme_bw() +
  theme(axis.text.y = element_text(face = "italic"))
```

1.0.4 Taxonomic resolution

Next, we evaluate the 16S rRNA amplicon sequencing taxonomic resolution for *Paenibacillus* species by comparing within and between species amplicon pairwise distance for the V12 and V4 regions. To differentiate between species the pairwise distances for within-species amplicon sequences must be less than the between species distances. Additionally ,the difference in amplicon sequence pairwise distances between and within species must be greater than the sequencing error rate to detect the difference. For our taxonomic resolution analysis, we used pattern matching to extract the V12 and V4 regions of the 16S rRNA sequences. We then generate a pairwise distance matrix for the two regions and compare the within and between species pairwise distances.

For our *in-silico* PCR we will use the following PCR primers:

Region	Direction	Primer
V12	Forward	27F - AGAGTTTGATCATGGCTCAG
	Reverse	336R - CACTGCTGCSYCCCGTAGGAGTCT

Region	Direction	Primer
V4	Forward	515F - GTGCCAGCMGCCGCGGTAA
	Reverse	806R - GGACTACHVGGGTWTCTAAT

1.0.4.1 V12

Extracting the V12 region from the database sequences, only sequences with containing both forward and reverse primers are included in the analysis.

```
forward_primer <- "AGAGTTTGATCATGGCTCAG"
## reverse complementing reverse primer
reverse_primer <- DNAString("CACTGCTGCSYCCCGTAGGAGTCT") %>%
    reverseComplement() %>%
    as.character()

## Finding sequeces with forward primer
forward_match <- Biostrings::vmatchPattern(forward_primer,
                                           subject = paeni_16S$seq,
                                           max.mismatch = 2) %>%
    as.list() %>% map_dfr(as.data.frame,.id = "seq_id")

## Finding sequences with reverse primer
reverse_match <- Biostrings::vmatchPattern(reverse_primer,
                                           subject = paeni_16S$seq,
                                           max.mismatch = 2,
                                           fixed = FALSE) %>%
    as.list() %>% map_dfr(as.data.frame,.id = "seq_id")

## sequences with both forward and reverse primers
seqs_to_use_ids <- intersect(forward_match$seq_id, reverse_match$seq_id)
seqs_to_use <- names(paeni_16S$seq) %in% seqs_to_use_ids

## Trimming sequences with both primers
paeni_V12 <- TrimDNA(paeni_16S$seq[seqs_to_use],
                     leftPatterns = forward_primer,
                     rightPatterns = reverse_primer,
                     type = "both")
```

```
## Finding left pattern: 100% internal, 0% flanking
##
## Finding right pattern: 100% internal, 0% flanking
##
## Time difference of 0.05 secs
```

```
## Excluding seqs with lenght 0
paeni_V12_seqs <- paeni_V12[[2]][width(paeni_V12[[2]]) != 0]
```

Generating a multiple sequence alignment using the `AlignSeqs` function in the `DECIPHER` package.

```
v12_align <- AlignSeqs(paeni_V12[[2]], verbose = FALSE)
```

The resulting alignment can be viewed using the `BrowseSeqs` function in the `DECIPHER` package.

```
BrowseSeqs(v12_align)
```

Generating pairwise distance matrix using the `DistanceMatrix` function in the `DECIPHER` package for taxonomic resolution analysis and converting distance matrix to a data frame for analysis.

```
v12_dist <- DistanceMatrix(v12_align,
                           correction = "none",
                           verbose = FALSE,
                           includeTerminalGaps = FALSE)

v12_dist_df <- v12_dist %>%
    as.data.frame() %>%
    rownames_to_column(var = "Keys") %>%
    gather("Keys2","distance", -Keys) %>%
    mutate(Keys = as.numeric(Keys), Keys2 = as.numeric(Keys2)) %>%
    filter(Keys < Keys2) %>%
    mutate(Keys = as.character(Keys), Keys2 = as.character(Keys2))

tax_df <- dplyr::select(paeni_16S$taxa, "Keys", "Species")
v12_dist_anno_df <- v12_dist_df %>%
    left_join(tax_df) %>%
    left_join(tax_df,by = c("Keys2" = "Keys")) %>%
    dplyr::rename(Keys_Species = Species.x, Keys2_Species = Species.y) %>%
        mutate(group_comp = if_else(Keys_Species == Keys2_Species,
                                    "within","between")) %>%
    filter(Keys_Species != "s__", Keys2_Species != "s__")
```

```
## Joining, by = "Keys"
```

1.0.4.2 V4

For the V4 region, we will use the same approach, extract amplicon region, filter extracted sequences based on amplicon length, generate pairwise distance matrix using a multiple sequence alignment, and then evaluate pairwise distances.

```
## Finding sequeces with forward primer
forward_match <- Biostrings::vmatchPattern("GTGCCAGCMGCCGCGGTAA",
                                           subject = paeni_16S$seq,
                                           fixed = FALSE) %>%
    as.list() %>% map_dfr(as.data.frame,.id = "seq_id")

## Finding sequences with reverse primer
reverse_match <- Biostrings::vmatchPattern("ATTAGAWACCCBDGTAGTCC",
                                           subject = paeni_16S$seq,
                                           fixed = FALSE) %>%
    as.list() %>% map_dfr(as.data.frame,.id = "seq_id")

## sequences with both forward and reverse primers
seqs_to_use_ids <- intersect(forward_match$seq_id, reverse_match$seq_id)
seqs_to_use <- names(paeni_16S$seq) %in% seqs_to_use_ids

## Extract amplicon region
paeni_V4 <- TrimDNA(paeni_16S$seq[seqs_to_use],
                                leftPatterns = "GTGCCAGCMGCCGCGGTAA",
                                rightPatterns = "ATTAGAWACCCBDGTAGTCC",
                                type = "both")
```

```
## Finding left pattern: 100% internal, 0% flanking
##
## Finding right pattern: 100% internal, 0% flanking
##
## Time difference of 0.17 secs
```

```
## Excluding seqs with lenght 0
paeni_V4_seqs <- paeni_V4[[2]][width(paeni_V4[[2]]) != 0]

### Calculate distance matrix from multiple sequence alignment
v4_align <- AlignSeqs(paeni_V4_seqs, verbose = FALSE)
v4_dist <- DistanceMatrix(v4_align,
                          correction = "none",
                          verbose = FALSE,
                          includeTerminalGaps = FALSE)

## Creating a data frame for exploratory analysis
v4_dist_df <- v4_dist %>%
    as.data.frame() %>%
    rownames_to_column(var = "Keys") %>%
    gather("Keys2","distance", -Keys) %>%
    mutate(Keys = as.numeric(Keys), Keys2 = as.numeric(Keys2)) %>%
    filter(Keys < Keys2) %>%
    mutate(Keys = as.character(Keys), Keys2 = as.character(Keys2))

tax_df <- dplyr::select(paeni_16S$taxa, "Keys", "Species")
v4_dist_anno_df <- v4_dist_df %>%
    left_join(tax_df) %>%
    left_join(tax_df,by = c("Keys2" = "Keys")) %>%
    dplyr::rename(Keys_Species = Species.x, Keys2_Species = Species.y) %>%
    mutate(group_comp = if_else(Keys_Species == Keys2_Species,
                                "within","between")) %>%
    filter(Keys_Species != "s__", Keys2_Species != "s__")
```

```
## Joining, by = "Keys"
```

```
## Excluding outlier sequence "329842" - mean pairwise distance to all other
## sequences is 0.2
 v4_dist_anno_filt <- filter(v4_dist_anno_df,
                             Keys != "329842",
                             Keys2 != "329842")
```

1.0.4.3 Amplicon Sequence Lengths

The trimmed sequence length varies for forward and reverse primers resulting in varying amplicon sizes for both the V12 and V4 amplicons (Fig. 2).

```
list(V12 = paeni_V12[[1]], V4 = paeni_V4[[1]]) %>%
    map_dfr(as.data.frame, .id = "amplicon") %>%
    filter(width != 0) %>%
    gather("key","value", -names, -amplicon) %>%
    mutate(key = if_else(key == "width", "length", key),
           key = factor(key, levels = c("length","start","end"))) %>%
    mutate(position_type = if_else(key == "length",
                                   "length", "position")) %>%
```

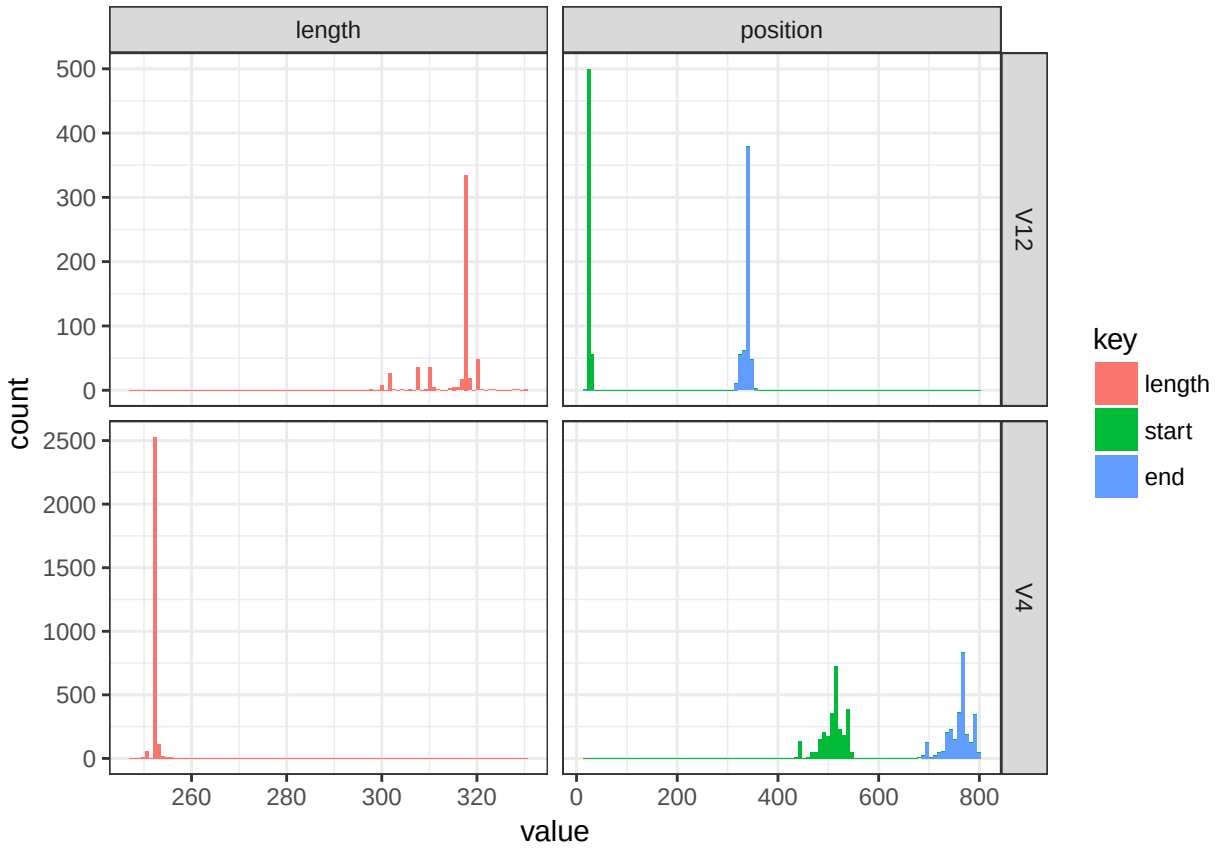

Figure 2: Primer trimmed sequence, amplicon, length and start and end positions relative to full length sequences for the V12 and V4 regions.

```
    ggplot() +
    geom_histogram(aes(x = value, fill = key), bins = 100) +
    facet_grid(amplicon~position_type, scales = "free") +
    theme_bw()
```

1.0.4.3.1 Genus Level Comparison

Pairwise distance is significantly different for within and between species comparisons indicating that the V12 and V4 regions are potentially suitable forclassifying members of the *Paenibacillus* genus to the species level (Fig. 3). Overall the V12 region had greater pairwise distances than V4 for both within and between species. It is important also to consider that the majority of sequences in the database were only classified to the genus level. Species-level information for these sequences might yield results that are inconsistent with our analysis. Additionally, our analysis does not identify the pairwise sequence distance required to classify a sequence as a novel *Paenibacillus* species.

```
list("V4" = v4_dist_anno_filt, "V12" = v12_dist_anno_df) %>%
    bind_rows(.id = "amplicon")  %>%
    ggplot(aes(x = group_comp, y = distance, fill = group_comp)) +
    geom_boxplot(aes(x = group_comp, y = distance, fill = group_comp)) +
    facet_wrap(~amplicon) +
    labs(x = "Amplicon", fill = "Species Comparison") +
        theme_bw() +
    theme(legend.position = "bottom")
```

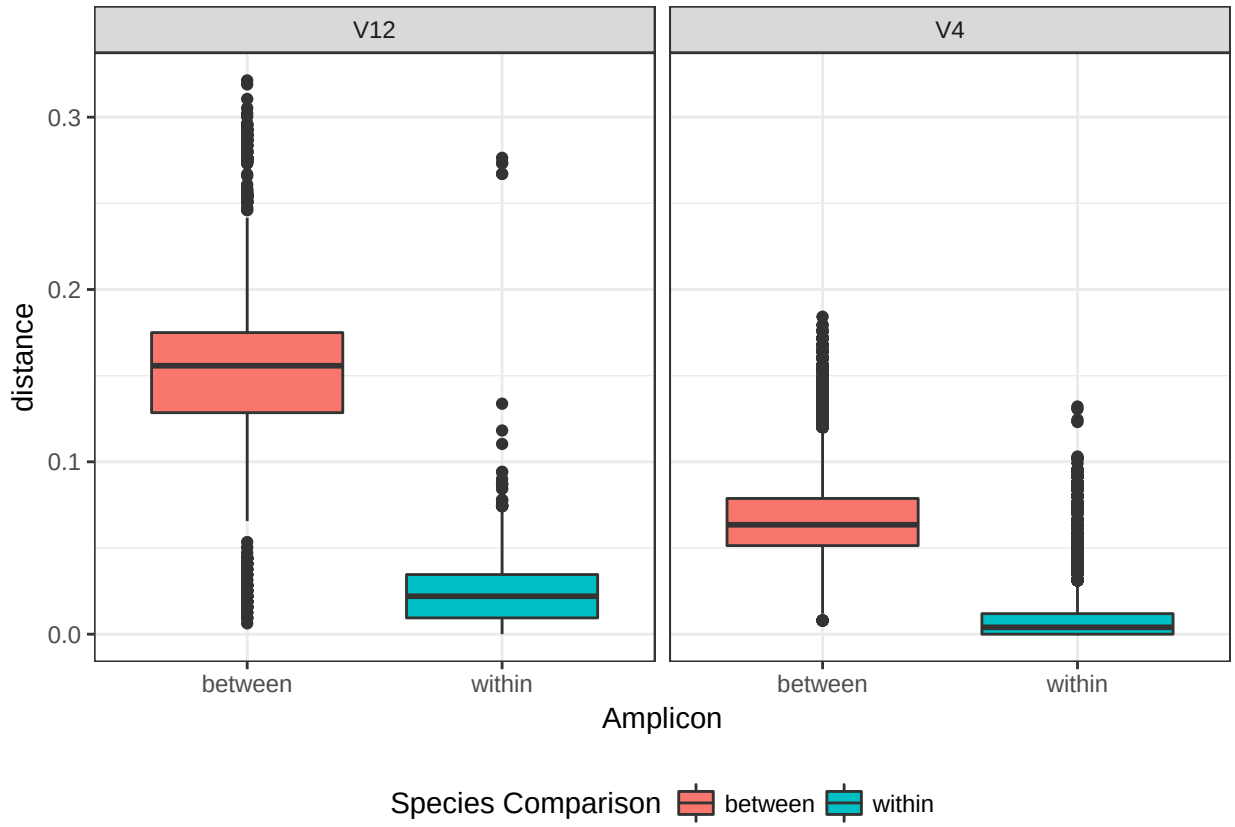

Figure 3: Distribution of within and between species pairwise distances for the V4 16S rRNA region. Sequences not classified to the species level were excluded from the analysis.

1.0.4.3.2 Species level comparison

While the overall pairwise distance is greater between species than within species for the *Paenibacillus* genus, it is important to understand how the within and between species pairwise distances compare for individual species. The heatmap below shows pairwise distance information for within and between different *Paenibacillus* species for the V12 and V4 regions (Fig. 4). Whether the sequences are assigned to more than one OTU depends on the pairwise sequence distance metric and linkage method employed by the clustering algorithm. In general though for species levels classification the maximum within species distance should be less than the minimum between species distance. For example as the maximum within species pairwise distance for *P. lentimorbus* is 0.13 and the minimum between species pairwise distance for *P. lentimorbus* and *P. alvei* is 0.08 (Fig. 4A), correctly assigning a V12 amplicon sequences to one of these two species is not possible.

```
v12_heat <- v12_dist_anno_df %>%
    group_by(Keys_Species, Keys2_Species, group_comp) %>%
    summarise(mean_dist = mean(distance, na.rm = TRUE),
              max_dist = max(distance),
              min_dist = min(distance)) %>%
    mutate(lab_dist = if_else(group_comp == "within",
                              max_dist, min_dist)) %>%
    ungroup() %>%
    mutate(Keys_Species = str_remove(Keys_Species, "s__")) %>%
```

```
    mutate(Keys2_Species = str_remove(Keys2_Species, "s__")) %>%
    ggplot() +
    geom_raster(aes(x = Keys_Species,
                    y = Keys2_Species,
                    fill = mean_dist)) +
    geom_text(aes(x = Keys_Species, y = Keys2_Species,
                  label = round(lab_dist,2)), size = 2, color = "white") +
    labs(x = "Species",y = "Species", fill = "Mean Dist") +
    theme_bw() +
    theme(axis.text.x = element_text(angle = -45, hjust = -0.1))

v4_heat <- v4_dist_anno_df %>%
    filter(Keys != "329842", Keys2 != "329842") %>%
    group_by(Keys_Species, Keys2_Species, group_comp) %>%
    summarise(mean_dist = mean(distance),
              max_dist = max(distance),
              min_dist = min(distance)) %>%
    mutate(lab_dist = if_else(group_comp == "within",
                              max_dist, min_dist)) %>%
    ungroup() %>%
    mutate(Keys_Species = str_remove(Keys_Species, "s__")) %>%
    mutate(Keys2_Species = str_remove(Keys2_Species, "s__")) %>%
    ggplot() +
    geom_raster(aes(x = Keys_Species,
                    y = Keys2_Species,
                    fill = mean_dist)) +
    geom_text(aes(x = Keys_Species, y = Keys2_Species,
                  label = round(lab_dist,2)), size = 2, color = "white") +
    labs(x = "Species",y = "Species", fill = "Mean Dist") +
    theme_bw() +
    theme(axis.text.x = element_text(angle = -45, hjust = -0.1))

ggpubr::ggarrange(v12_heat, v4_heat,labels = "AUTO",
                  ncol = 1, nrow = 2,align = "v")
```

1.0.5 Conclusion

Here we demonstrate how the *metagenomeFeatures* package in conjunction with one of the associated 16S rRNA database packages, *greengenes13.5MgDb*, and other R packages, can be used to evaluate whether species-level taxonomic classification is possible for a specific amplicon region. The approach used here can easily be extended to use different 16S rRNA databases (starting with a different `MgDb`class object), taxonomic groups (changing filtering parameters), or amplicon regions (changing primer sequences).

1.1 Session Information

1.1.1 System Information

```
s_info <- devtools::session_info(include_base = FALSE)
pander::pander(s_info$platform)
```

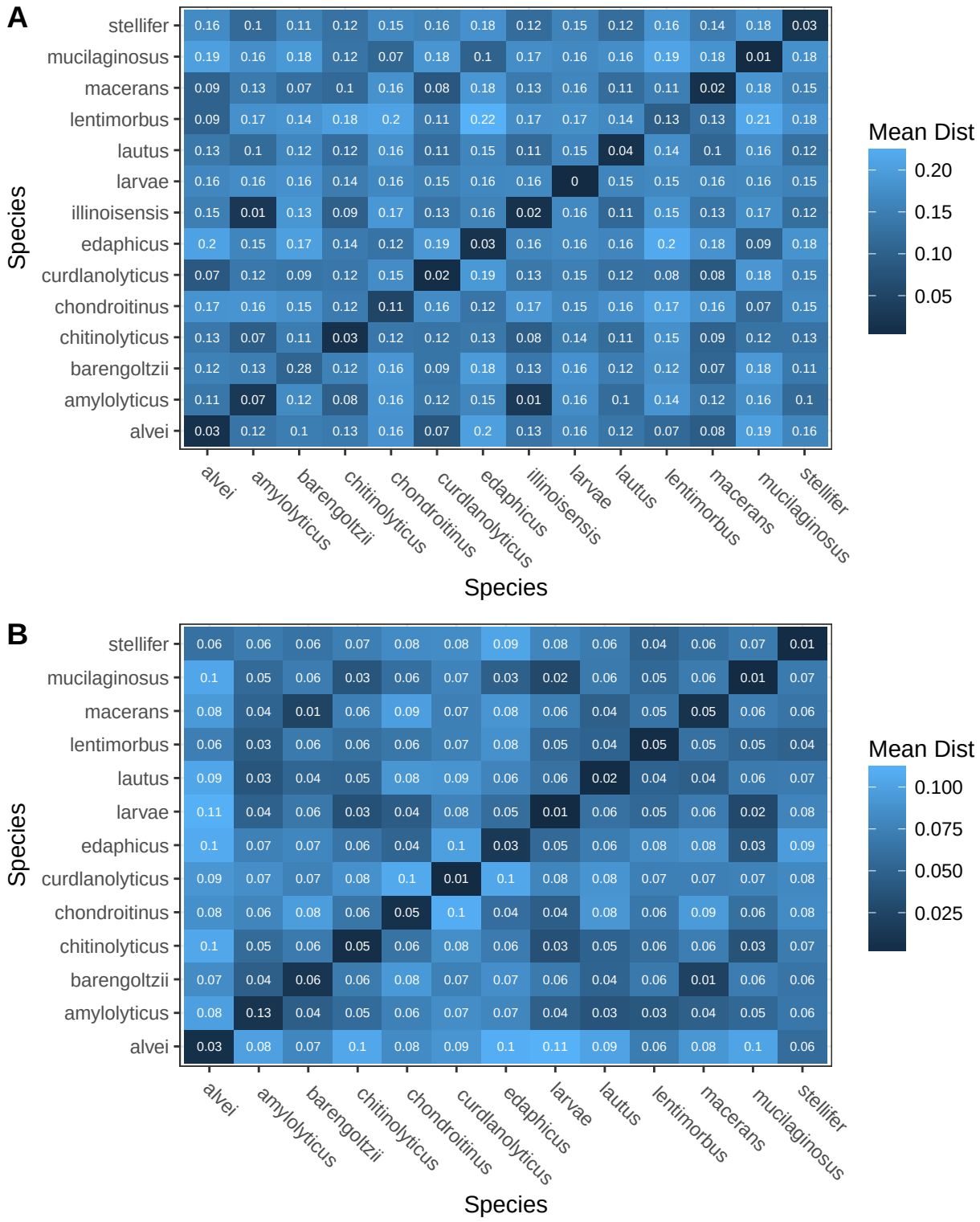

Figure 4: Pairwise distances between *Paenibacillus* species (A) V12 and (B) V4 amplicon regions. Fill color indicates the mean pairwise sequence distance within and between species. The text indicates the maximum pairwise distance for within-species comparisons, values along the diagonal, and maximum pairwise distance for between species comparisons. Different number of species are included in the V12 and V4 plots as there are no full-length *P. chondroitinus* sequences with the V12 primer in the database.

```
## Warning in pander.default(s_info$platform): No pander.method for
## "platform_info", reverting to default.
```

- **version**: R version 3.5.0 (2018-04-23)
- **system**: x86_64, darwin16.7.0
- **ui**: unknown
- **language**: (EN)
- **collate**: en_US.UTF-8
- **tz**: America/New_York
- **date**: 2018-05-23

1.1.2 Package Versions

```
s_info$packages %>% filter(`*` == "*") %>% select(-`*`) %>%
    knitr::kable(booktabs = TRUE)
```

package	version	date	source
base	3.5.0	2018-05-02	local
bindrcpp	0.2.2	2018-03-29	CRAN (R 3.5.0)
Biobase	2.40.0	2018-05-08	Bioconductor
BiocGenerics	0.26.0	2018-05-08	Bioconductor
Biostrings	2.48.0	2018-05-08	Bioconductor
datasets	3.5.0	2018-05-02	local
DECIPHER	2.8.1	2018-05-17	Bioconductor
dplyr	0.7.4	2017-09-28	CRAN (R 3.5.0)
forcats	0.3.0	2018-02-19	CRAN (R 3.5.0)
ggplot2	2.2.1	2016-12-30	CRAN (R 3.5.0)
ggpubr	0.1.6	2017-11-14	CRAN (R 3.5.0)
graphics	3.5.0	2018-05-02	local
grDevices	3.5.0	2018-05-02	local
greengenes13.5MgDb	2.0.0	2018-05-08	local (HCBravoLab/greengenes13.5MgDb@NA)
IRanges	2.14.10	2018-05-17	Bioconductor
magrittr	1.5	2014-11-22	CRAN (R 3.5.0)
metagenomeFeatures	2.0.0	2018-05-08	Bioconductor
methods	3.5.0	2018-05-02	local
parallel	3.5.0	2018-05-02	local
purrr	0.2.4	2017-10-18	CRAN (R 3.5.0)
readr	1.1.1	2017-05-16	CRAN (R 3.5.0)
RSQLite	2.1.1	2018-05-06	CRAN (R 3.5.0)
S4Vectors	0.18.2	2018-05-17	Bioconductor
stats	3.5.0	2018-05-02	local
stats4	3.5.0	2018-05-02	local
stringr	1.3.0	2018-02-19	CRAN (R 3.5.0)
tibble	1.4.2	2018-01-22	CRAN (R 3.5.0)
tidyr	0.8.0	2018-01-29	CRAN (R 3.5.0)
tidyverse	1.2.1	2017-11-14	CRAN (R 3.5.0)
utils	3.5.0	2018-05-02	local
XVector	0.20.0	2018-05-08	Bioconductor