1.quality_control

Date: October 23, 2019 Author: Ciera Martinez

Purpose

The purpose of this notebook is to perform quality control on the data to understand exactly what we have outputed.

Terminolgy

- **TFBS**: Transcription Factor Binding Site. A DNA motif that is known to bind to Transciption Factors (protiens), which has been shown to be a mechanism to direct gene transcription processes.
- **alignment**: nucleotides aligned based on similarity. You can view the alignment in the alignment files located https://drive.google.com/open?id=1UEXg0QMDFKIrvwnTxo64t2AWseYOCfD9
- **orthlogous regions**: Refering to part of the alignment that is shared across the species.
- **orthologous TFBS region**: A part of the alignment that spans a called motif. In this case the Orthologous TFBS Region is always 6 base pairs long.
- **called TFBS**: these are the TFBS that have a high enough scor (in this case above 7) to be identified as a likely biologically active transcriotion binding site.

```
## Libraries
## Read in cleaned data

library(reshape2)
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------ tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts --------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2); theme_set(theme_bw())

dataset1 <- read.csv("../data/all_data_bcd_2019_10_01_clean.csv")
head(dataset1)
```

```
##     strand align_position      score  region enhancer_func  species
## 1 positive            972 -8.1578407 VT14010             0 MEMB002A
## 2 positive            972 -0.4367419 VT14010             0 MEMB002B
## 3 positive            972 -8.1578407 VT14010             0 MEMB002C
## 4 positive            972  7.3511605 VT14010             0 MEMB002D
## 5 positive            972  3.8917291 VT14010             0 MEMB002E
## 6 positive            972 -0.4367419 VT14010             0 MEMB002F
##   raw_position raw_seq before_seq after_seq TFBS_called
## 1          376  CAACCT     AATTGC    AGCAAT          no
## 2          375  CAATCT     AATTGC    AGCAAT          no
## 3          376  CAACCT     AATTGC    AGCAAT          no
## 4          376  TAATCT     AATTGC    AGCAAT         yes
## 5          376  TAATCG     AATAGC    AGCTAT          no
## 6          376  CAATCT     AATTGC    AGCTAT          no
```

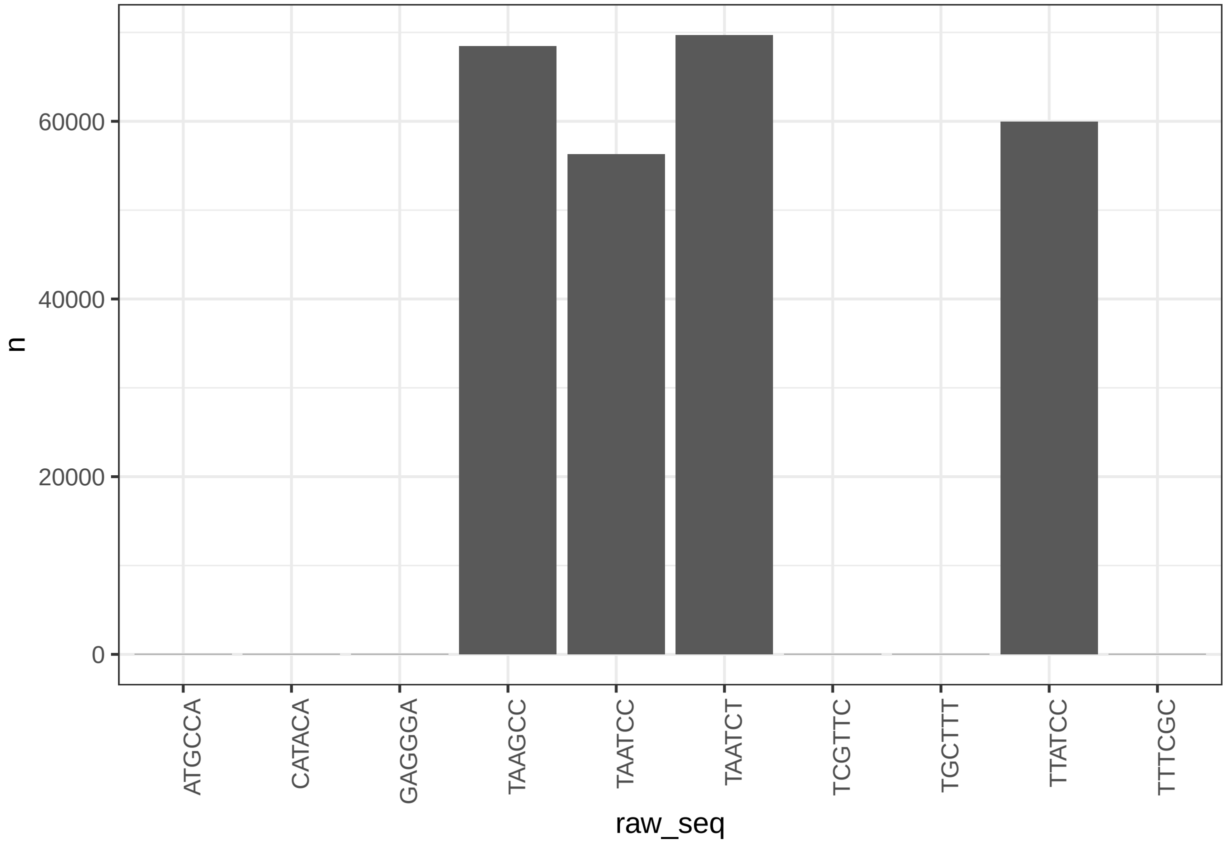
Part 1a: Testing scoring

Summary: Something is up here, yes, the main motifs, TAAGCC, TAATCC, TAATCT and TTATCC are represented mostly, but why and the hell are there two represenatives from these other categories? Looking closer at these weirdos, it seems like it all comes from the same region (VT40027) and the same species (MEMB005D). I will just remove this region, since it only occurs in VT40027.

```r
dim(dataset1)
```

```
## [1] 1367580      11
```

```r
dataset1 %>%
  filter(score >= 7) %>%
  group_by(raw_seq, TFBS_called) %>%
  tally() %>%
  ggplot(., aes(raw_seq,n)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



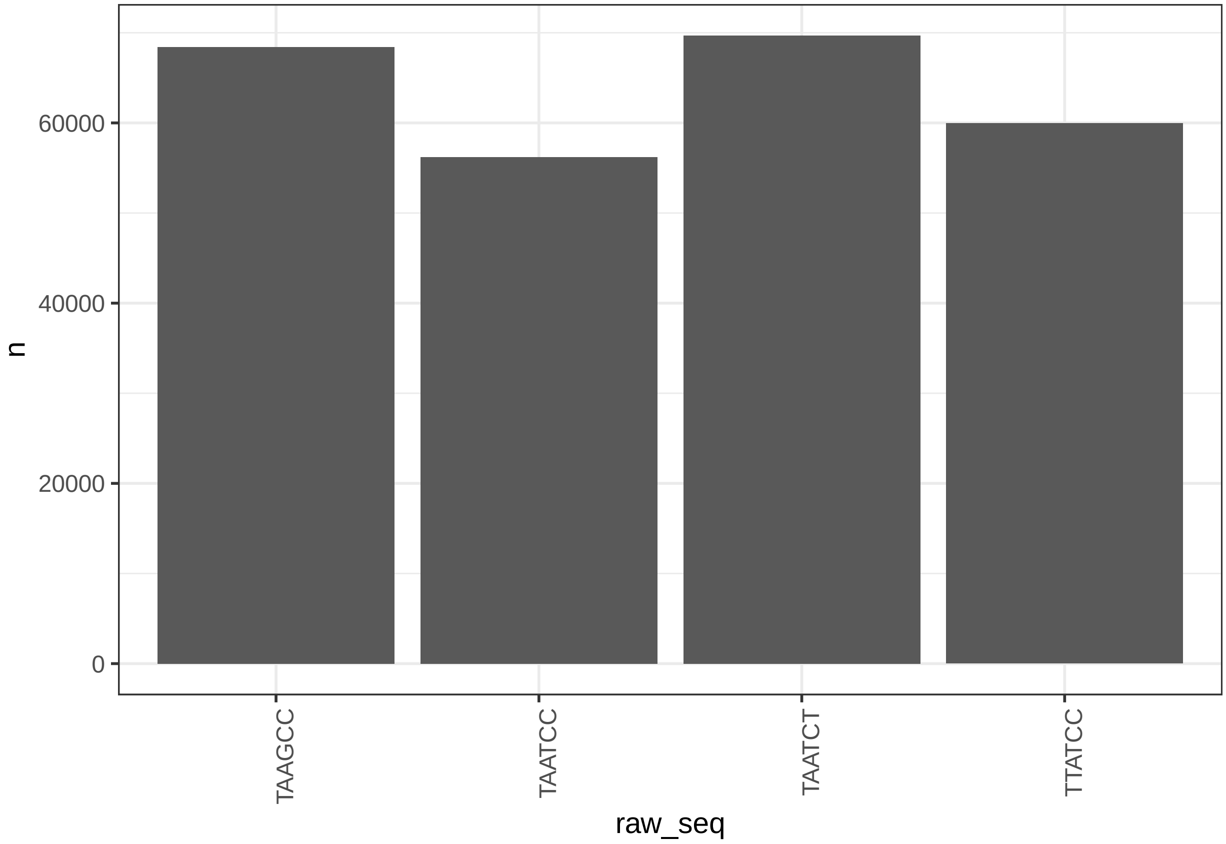
```
## For example:
weirdos <- c("ATGCCA", "CATACA", "GAGGGA", "TCGTTC", "TCGTTC", "TGCTTT", "TTCGC")

## Show me all the rows with the Weirdos
dataset1 %>%
  filter(score >= 7 & raw_seq %in% weirdos)
```

```
##      strand align_position     score  region enhancer_func  species
## 1  positive           2876 11.612828 VT40027             1 MEMB005D
## 2  negative           2938  7.351161 VT40027             1 MEMB005D
## 3  positive           2942 11.612828 VT40027             1 MEMB005D
## 4  negative             86  7.351161 VT40027             1 MEMB005D
## 5  negative             53  7.351161 VT40027             1 MEMB005D
## 6  positive           2271  7.351161 VT40027             1 MEMB005D
## 7  positive           2513  7.351161 VT40027             1 MEMB005D
## 8  positive           3018 11.612828 VT40027             1 MEMB005D
## 9  negative           3068  7.351161 VT40027             1 MEMB005D
## 10 positive           3072 11.612828 VT40027             1 MEMB005D
##    raw_position raw_seq before_seq after_seq TFBS_called
## 1          1982  TGCTTT     TTTTTT    TTTTTT         yes
## 2          2032  GAGGGA     GATCCT    ACGAAC         yes
## 3          2036  TCGTTC     TCTCCC    GTTTTT         yes
## 4            53  ATGCCA     AATGGG    TTTAAA         yes
## 5            53  ATGCCA     AATGGG    TTTAAA         yes
## 6          1645  CATACA     ATGGTA    TTTCGA         yes
## 7          1645  CATACA     ATGGTA    TTTCGA         yes
## 8          1982  TGCTTT     TTTTTT    TTTTTT         yes
## 9          2032  GAGGGA     GATCCT    ACGAAC         yes
## 10         2036  TCGTTC     TCTCCC    GTTTTT         yes
```

```
## Remove this weird region
dataset1 <- dataset1 %>%
  filter(region != "VT40027")

## Re- test with weird removed region
dataset1 %>%
  filter(score >= 7) %>%
  group_by(raw_seq, TFBS_called) %>%
  tally() %>%
  ggplot(., aes(raw_seq,n)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Part 1b: Testing Orthologous region grabbing

It looks to be working very well, except that sometimes it doesn't.

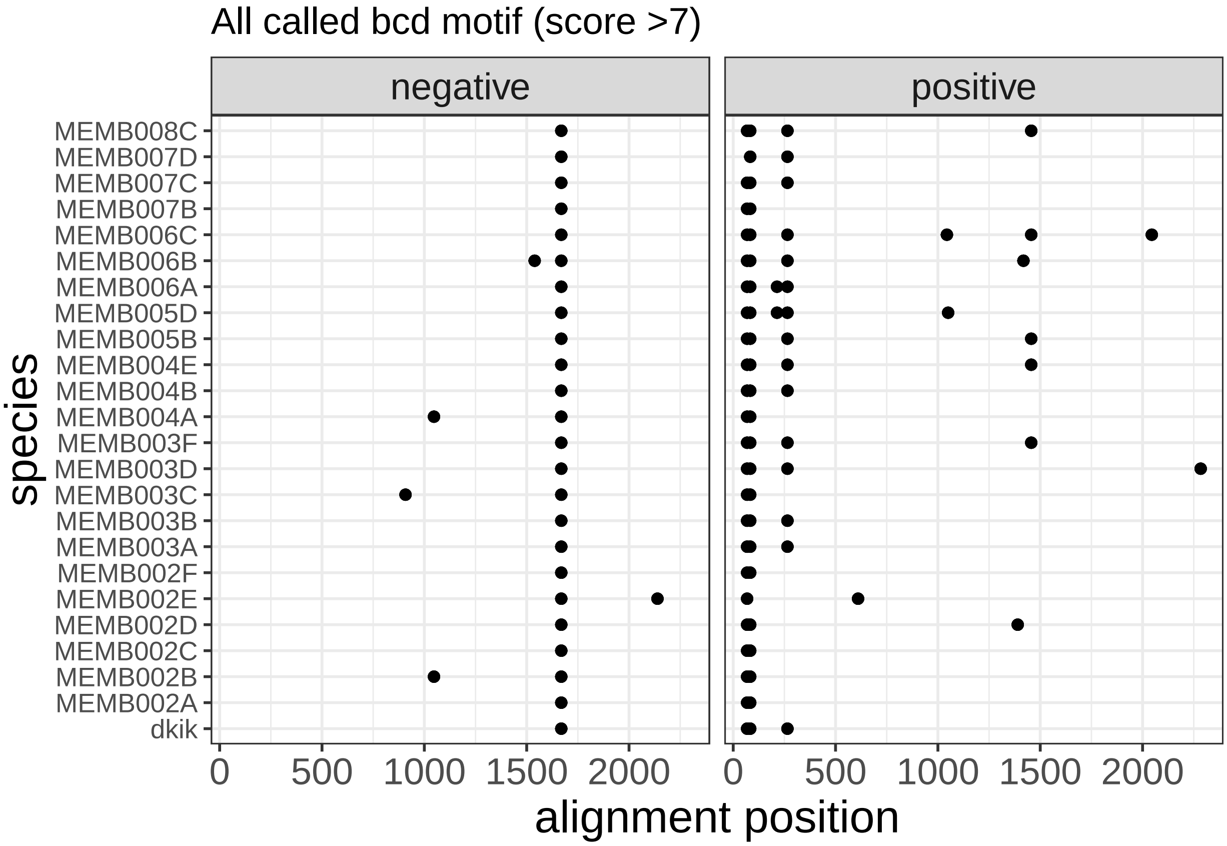
Example 1:

Problem: All orthologous regions of the sequences are not grabbed

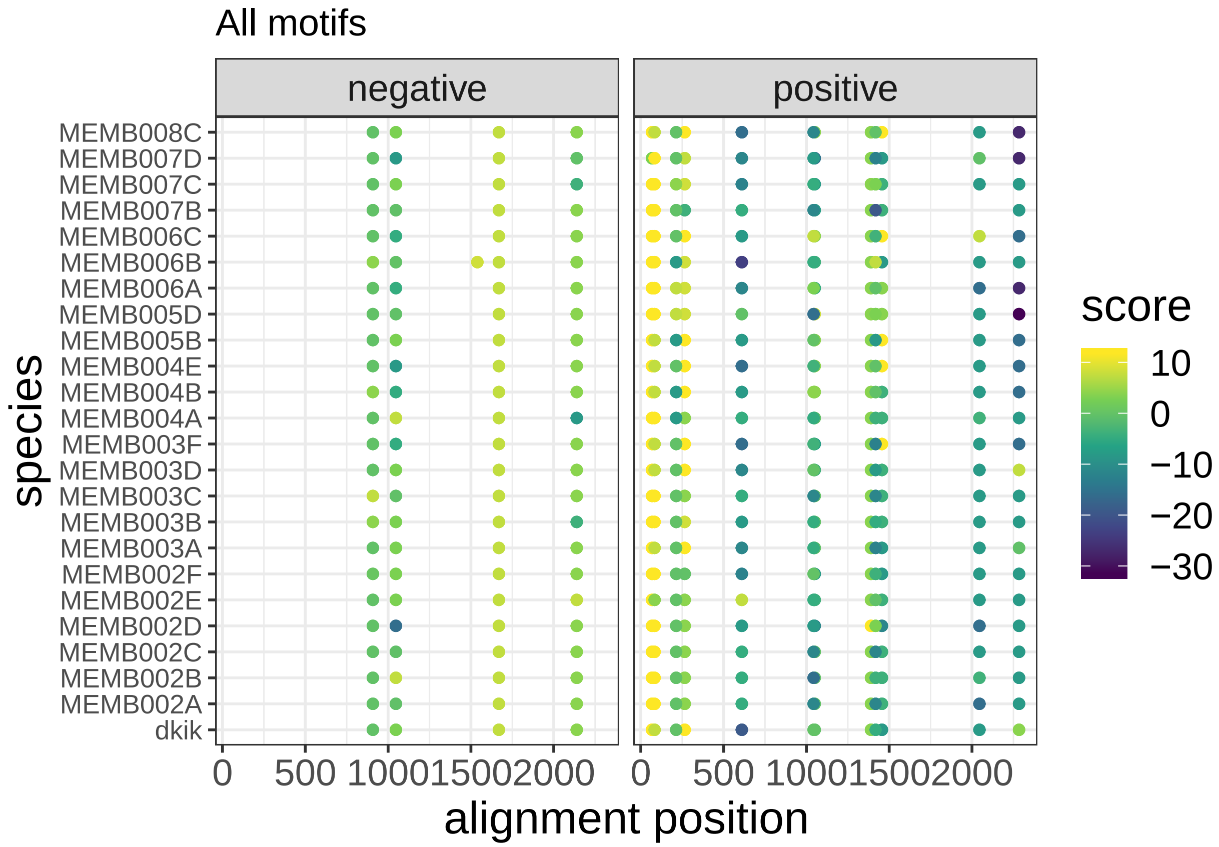
In the negative strand there is a lone called motif around align_position 1500, the other orthologous regions were not grabbed. This is likely due to that species (MEMB006B) having an inserted region.

```
### Looking at the "yeses"
## Scores look correct when using the called motif

## With filter
dataset1 %>%
  filter(region == unique(dataset1$region)[6] & TFBS_called == "yes") %>%
  ggplot(., aes(align_position, species)) +
  geom_point() + facet_grid(.~strand) +
    theme(text = element_text(size = 17),
          axis.text.y = element_text(size = 10),
        plot.title = element_text( size=14)) +
  labs(title="All called bcd motif (score >7)", x = "alignment position", y = "species")
```



```
## Without filter
dataset1 %>%
  filter(region == unique(dataset1$region)[6]) %>%
  ggplot(., aes(align_position, species, color = score)) +
  geom_point() + scale_color_viridis_c() + facet_grid(.~strand) +
      theme(text = element_text(size = 17),
      axis.text.y = element_text(size = 10),
      plot.title = element_text( size=14)) +
  labs(title="All motifs", x = "alignment position", y = "species")
```



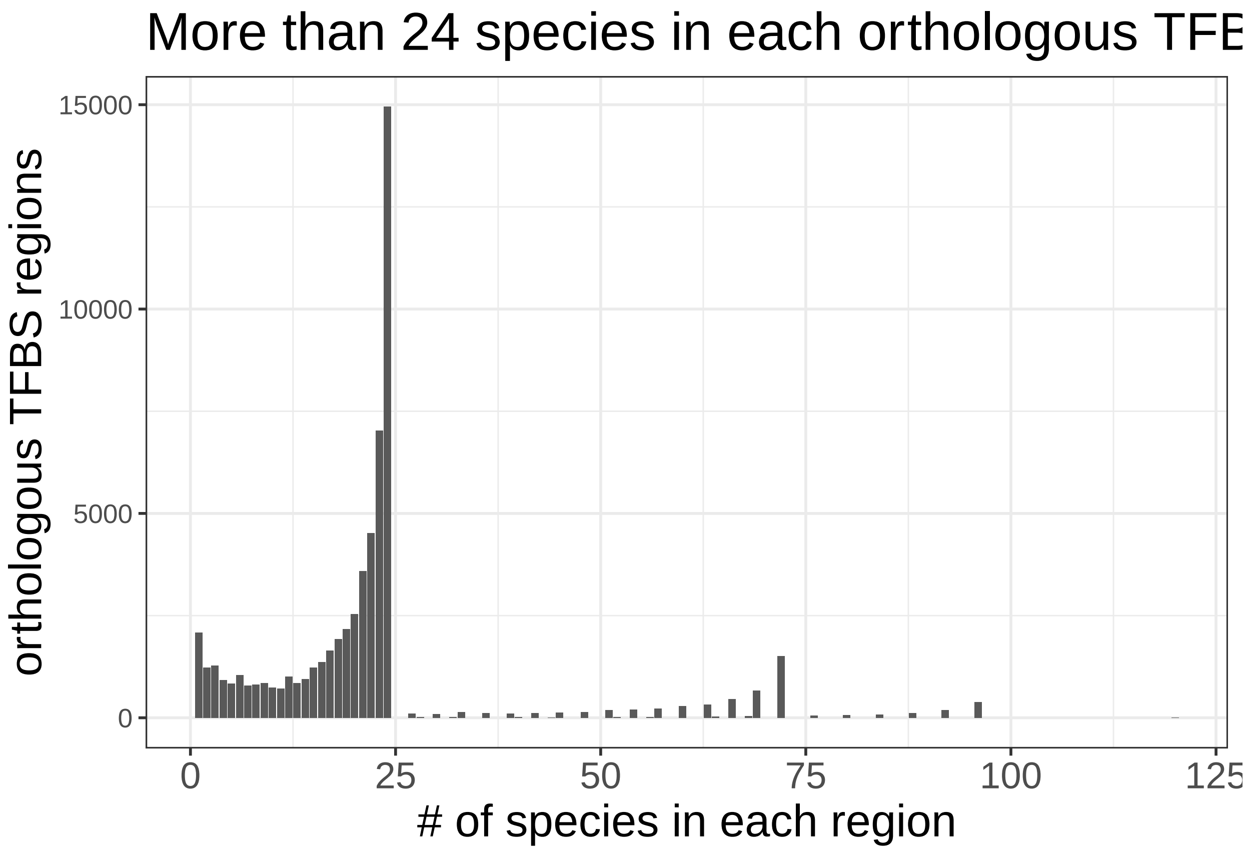
```
dataset1 %>%
  filter(region == unique(dataset1$region)[6] & species == "MEMB006B") %>%
  filter(strand == "negative" & TFBS_called == "yes")
```

```
##     strand align_position    score  region enhancer_func  species
## 1 negative           1669 7.351161 VT21534             0 MEMB006B
## 2 negative           1539 8.354094 VT21534             0 MEMB006B
## 3 negative           1669 7.351161 VT21534             0 MEMB006B
## 4 negative           1539 8.354094 VT21534             0 MEMB006B
## 5 negative           1669 7.351161 VT21534             0 MEMB006B
## 6 negative           1539 8.354094 VT21534             0 MEMB006B
## 7 negative           1669 7.351161 VT21534             0 MEMB006B
## 8 negative           1539 8.354094 VT21534             0 MEMB006B
##   raw_position raw_seq before_seq after_seq TFBS_called
## 1         1590  TAATCT     CATAAT    TTTGGT         yes
## 2         1463  TTATCC     GCGCCG    GTGTCG         yes
## 3         1590  TAATCT     CATAAT    TTTGGT         yes
## 4         1463  TTATCC     GCGCCG    GTGTCG         yes
## 5         1590  TAATCT     CATAAT    TTTGGT         yes
## 6         1463  TTATCC     GCGCCG    GTGTCG         yes
## 7         1590  TAATCT     CATAAT    TTTGGT         yes
## 8         1463  TTATCC     GCGCCG    GTGTCG         yes
```

Problem: Example 1 brings up another problem, **there are duplicate rows in this example**. How many identical rows are there and is there a pattern to this? This could be easily fixed by removing duplicate rows, but we should talk to Niharika.

For example see the graph below:

```
dataset1 %>%
  group_by(region, align_position) %>%
  tally() %>%
  group_by(n) %>%
  tally() %>%
  ggplot(., aes(n, nn)) +
  geom_bar(stat = "identity") +
      theme(text = element_text(size = 17),
      axis.text.y = element_text(size = 10),
      plot.title = element_text( size=20)) +
 labs(title="More than 24 species in each orthologous TFBS region", y = "orthologous TFBS regions", x =
```



Now let's check how many duplicated rows there are in our dataset?

```
# Check original
nrow(dataset1)
```

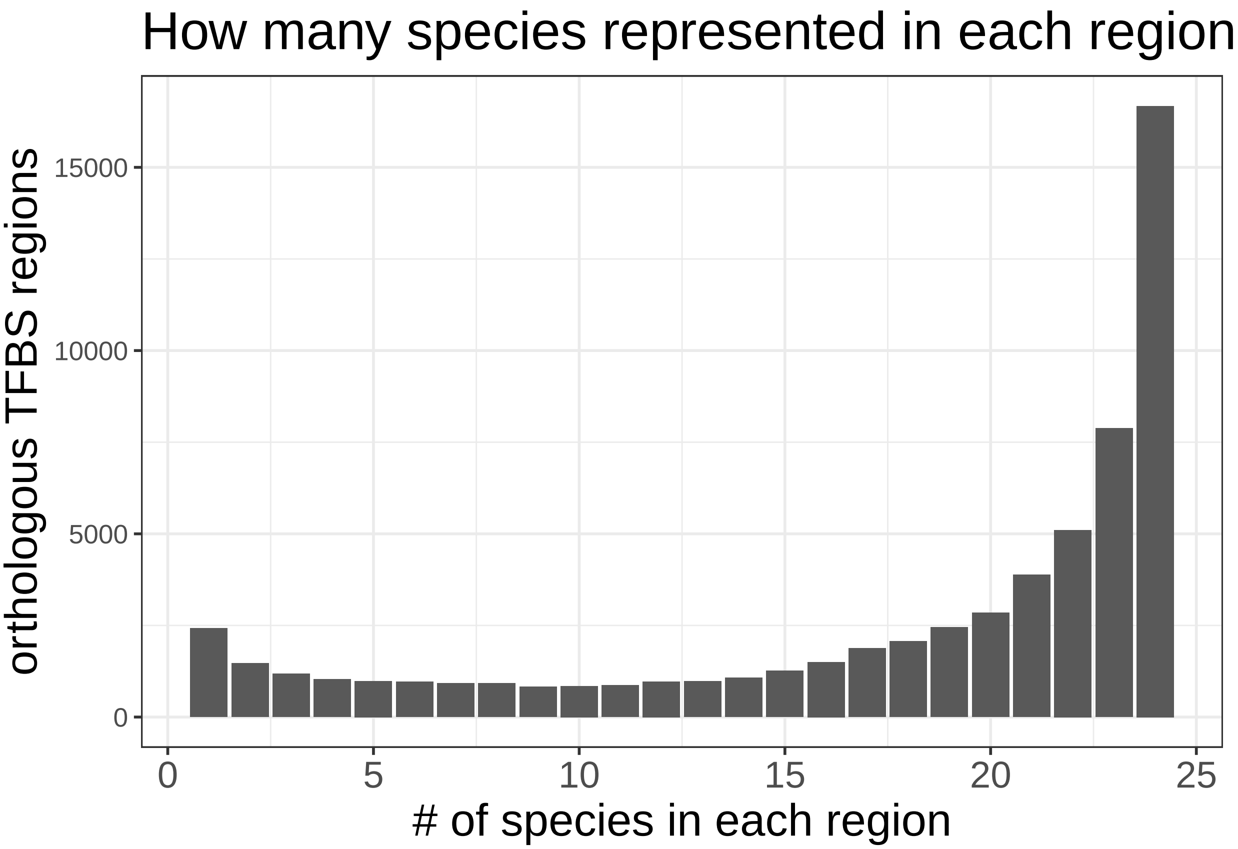
```
## [1] 1366706
```

```
## remove duplicated rows
dataset2 <- dataset1 %>%
  distinct()
```

```
## How many were removed?
nrow(dataset1) - nrow(dataset2)
```

```
## [1] 273798
```

```
## Visualize
dataset2 %>%
  group_by(region, align_position, strand) %>%
  tally() %>%
  group_by(n) %>%
  tally() %>%
  ggplot(., aes(n, nn)) +
  geom_bar(stat = "identity") +
      theme(text = element_text(size = 17),
      axis.text.y = element_text(size = 10),
      plot.title = element_text( size=20)) +
  labs(title="How many species represented in each region", y = "orthologous TFBS regions", x = "# of sp
```



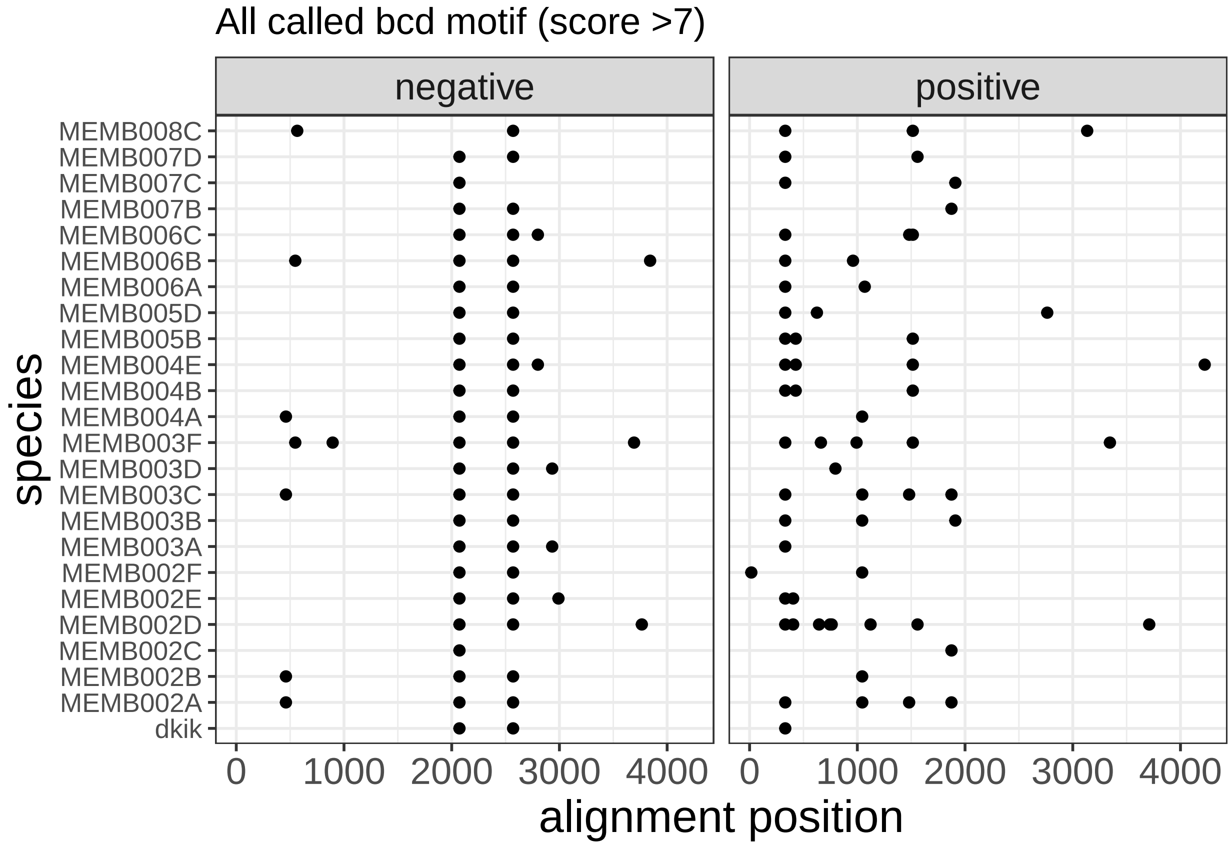
Conclusion: There were 273,798 duplicated motif represented in this data.

Next Steps: Just remove for now, but have Niharika fix in the pipeline.

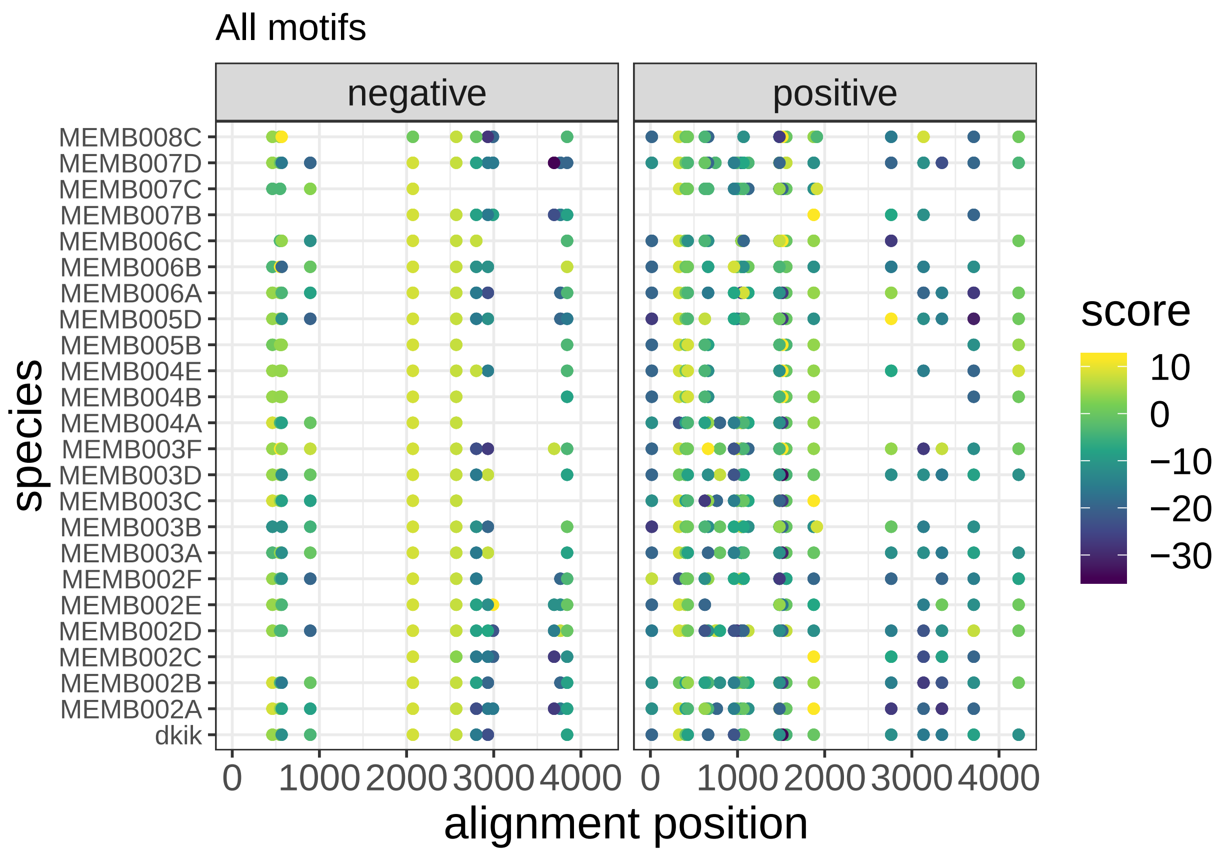
Example 2

This alignment has a lot going on. First off, there are a lot of Bcd TFBS being called. It really shows that there must be some correlation with how conserved the alignments are. It would be great to score each alignment position based on conservation. This might be need to normalized somehow at some point, but at this point it is a low priority.

```
## With filter
dataset2 %>%
  filter(region == unique(dataset1$region)[7] & TFBS_called == "yes") %>%
  ggplot(., aes(align_position, species)) +
  geom_point() + facet_grid(.~strand) +
    theme(text = element_text(size = 17),
          axis.text.y = element_text(size = 10),
        plot.title = element_text( size=14)) +
  labs(title="All called bcd motif (score >7)", x = "alignment position", y = "species")
```



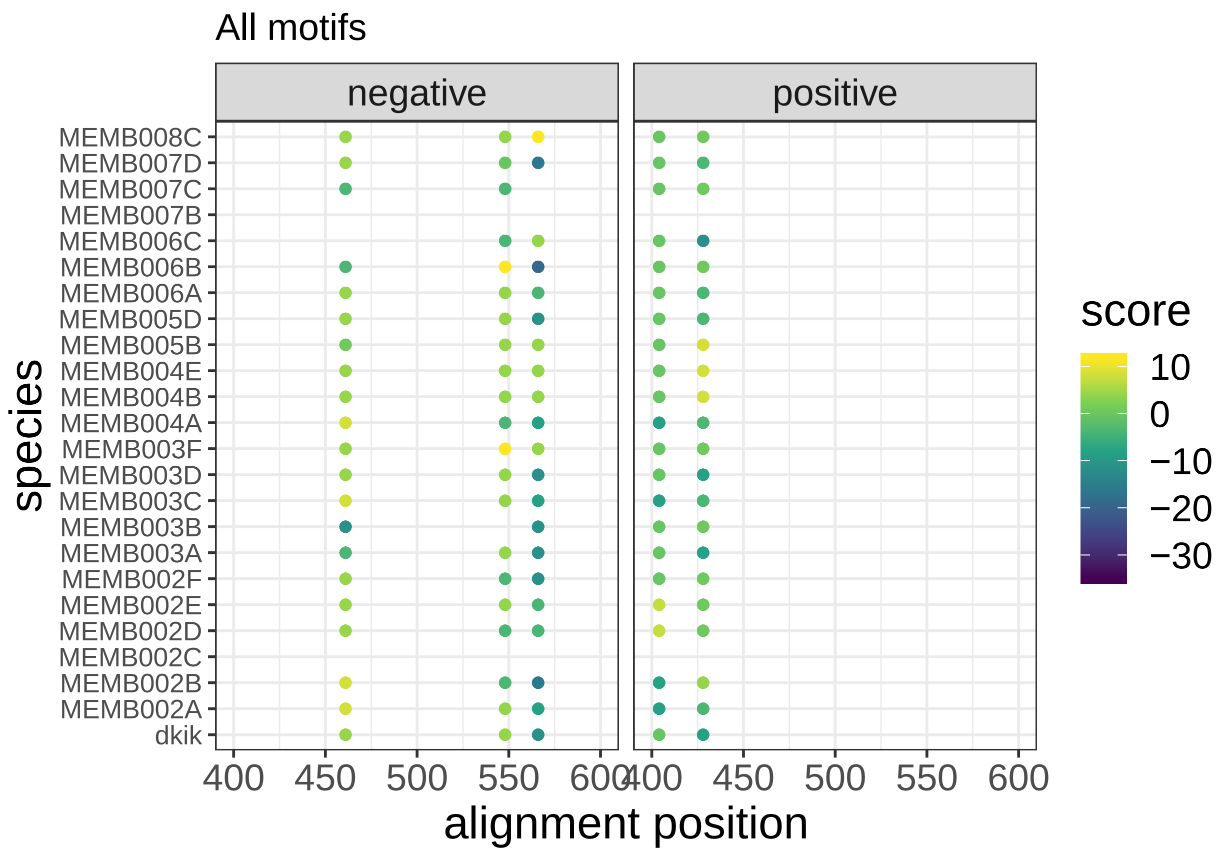
```
## Without filter
dataset1 %>%
  filter(region == unique(dataset1$region)[7]) %>%
  ggplot(., aes(align_position, species, color = score)) +
  geom_point() + scale_color_viridis_c() + facet_grid(.~strand) +
      theme(text = element_text(size = 17),
      axis.text.y = element_text(size = 10),
      plot.title = element_text( size=14)) +
  labs(title="All motifs", x = "alignment position", y = "species")
```



Lets take a closer look by zooming in.

```
dataset1 %>%
  filter(region == unique(dataset1$region)[7]) %>%
  ggplot(., aes(align_position, species, color = score)) +
  geom_point() + scale_color_viridis_c() + facet_grid(.~strand) +
      theme(text = element_text(size = 17),
      axis.text.y = element_text(size = 10),
      plot.title = element_text( size=14)) +
  labs(title="All motifs", x = "alignment position", y = "species")  +
  xlim(400, 600)
```

```
## Warning: Removed 2088 rows containing missing values (geom_point).
```

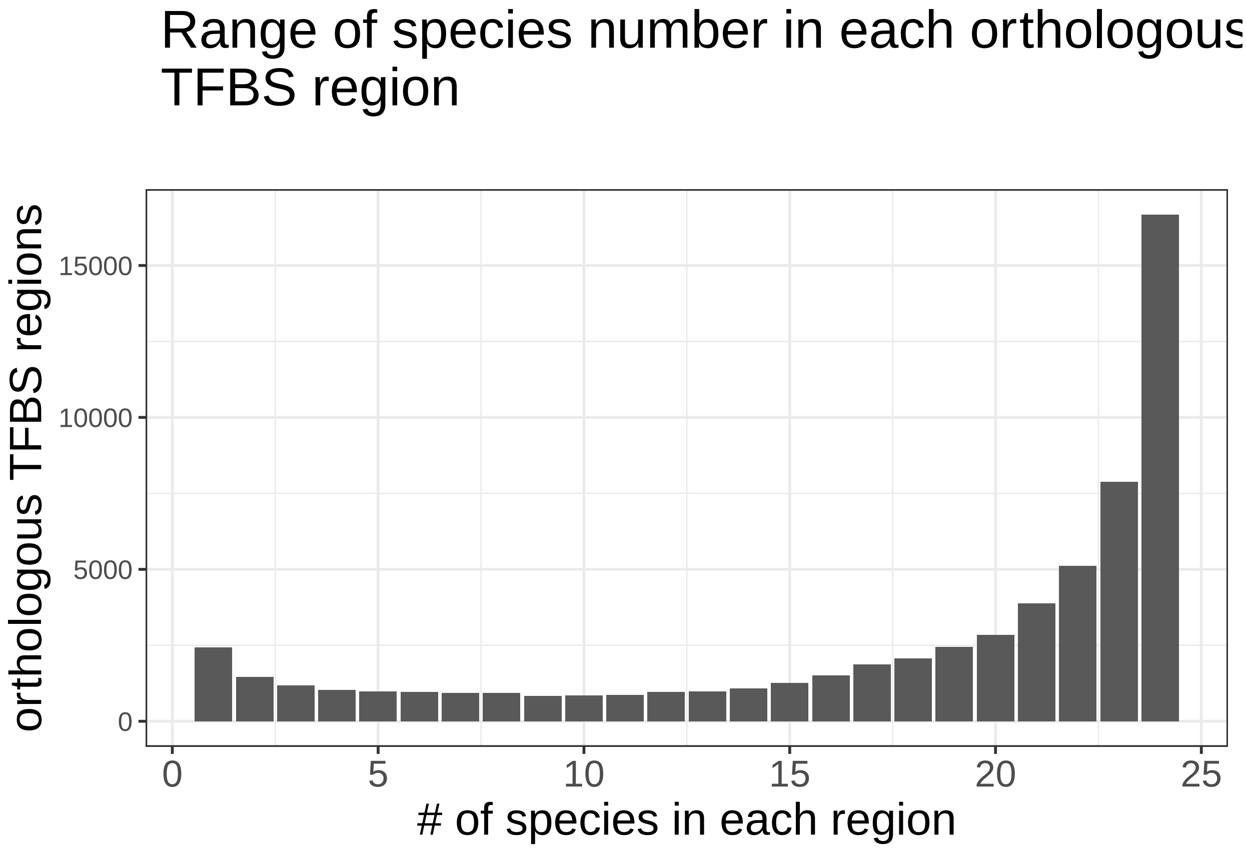


When looking closer then going to the alignment file, you see clearly that some of the regions are just going to be problems becauase of low conservation. It would be good to caluculate genetic variablity and rate of evolution across the entire region to get an idea of what is had been removed. Some questions that we really need to think about and understand are 1. **Can we only really use the dataset when we have 24 represenative regions?** I am thinking the answer is yes. 2. **What is the extent of gaps causing the problem and what is the extent of short sequences causing the problems?**. 3. **Are there certain species that could be removed that would greatly increase the orhtologous region datatset?** These last two questions really need to be explored. Let me do a quick look at what would happen if we removed all the regions that do not have all 24 species represented.

Part 2a: Looking into what the dataset would look like with only the representative species

Again, this is the distribution at the moment.

```
dataset2 %>%
  group_by(region, align_position) %>%
  tally() %>%
  group_by(n) %>%
  tally() %>%
  ggplot(., aes(n, nn)) +
  geom_bar(stat = "identity") +
   theme(text = element_text(size = 17),
      axis.text.y = element_text(size = 10),
      plot.title = element_text( size=20)) +
  labs(title=" Range of species number in each orthologous \n TFBS region \n", x = "# of species in eacl
```

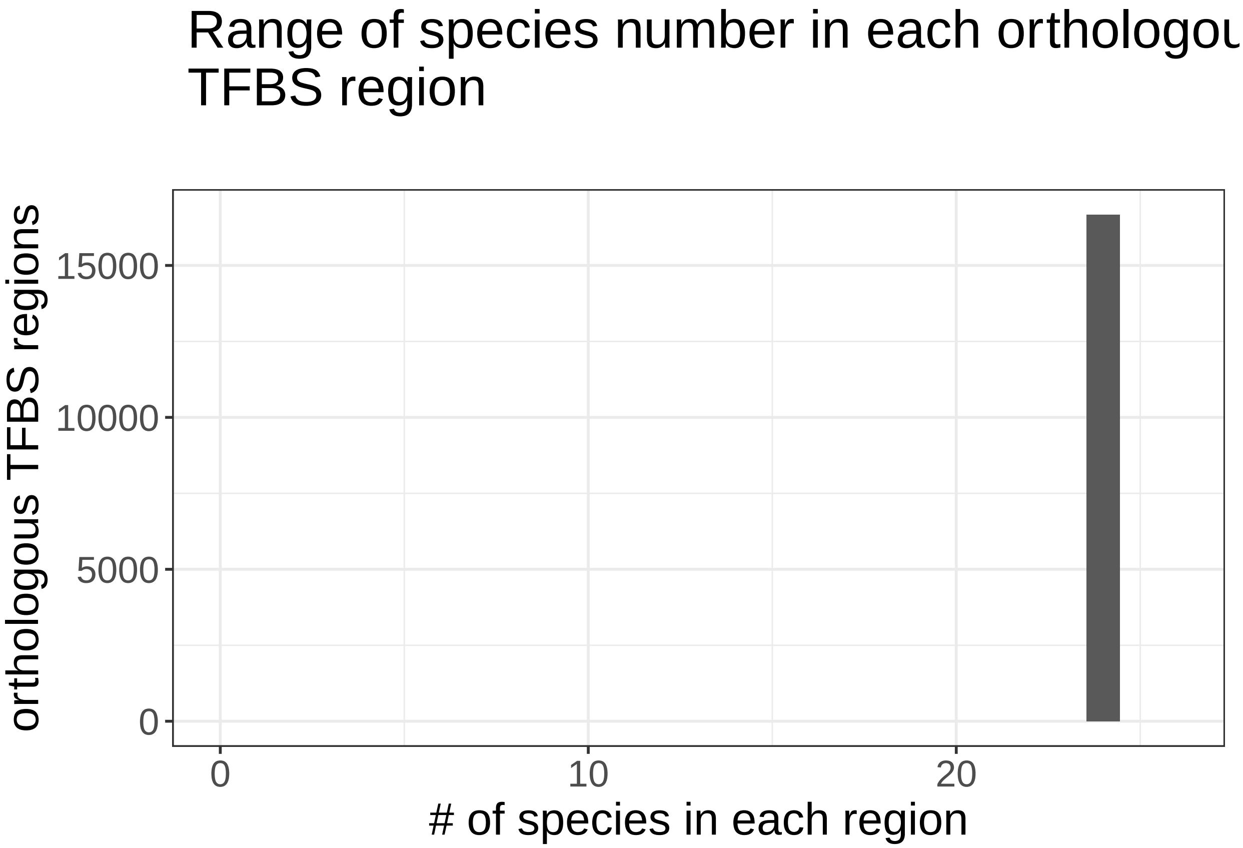


Now with represenative species removed

```
## First we need to filter just the ones that have 24 represenative species
## all the regions that will be used to filter by
## ps this filtering method rulz! `filter(n() == 24)`
## You can filter by grouping number or I assume anything else you calculate
## Reference: https://stackoverflow.com/questions/26573285/using-filter-with-count

## Takes a few min
dataset3 <- dataset2 %>%
  group_by(region, align_position, strand) %>%
  filter(n() == 24)

## Double check removal went well
dataset3 %>%
  group_by(region, align_position) %>%
  tally() %>%
  group_by(n) %>%
  tally() %>%
  ggplot(., aes(n, nn)) +
  geom_bar(stat = "identity") +
  xlim(0,26) +
  theme(text = element_text(size = 17),
        plot.title = element_text( size=20)) +
   labs(title=" Range of species number in each orthologous \n TFBS region \n", x = "# of species in eac
```



Summary: This shows how we gained 24 represenative species when we removed the duplicates. Now we have 16673, when before we had 14950, **gained 1723**.

Overall though I think this is a nice dataset where we have 16,673 orthologous TFBS regions.

Final quality control dataset

```
write.csv(dataset3, "../data/all_data_bcd_2019_10_01_after_QC.csv", row.names = FALSE)
```

Next Steps

Niharika

1. Trace Bug: There are duplicate rows in the data. See Part 1.B. Why? We need to figure out why. Is it a problem with `motif_extraction`? Or does it have to do with the input data?
2. We need to have controls for an upcoming experiment in which we test the rate of evolution (rate of nuceotide substituions) at each position of the TFBS. In order to do this, we need to compare with random 6bp nucleotide regions in each of the alignments. Can you use motif extraction to randomly isolate 20 6bp regions in each of the alignment files?

Zoe

1. Make sure you have the ability to view alignments. The program I use is Jalview.
2. Play around with the data to understand it better. Why are we missing species in orthologous regions? Is it always because of gaps? Look through a few more examples. Are there any patterns that are missed? Is it because the sequence is short? Or because there is a gap in the center?
3. Are there species that are preferentially missing from the orhtologous TFBS regions?
4. Look at the distribution of how many called bicoid TFBS sites (`TFBS_called == "yes"`) there are in each region. Do certain species have more or less than average?
5. Testing overlapp (coming soon). I will soon get you a new group of TFBS positions, hunchback (hb), you will need to explore position and categorize if they have overlapping positions.