

# Analysis of BGC content across phylum Cyanobacteriota

## Setup

### Read in data

```
# Map antiSMASH classes to their categories
bgc_class <- read_tsv("./data/2025-01-16-1256-bgc_class_ref.tsv") %>% select(!owner_id)
class_to_cat <- bgc_class$bgc_category
names(class_to_cat) <- bgc_class$class_name

# NCBI Taxonomy data for Cyanobacteriota assemblies
cyano_asm_tax <- read_tsv("data/cyano_asm_tax.tsv")

# From SMC: All antiSMASH 'region's for Cyanobacteriota genomes
regions <- read_tsv("./data/2025-02-26-1456-cyano_as_regions.tsv")

# Table of NCBI assemblies at "chromosome" or "complete" quality levels
ncbi_hiq_meta <- read_tsv("data/ncbi_cyano_HiQualityGenomes_metadata.tsv")
```

### Clean data

```
# Convert BGC 'class' to vector, add in BGC 'category' as vector, order levels of classes and categories
regions <- regions %>%
  mutate(classes = map(region_class, function(class_string) {
    (
      if (str_starts(class_string, fixed("["))) {
        fromJSON(class_string)
      } else {
        c(class_string)
      }
    )
  })) %>%
  mutate(categories = classes %>% map(function(cls_vec) {
    (
      map_vec(cls_vec, function(cls_str) class_to_cat[[cls_str]]) %>% unique() %>% sort()
    )
  })) %>%
  mutate(cats_str = categories %>% map_chr(function(x) str_flatten(x, collapse = ", "))) %>%
  add_count(cats_str) %>%
  mutate(
    cats_str = forcats::fct_reorder(cats_str, desc(n)),
    class_str = classes %>% map_chr(function(x) str_flatten(x, collapse = ", "))
  )
```

```
regions
```

```
## # A tibble: 32,112 x 26
##   region_gene_id bgc_id region_length contig_name region_start_nt region_end_nt
##   <dbl> <dbl> <dbl> <chr> <dbl> <dbl>
## 1 3944444400 2.05e6 20822 NZ_KK07376~ 1448721 1469542
## 2 427693693 2.21e6 6191 NZ_NMQI010~ 1 6191
## 3 427830749 2.21e6 41152 NZ_NJHU010~ 20319 61470
## 4 427867799 2.21e6 29334 NZ_NJHW010~ 5515 34848
## 5 427879170 2.21e6 5638 NZ_NJHW010~ 1 5638
## 6 442232549 2.29e6 27366 NZ_VIKX010~ 1 27366
## 7 442951361 2.29e6 61852 NZ_BJCK010~ 121631 183482
## 8 444447386 2.30e6 21923 NZ_WVIC010~ 13038 34960
## 9 445057742 2.31e6 40108 NZ_JAAGOG0~ 26815 66922
## 10 446681012 2.31e6 35928 NZ_QMEA010~ 1 35928
## # i 32,102 more rows
## # i 20 more variables: bgc_annotation_id <dbl>, region_class <chr>,
## # region_category <lgl>, contig_edge <lgl>, smc_id <dbl>, accession_id <chr>,
## # size_bp <dbl>, n_scaffolds <dbl>, data_source_description <chr>,
## # tax_phylum <chr>, tax_class <chr>, tax_order <chr>, tax_family <chr>,
## # tax_genus <chr>, tax_species <chr>, classes <list>, categories <list>,
## # cats_str <fct>, n <int>, class_str <chr>
```

## Explore data

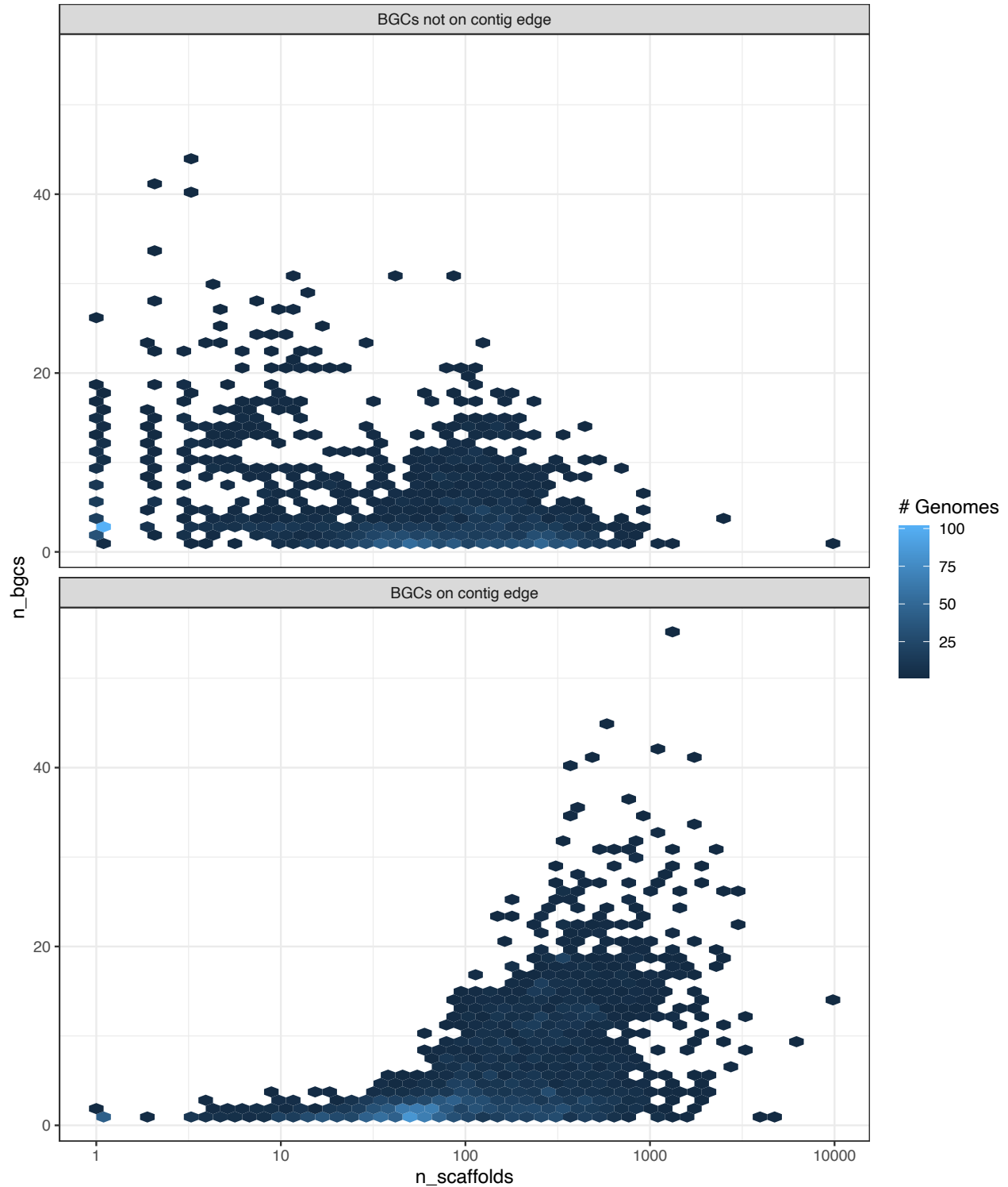
The repetitive composition of many BGCs makes them a challenge during genome assembly, resulting in over-inflation of BGC counts when BGCs are split between the ends of two different contigs. Focusing on high-quality genomes can therefore ensure a higher-quality dataset.

**How fragmented are the full set of genomes, and how does that impact BGC counts?**

```
regions %>%
  group_by(smc_id, n_scaffolds, contig_edge) %>%
  summarize(n_bgcs = n()) %>%
  ungroup() %>%
  ggplot(aes(x = n_scaffolds, y = n_bgcs)) +
  stat_bin_hex(bins = 50) +
  scale_x_log10(breaks = breaks_log()) +
  guides(fill = guide_colorbar(title = "# Genomes")) +
  facet_wrap(. ~ contig_edge, ncol = 1, labeller = as_labeller(c("FALSE" = "BGCs not on contig edge", "
  theme_bw() +
  ggtitle("Fragmented genomes have inflated BGC counts", subtitle = "Full dataset")
```

## Fragmented genomes have inflated BGC counts

Full dataset



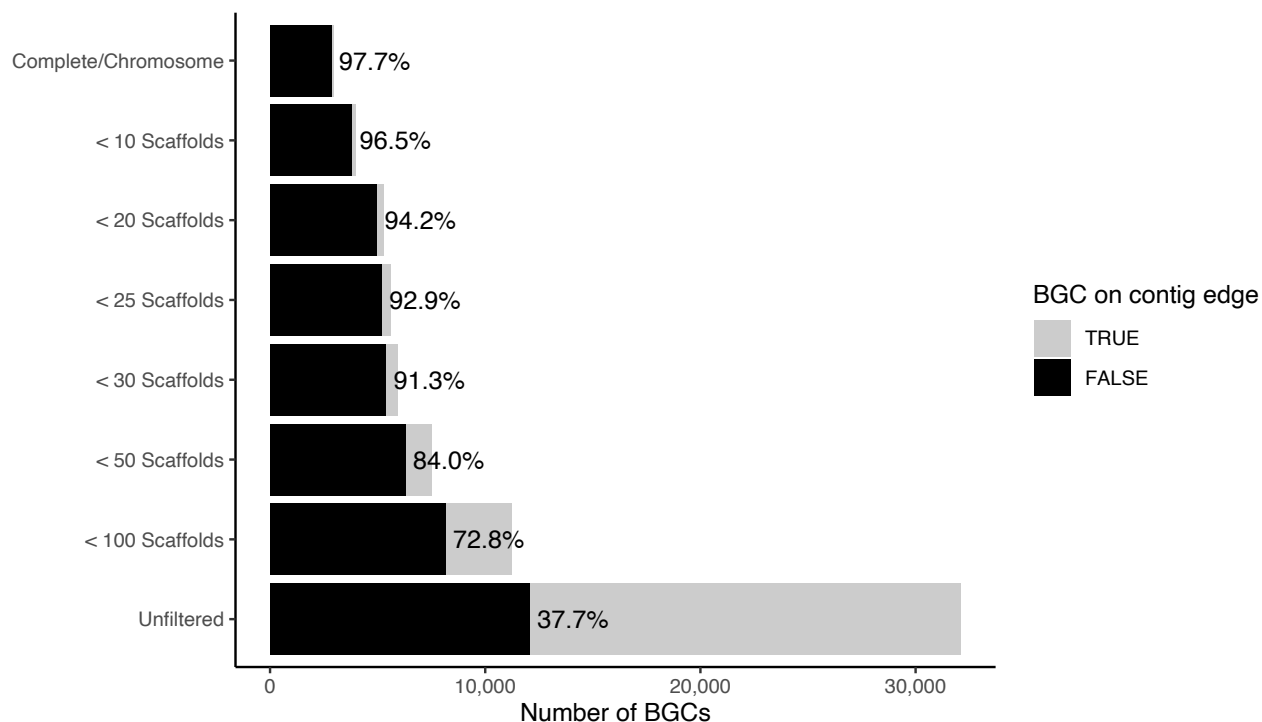
This figure depicts the number of BGCs against the number of scaffolds in a genome. To help avoid overplotting (i.e. many overlapping data points misrepresenting the distribution of the data), the colors of each spot in the figure correspond to how many data points overlap at those coordinates.

How does the proportion of BGCs off/on a contig edge change if we filter for high-quality genomes in different ways?

```
filters_df <- bind_rows(
  regions %>%
    group_by(contig_edge) %>%
    summarize(filter = "Unfiltered", n = n()) %>%
    ungroup() %>%
    mutate(pct = 100 * n / sum(n)),
  regions %>%
    semi_join(ncbi_hiq_meta, by = join_by(accession_id == `Assembly Accession`)) %>%
    group_by(contig_edge) %>%
    summarize(filter = "Complete/Chromosome", n = n()) %>%
    mutate(pct = 100 * n / sum(n)),
  regions %>%
    filter(n_scaffolds < 10) %>%
    group_by(contig_edge) %>%
    summarize(filter = "< 10 Scaffolds", n = n()) %>%
    mutate(pct = 100 * n / sum(n)),
  regions %>%
    filter(n_scaffolds < 20) %>%
    group_by(contig_edge) %>%
    summarize(filter = "< 20 Scaffolds", n = n()) %>%
    mutate(pct = 100 * n / sum(n)),
  regions %>%
    filter(n_scaffolds < 25) %>%
    group_by(contig_edge) %>%
    summarize(filter = "< 25 Scaffolds", n = n()) %>%
    mutate(pct = 100 * n / sum(n)),
  regions %>%
    filter(n_scaffolds < 30) %>%
    group_by(contig_edge) %>%
    summarize(filter = "< 30 Scaffolds", n = n()) %>%
    mutate(pct = 100 * n / sum(n)),
  regions %>%
    filter(n_scaffolds < 50) %>%
    group_by(contig_edge) %>%
    summarize(filter = "< 50 Scaffolds", n = n()) %>%
    mutate(pct = 100 * n / sum(n)),
  regions %>%
    filter(n_scaffolds < 100) %>%
    group_by(contig_edge) %>%
    summarize(filter = "< 100 Scaffolds", n = n()) %>%
    mutate(pct = 100 * n / sum(n)),
)

ggplot(filters_df, aes(x = filter, y = n)) +
  geom_col(aes(fill = fct_rev(as_factor(contig_edge))), position = position_stack()) +
  geom_text(aes(y = n, label = sprintf("%1.1f%%", pct)), data = filters_df %>% filter(contig_edge == FA
  scale_x_discrete(name = "", limits = c("Unfiltered", "< 100 Scaffolds", "< 50 Scaffolds", "< 30 Scaff
  scale_y_continuous(name = "Number of BGCs", labels = label_comma()) +
  scale_fill_manual(name = "BGC on contig edge", values = c("gray80", "black")) +
  theme_classic() +
```

```
coord_flip()
```



This figure depicts the counts of BGCs that are on a contig edge vs. those that are not, depending on how we define what a “high-quality genome” is. Complete/Chromosome refers to the genomes at the “Complete” or “Chromosome” assembly levels on NCBI.

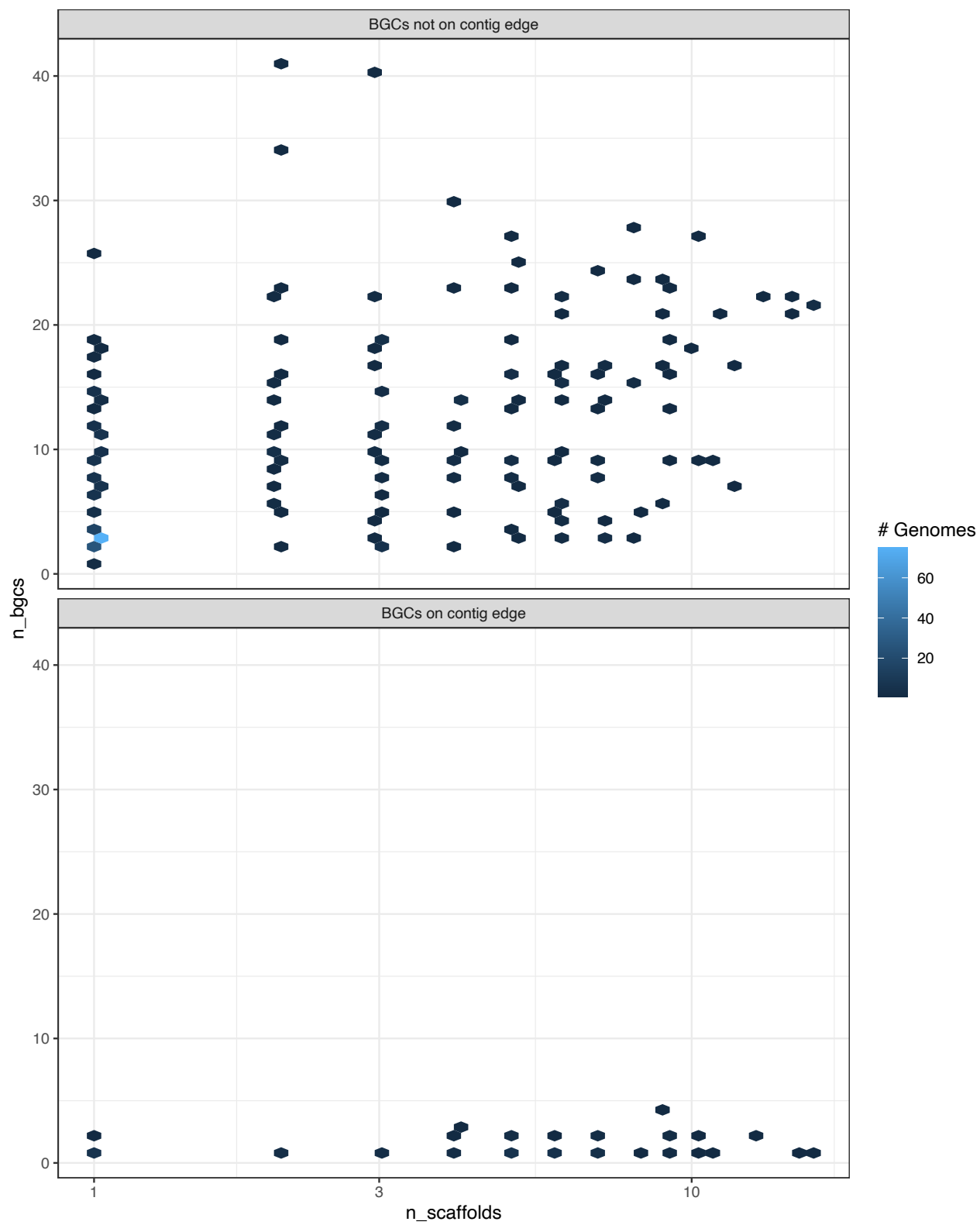
Based on this figure, and to be most conservative in this analysis, we will be going with the most restrictive criteria – using only genomes of “Chromosome” or “Complete” assembly quality as listed on NCBI.

## Filter the dataset to high-quality genomes

Repeat the figure from above, and we should see that most BGCs are not on a contig edge.

```
regions <- regions %>%
  semi_join(ncbi_hiq_meta, by = join_by(accession_id == `Assembly Accession`))

regions %>%
  group_by(smc_id, tax_genus, n_scaffolds, contig_edge) %>%
  summarize(n_bgcs = n()) %>%
  ggplot(aes(x = n_scaffolds, y = n_bgcs)) +
  stat_bin_hex(bins = 50) +
  facet_wrap(. ~ contig_edge, ncol = 1, labeller = as_labeller(c("FALSE" = "BGCs not on contig edge", "TRUE" = "BGCs on contig edge"))) +
  scale_x_log10(breaks = breaks_log()) +
  guides(fill = guide_colorbar(title = "# Genomes")) +
  theme_bw()
```



## Analyze data

Now we will proceed with our analysis, with the goal of looking at the BGC content of phylum Cyanobacteriota across the axes of length, BGC category, and taxonomy.

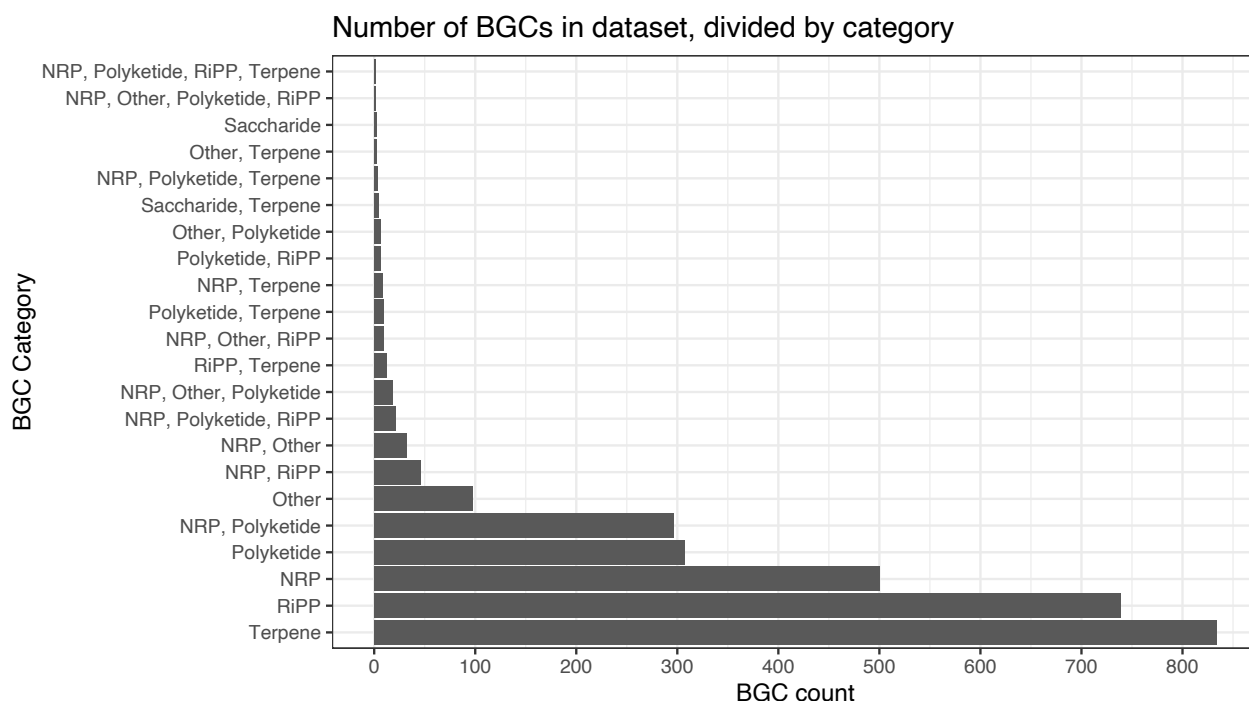
Note: AntiSMASH-annotated BGCs are assigned one or more of several dozen BGC “classes” based on the detection rule(s) triggered. These classes can also be grouped into one of 7 “categories” as defined by MIBiG – namely Polyketide, NRP, RiPP, Terpene, Saccharide, Alkaloid, and Other.

### Summary statistics of BGC length across BGC categories

```
## # A tibble: 22 x 7
## # Groups:   cats_str [22]
##   cats_str          n min_len max_len mean_len median_len      sd
##   <fct>          <int>   <dbl>   <dbl>    <dbl>    <dbl>   <dbl>
## 1 Terpene          834   13541   39454   20403.    20612.  2079.
## 2 RiPP             739    7553   76778   29623.    24133  10016.
## 3 NRP              500   20873   97395   47693.    43942. 10078.
## 4 Polyketide       307   22218   87362   46966.    46241   6996.
## 5 NRP, Polyketide  296   15761  190414   73530.    70899 24984.
## 6 Other            97   10034   60591   23411.    20762 10296.
## 7 NRP, RiPP        46   44214  131611   68485.    64902. 19801.
## 8 NRP, Other        32   43141   99576   76136.    78536. 20579.
## 9 NRP, Polyketide, RiPP 21   58186  257631  124101.    91425 63076.
## 10 NRP, Other, Polyketide 18   47242  154012   80890.    73720. 33500.
## # i 12 more rows
```

### How many BGCs in each category? (counting hybrids of categories as separate)

```
category_counts <- region_summary %>%
  ggplot(aes(y = reorder(cats_str, desc(n)))) +
  geom_col(aes(x = n)) +
  scale_x_continuous(name = "BGC count", breaks = breaks_width(100)) +
  scale_y_discrete(name = "BGC Category") +
  theme_bw() +
  ggtitle("Number of BGCs in dataset, divided by category")
category_counts
```



```
ggsave("./figs/svg/category_counts.svg", category_counts, device = "svg")
ggsave("./figs/png/category_counts.png", category_counts, device = "png")
```

### How many BGCs in each category? (lumping all hybrids into one except NRPS-PKS)

```
# Lump any hybrid category with fewer than 80 BGCs into an "all other" category
# - Threshold determined arbitrarily to improve visualization
region_summary_lumped <- region_summary %>%
  mutate(
    group = if_else(n < 80, "All other hybrids", cats_str),
    group = group %>% fct_reorder(n)
  )

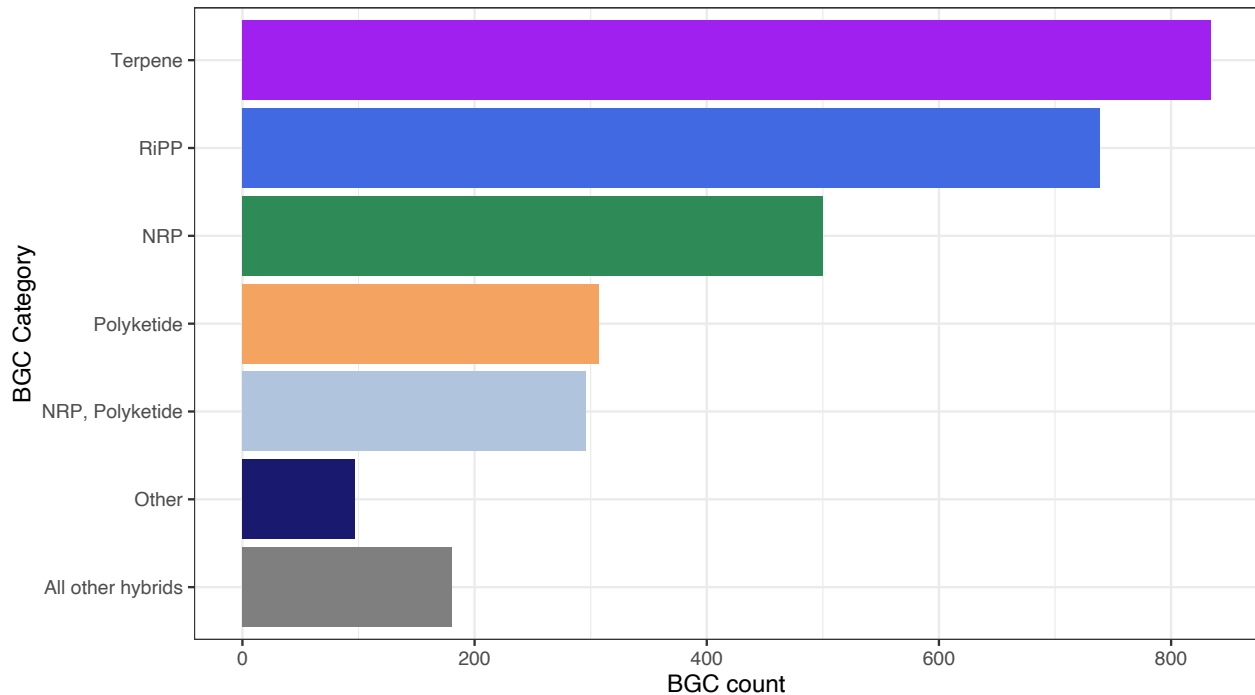
# Keep a reference DF handy for which categories got lumped
lump_groups <- region_summary_lumped %>% select(cats_str, group)

# Use the MIBiG / antiSMASH coloring scheme
cat_colors <- c(
  "Polyketide" = "#f4a460",
  "NRP" = "#2e8b57",
  "RiPP" = "#4169e1",
  "Terpene" = "purple",
  "Saccharide" = "#deb887",
  "Other" = "#191970",
  "NRP, Polyketide" = "lightsteelblue",
  "All other hybrids" = "gray50"
)

# Plot it
```



```
lumped_category_counts <- region_summary_lumped %>%
  ggplot(aes(y = reorder(group, n))) +
  geom_col(aes(x = n, fill = group)) +
  scale_x_continuous(name = "BGC count", breaks = breaks_extended()) +
  scale_y_discrete(name = "BGC Category") +
  scale_fill_manual(values = cat_colors) +
  theme_bw() +
  guides(fill = "none")
lumped_category_counts
```



```
ggsave("./figs/svg/category_counts_lumped.svg", lumped_category_counts, device = "svg")
ggsave("./figs/png/category_counts_lumped.png", lumped_category_counts, device = "png")
```

## How do BGCs vary in length by category (or combination of categories)?

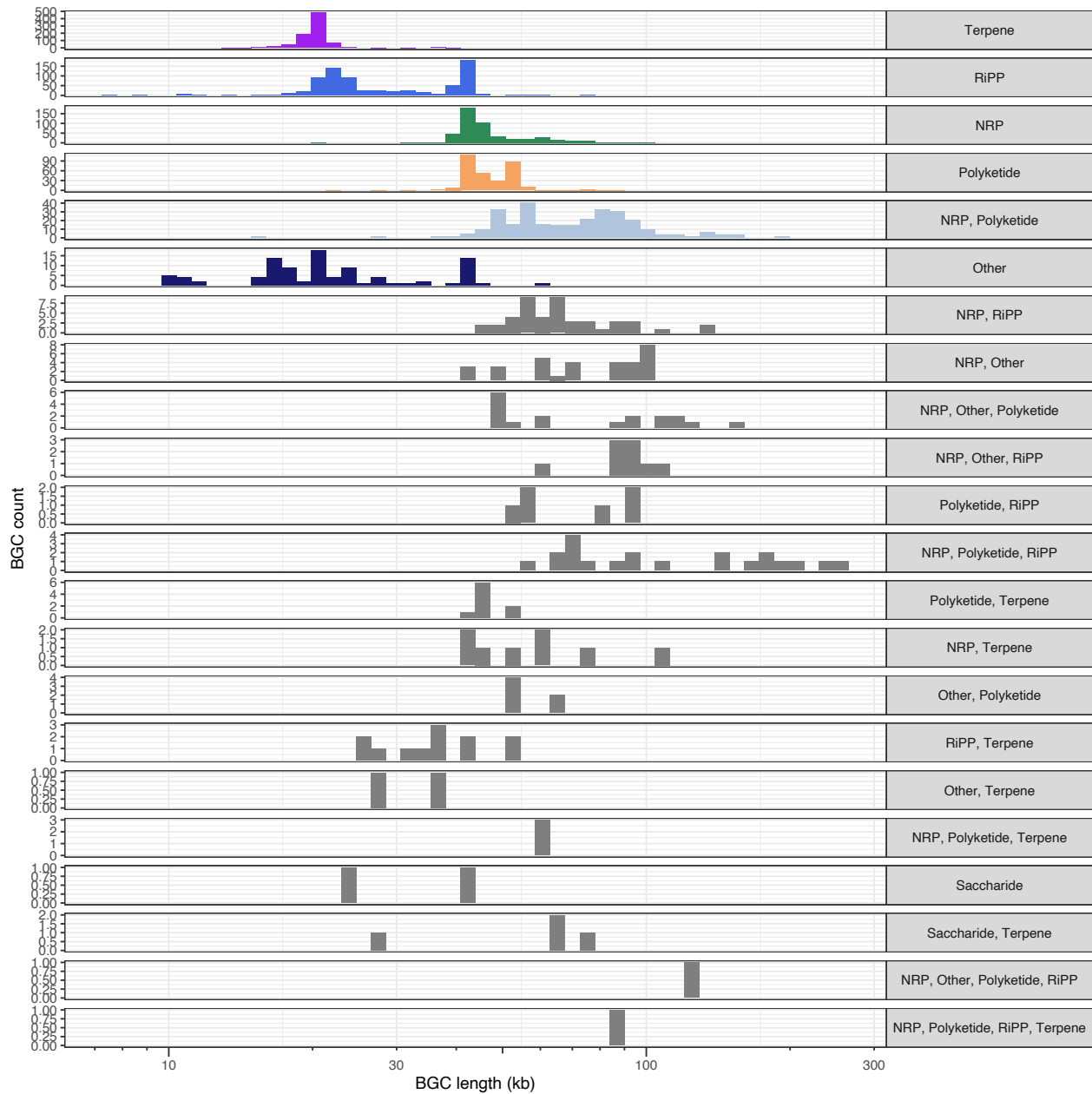
### Un-lumped categories

```
regions_lumped <- regions %>% left_join(lump_groups, by = "cats_str")

region_hist <- ggplot(regions_lumped, aes(
  x = region_length / 1000,
)) +
  geom_histogram(aes(fill = group), bins = 50) +
  scale_x_log10(name = "BGC length (kb)", guide = "axis_logticks", breaks = breaks_log(), labels = label_log()) +
  scale_y_continuous(name = "BGC count", breaks = breaks_extended(), labels = label_comma()) +
  scale_fill_manual(values = cat_colors) +
  facet_grid(rows = vars(cats_str), scales = "free_y") +
  theme_bw() +
  theme(strip.text.y.right = element_text(angle = 0)) +
```

```
guides(fill = FALSE)
```

```
region_hist
```



```
ggsave("../figs/svg/region_hist.svg", region_hist, device = "svg")
ggsave("../figs/png/region_hist.png", region_hist, device = "png")
```

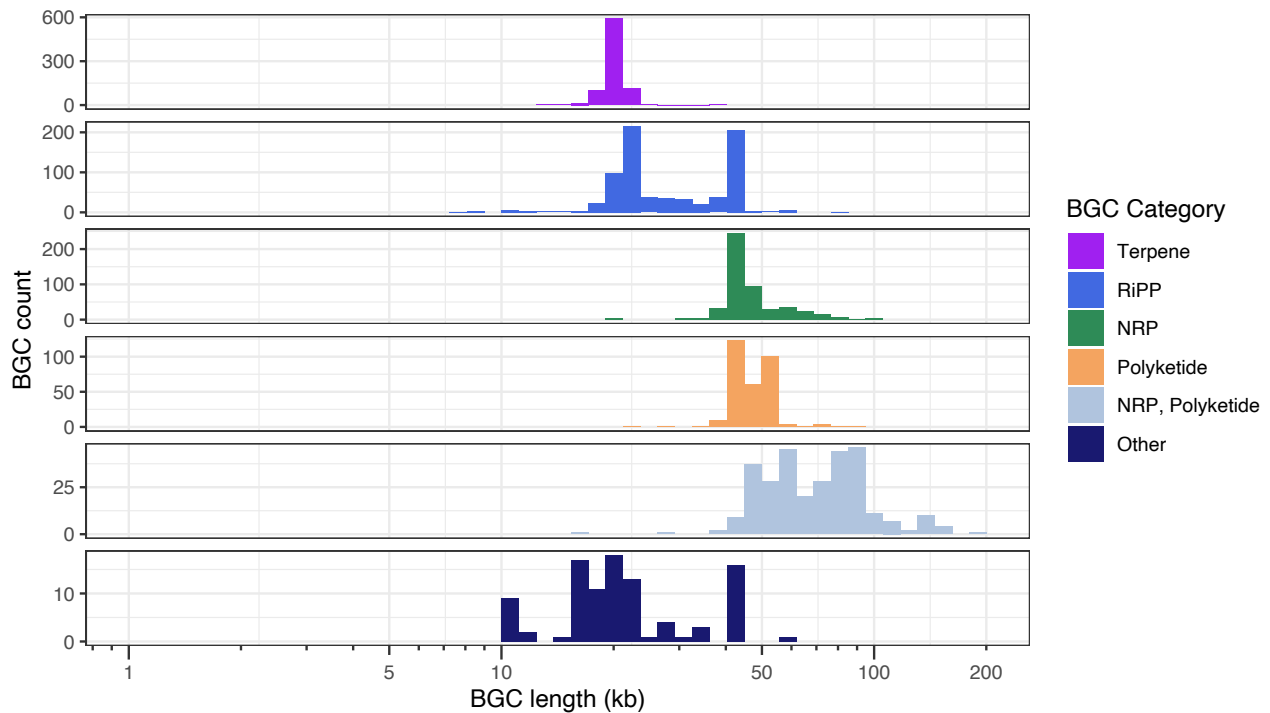
Lumped categories (again, except for NRPS-PKS hybrids)

```
region_hist_lumped <- regions_lumped %>%
  filter(group != "All other hybrids") %>%
  ggplot(aes(x = region_length / 1000)) +
```

```

geom_histogram(aes(fill = group), bins = 50) +
scale_x_log10(name = "BGC length (kb)", guide = "axis_logticks", limits = c(1, NA), breaks = c(1, 5, 10, 50, 100, 200)) +
scale_y_continuous(name = "BGC count", breaks = breaks_extended(n = 3)) +
scale_fill_manual(values = cat_colors) +
facet_wrap(vars(group), ncol = 1, scales = "free_y") +
guides(fill = guide_legend(title = "BGC Category")) +
theme_bw() +
theme(
  strip.background = element_blank(),
  strip.text = element_blank()
)
region_hist_lumped

```



```

ggsave("./figs/svg/region_hist_lumped.svg", region_hist_lumped, device = "svg")
ggsave("./figs/png/region_hist_lumped.png", region_hist_lumped, device = "png")

```

## How does BGC count vary across genera and by category?

Table:

```

tax_count <- regions_lumped %>%
  group_by(tax_genus, cats_str) %>%
  summarize(num_bgcs = n()) %>%
  mutate(tax_genus = tax_genus %>% fct_reorder(num_bgcs)) %>%
  left_join(lump_groups, by = "cats_str") # %>%

tax_count

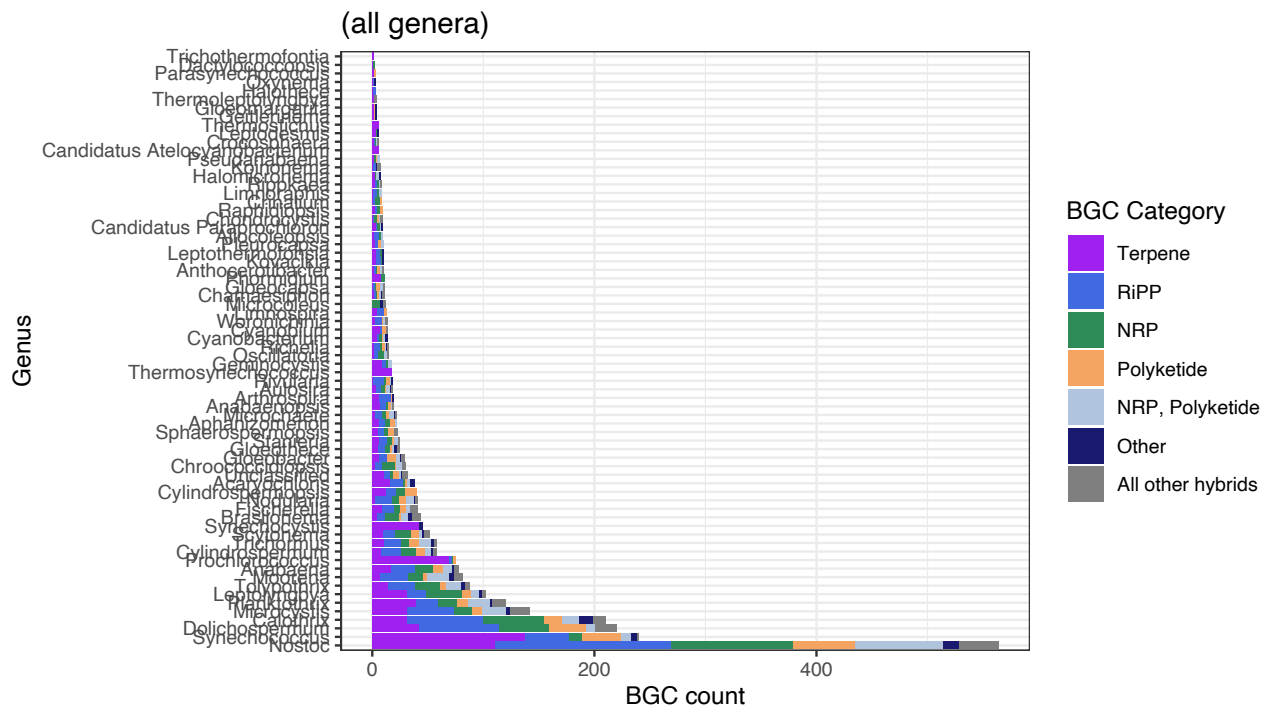
```

```
## # A tibble: 388 x 4
```

```
## # Groups:   tax_genus [70]
##   tax_genus   cats_str      num_bgcs group
##   <fct>      <fct>      <int> <fct>
## 1 Acaryochloris Terpene      16 Terpene
## 2 Acaryochloris RiPP        12 RiPP
## 3 Acaryochloris NRP         2 NRP
## 4 Acaryochloris Polyketide   2 Polyketide
## 5 Acaryochloris NRP, Polyketide 2 NRP, Polyketide
## 6 Acaryochloris Other        4 Other
## 7 Allocatedopsis Terpene      2 Terpene
## 8 Allocatedopsis RiPP        4 RiPP
## 9 Allocatedopsis NRP         2 NRP
## 10 Allocatedopsis NRP, Polyketide 1 NRP, Polyketide
## # i 378 more rows
```

Raw BGC counts by genus

```
p_all <- tax_count %>%
  group_by(tax_genus) %>%
  filter(sum(num_bgcs) > 0) %>%
  ggplot(aes(x = fct_infreq(tax_genus, w = num_bgcs))) +
  geom_col(aes(y = num_bgcs, fill = group), position = position_stack(reverse = TRUE)) +
  scale_y_continuous(name = "BGC count", breaks = breaks_extended()) +
  scale_x_discrete(name = "Genus") +
  scale_fill_manual(name = "BGC Category", values = cat_colors) +
  coord_flip() +
  theme_bw() +
  ggtitle("(all genera)")
p_all
```



```
ggsave("./figs/svg/genus_counts_all.svg", p_all, device = "svg")
ggsave("./figs/png/genus_counts_all.png", p_all, device = "png")
```

In order to normalize BGC counts to a per-genome basis, we must also know how many Cyano genomes *lacked* BGCs (as detected by antiSMASH).

```
# Plaintext file listing all the accessions that had no BGCs
cyano_nohits <- read_tsv("data/ncbi_cyano_nohit_accs.txt", col_names = c("accession_id")) %>%
  left_join(cyano_asm_tax, by = join_by(accession_id == assembly_accession))

# Incorporate these into our genome counts
cyano_nohit_genus_counts <- cyano_nohits %>%
  group_by(genus) %>%
  summarize(n_nohits = n()) %>%
  mutate(genus = replace_na(genus, "Unclassified")) %>%
  arrange(genus)

genomes_by_genus <- read_tsv("data/2025-02-25-1442-cyano_smc_src_counts_by_genus.tsv")
genomes_by_genus <- genomes_by_genus %>%
  left_join(cyano_nohit_genus_counts, by = join_by(tax_genus == genus)) %>%
  mutate(n_nohits = replace_na(n_nohits, 0)) %>%
  mutate(tot_genomes = n_nohits + n_sources, .keep = "unused")
```

Plot the BGCs per genome (normalized to 100%) alongside number of BGCs per genus and genomes per genus

```
# Prepare the dataframe specific to this set of plots
df_plots <- tax_count %>%
  filter(tax_genus != "Unclassified") %>%
  left_join(genomes_by_genus, by = "tax_genus") %>%
  mutate(bgc_dens = num_bgcs / tot_genomes) %>%
  group_by(tax_genus) %>%
  mutate(tot_bgc_dens = sum(bgc_dens), tot_bgcs = sum(num_bgcs))

# Plot BGCs per genome, colored by category and divided by genus
p_bgc_dens <- df_plots %>%
  ggplot(aes(x = fct_rev(tax_genus))) +
  geom_col(aes(y = bgc_dens, fill = group), position = position_fill(reverse = TRUE)) +
  scale_y_continuous(name = "BGC proportion") +
  scale_x_discrete(name = "Genus") +
  scale_fill_manual(name = "BGC Category", values = cat_colors) +
  coord_flip() +
  theme_bw() +
  theme(legend.position = "bottom")
# p_bgc_dens

# Plot total BGC count by genus
p_bgc_ct <- df_plots %>%
  ggplot(aes(x = fct_rev(tax_genus))) +
  geom_col(aes(y = tot_bgcs), data = df_plots %>% select(tax_genus, tot_bgcs, tot_bgc_dens) %>% distinct()) +
  scale_y_continuous(
    name = "BGC count",
    trans = transform_pseudo_log(base = 10),
```

```

    breaks = c(0, 1, 5, 10, 20, 50, 100, 200, 500)
  ) +
  coord_flip() +
  theme_bw() +
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_blank()
  )

# Plot genome count by genus
genome_counts <- df_plots %>%
  group_by(tax_genus, tot_genomes) %>%
  summarize(tot_bgc_dens = sum(bgc_dens), tot_bgcs = sum(num_bgcs)) %>%
  arrange(tot_bgc_dens)
genome_counts

```

```

## # A tibble: 69 x 4
## # Groups:   tax_genus [69]
##   tax_genus      tot_genomes tot_bgc_dens tot_bgcs
##   <chr>          <dbl>      <dbl>    <int>
## 1 Prochlorococcus    1023      0.0733     75
## 2 Pseudanabaena       75      0.0933      7
## 3 Cyanobium         131      0.107     14
## 4 Microcoleus        66      0.182     12
## 5 Crocosphaera       19      0.316      6
## 6 Phormidium         32      0.344     11
## 7 Synechococcus     579      0.415    240
## 8 Microcystis       329      0.432    142
## 9 Aphanizomenon      38      0.579     22
## 10 Fischerella       64      0.641     41
## # i 59 more rows

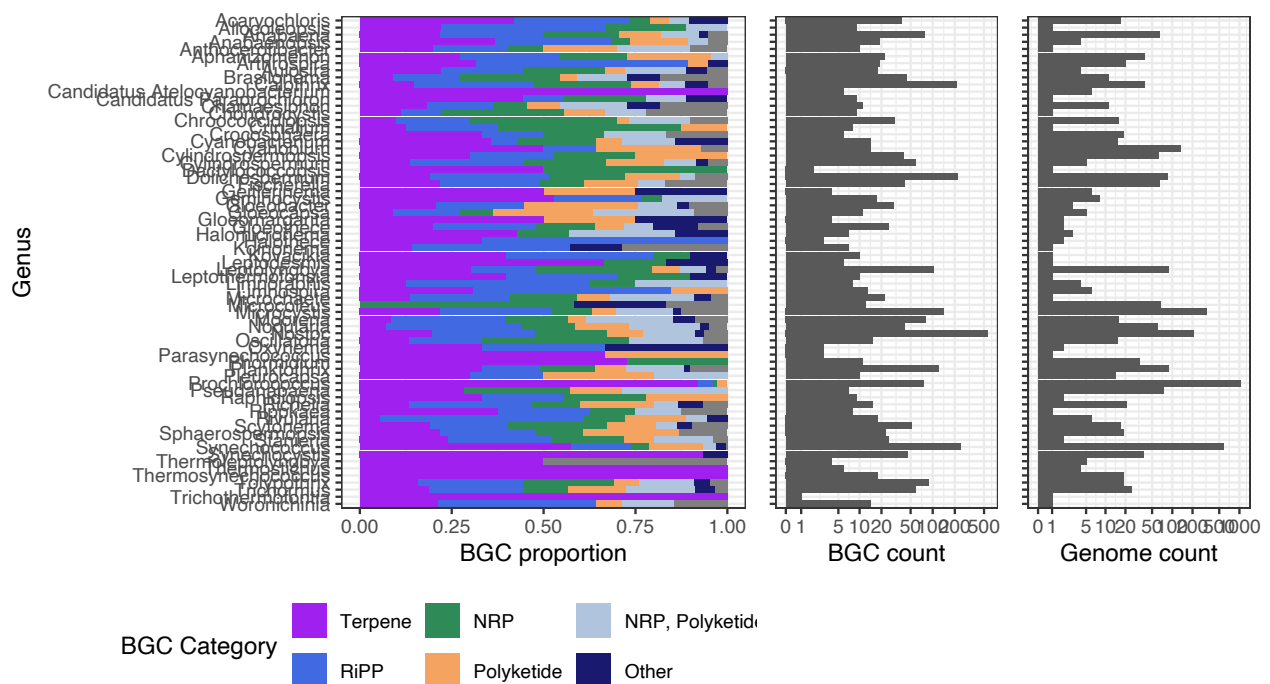
```

```

p_genome_ct <- genome_counts %>%
  ggplot(aes(x = fct_rev(tax_genus))) +
  geom_col(aes(y = tot_genomes)) +
  scale_y_continuous(
    name = "Genome count",
    trans = transform_pseudo_log(base = 10),
    breaks = c(0, 1, 5, 10, 20, 50, 100, 200, 500, 1000)
  ) +
  coord_flip() +
  theme_bw() +
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_blank()
  )
# p_genome_ct

# Plot them all together
p4 <- plot_grid(p_bgc_dens, p_bgc_ct, p_genome_ct, align = "h", rel_widths = c(3, 1, 1), nrow = 1)
p4

```



```
ggsave("./figs/png/split_proportional.png", p4, width = 8, height = 11, device = "png")
```

Plot the BGCs per genome (not normalized to 100%) alongside number of BGCs per genus and genomes per genus

```
# Prepare a dataframe specific to this set of plots
# df_plots <- tax_count %>%
#   filter(tax_genus != "Unclassified") %>%
#   left_join(genomes_by_genus, by = "tax_genus") %>%
#   mutate(bgc_dens = num_bgcs / tot_genomes) %>%
#   group_by(tax_genus) %>%
#   mutate(tot_bgc_dens = sum(bgc_dens), tot_bgcs = sum(num_bgcs))

# Plot BGCs per genome, colored by category and divided by genus
p_bgc_dens <- df_plots %>%
  ggplot(aes(x = fct_reorder(tax_genus, tot_bgc_dens, .desc = T))) +
  geom_col(aes(y = bgc_dens, fill = group), position = position_stack(reverse = TRUE)) +
  scale_y_continuous(name = "BGCs per genome") +
  scale_x_discrete(name = "Genus") +
  scale_fill_manual(name = "BGC Category", values = cat_colors) +
  coord_flip() +
  guides(fill = guide_legend(position = "inside")) +
  theme_bw() +
  theme(legend.justification.inside = c(0.99, 0.99))
# p_bgc_dens

# Plot total BGC count by genus
p_bgc_ct <- df_plots %>%
  ggplot(aes(x = fct_reorder(tax_genus, tot_bgc_dens, .desc = T))) +
  geom_col(aes(y = tot_bgcs), data = df_plots %>% select(tax_genus, tot_bgcs, tot_bgc_dens) %>% distinct()) +
  scale_y_continuous(
```

```

    name = "BGC count",
    trans = transform_pseudo_log(base = 10),
    breaks = c(0, 1, 5, 10, 100, 500)
  ) +
  coord_flip() +
  theme_bw() +
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_blank()
  )

# Plot genome count by genus
# genome_counts <- df_plots %>%
#   group_by(tax_genus, tot_genomes) %>%
#   summarize(tot_bgc_dens = sum(bgc_dens), tot_bgcs = sum(num_bgcs)) %>%
#   arrange(tot_bgc_dens)
# genome_counts

p_genome_ct <- genome_counts %>%
  ggplot(aes(x = fct_reorder(tax_genus, tot_bgc_dens, .desc = T))) +
  geom_col(aes(y = tot_genomes)) +
  scale_y_continuous(name = "Genome count", trans = scales::transform_pseudo_log(base = 10), breaks = c
  coord_flip() +
  theme_bw() +
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_blank()
  )
# p_genome_ct

# Plot them all together
p5 <- plot_grid(p_bgc_dens, p_bgc_ct, p_genome_ct, align = "h", rel_widths = c(3, 1, 1), nrow = 1)
df_plots

```

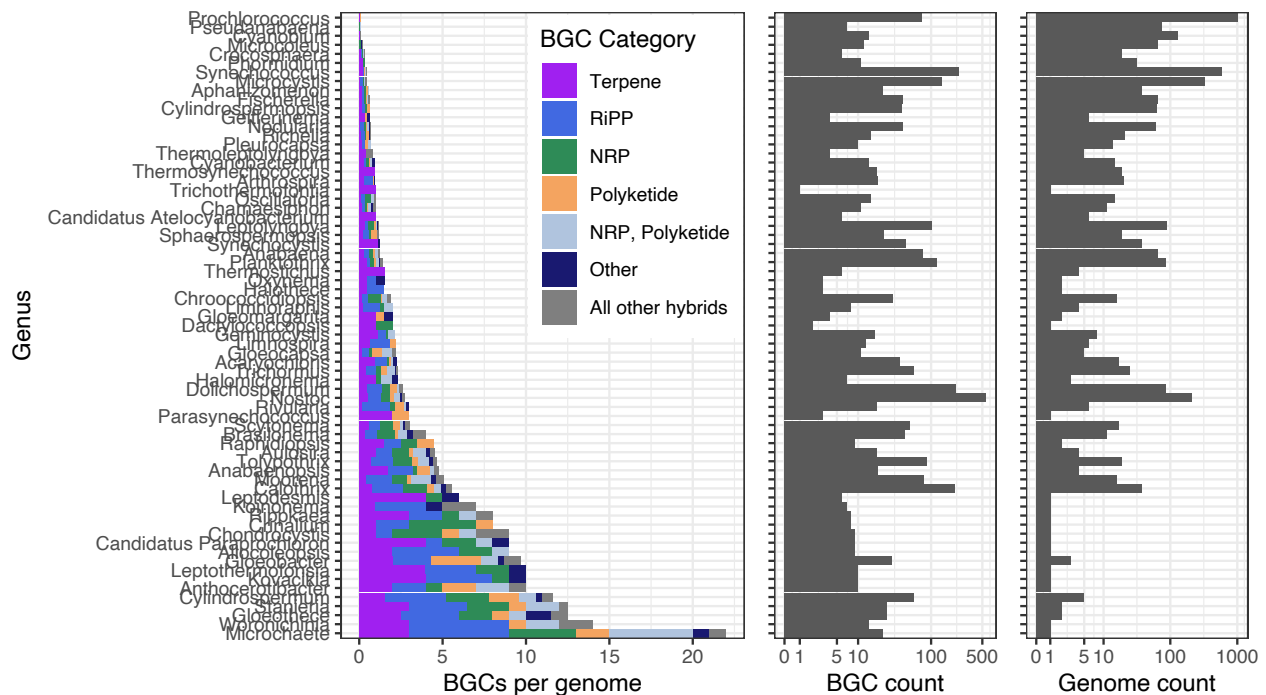
```

## # A tibble: 378 x 8
## # Groups:   tax_genus [69]
##   tax_genus cats_str num_bgcs group tot_genomes bgc_dens tot_bgc_dens tot_bgcs
##   <chr>      <fct>      <int> <fct>      <dbl>      <dbl>      <dbl>      <int>
## 1 Acaryochl~ Terpene      16 Terp~        17      0.941      2.24      38
## 2 Acaryochl~ RiPP        12 RiPP        17      0.706      2.24      38
## 3 Acaryochl~ NRP         2 NRP         17      0.118      2.24      38
## 4 Acaryochl~ Polyket~      2 Poly~        17      0.118      2.24      38
## 5 Acaryochl~ NRP, Po~      2 NRP,~        17      0.118      2.24      38
## 6 Acaryochl~ Other         4 Other        17      0.235      2.24      38
## 7 Allocoleo~ Terpene      2 Terp~         1      2          9          9
## 8 Allocoleo~ RiPP         4 RiPP         1      4          9          9
## 9 Allocoleo~ NRP         2 NRP         1      2          9          9
## 10 Allocoleo~ NRP, Po~      1 NRP,~         1      1          9          9
## # i 368 more rows

```

p5





```
ggsave("./figs/png/split_triple.png", p5, width = 8, height = 11, device = "png")
```

## Create Figure 2 for the manuscript

```
p_a <- plot_grid(
  p_bgc_dens + theme(axis.text.y = element_text(size = 8)),
  p_bgc_ct,
  p_genome_ct,
  align = "h",
  rel_widths = c(3, 1, 1),
  nrow = 1
)

p_b <- ggplot(
  region_summary_lumped,
  aes(y = reorder(group, n))
) +
  geom_col(aes(x = n, fill = group)) +
  scale_x_continuous(name = "BGC count", breaks = breaks_extended()) +
  scale_y_discrete(name = "BGC Category") +
  scale_fill_manual(values = cat_colors) +
  theme_bw() +
  guides(fill = "none")

p_c <- regions_lumped %>%
  filter(group != "All other hybrids") %>%
  ggplot(aes(x = region_length / 1000)) +
  geom_histogram(aes(fill = group), bins = 50) +
  scale_x_log10(name = "BGC length (kb)", guide = "axis_logticks", limits = c(1, NA), breaks = c(1,
```

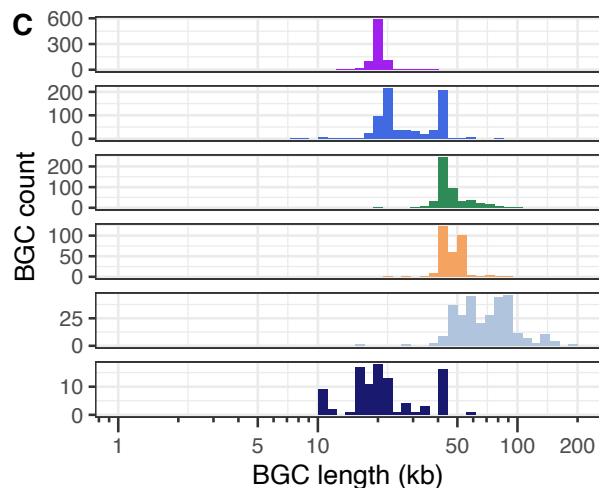
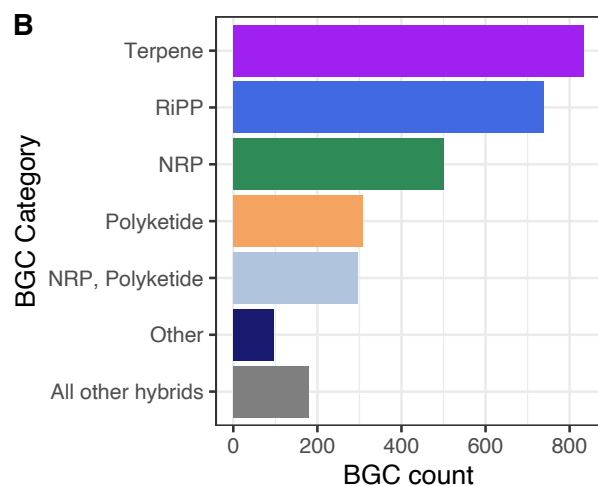
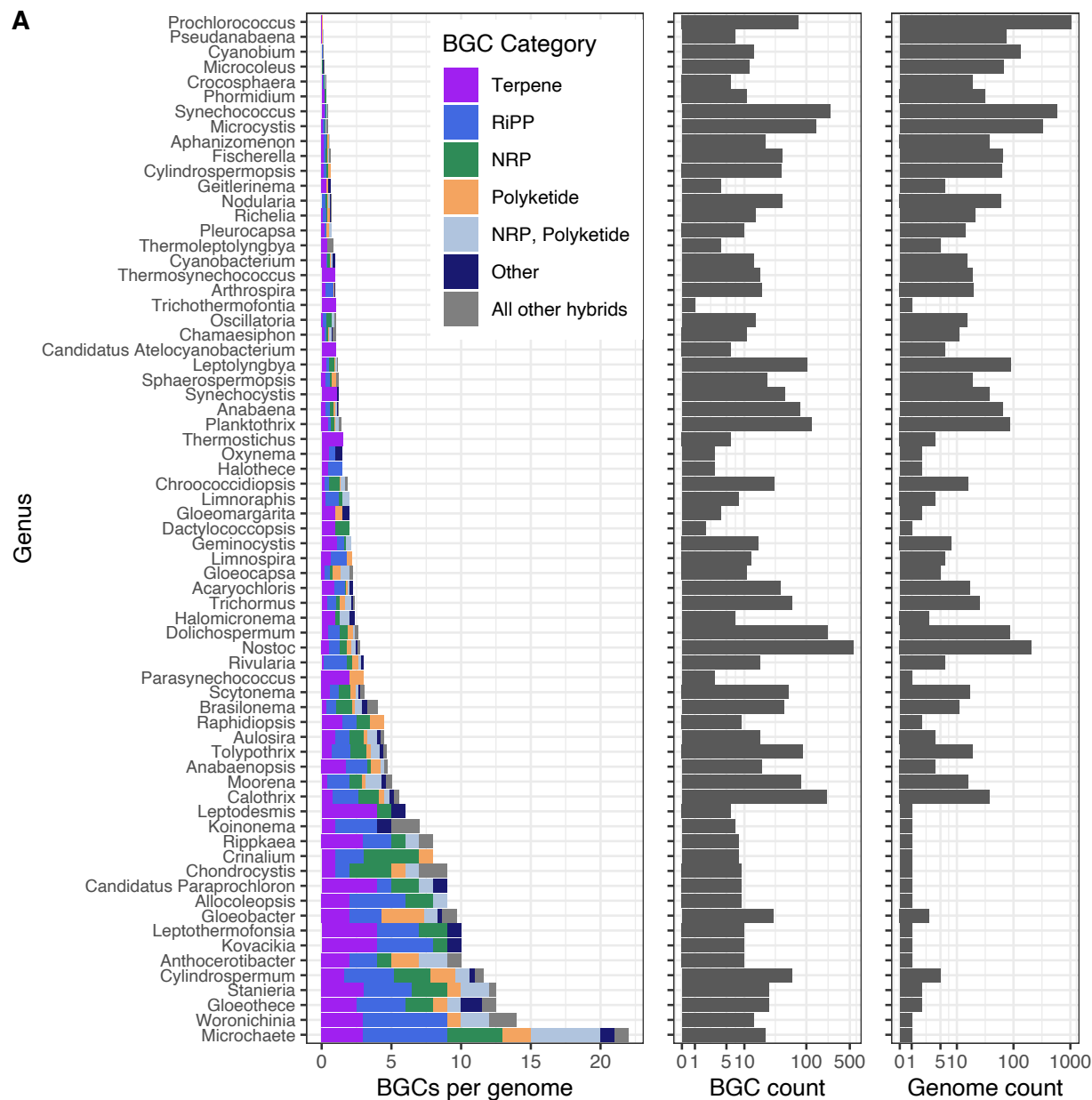
```

scale_y_continuous(name = "BGC count", breaks = breaks_extended(n = 3)) +
scale_fill_manual(values = cat_colors) +
facet_wrap(vars(group), ncol = 1, scales = "free_y") +
guides(fill = "none") +
theme_bw() +
theme(
  strip.background = element_blank(),
  strip.text = element_blank()
)

bottom_row <- plot_grid(p_b, p_c, nrow = 1, labels = c("B", "C"), label_size = 12)

fig2 <- plot_grid(p_a, bottom_row, nrow = 2, labels = c("A", ""), label_size = 12, rel_heights = c(2.5,
fig2

```



```
ggsave("./figs/_fig2.png", fig2, width = 7, height = 10, device = "png")  
ggsave("./figs/_fig2.pdf", fig2, width = 7, height = 10, device = "pdf")
```