Color palettes for visualizations:

```
palette_pretty <- c("#009E24", "#0072B2","#E69F00", "#FF0000",
                    "#979797", "#5530AA", "#1E1E1E")
palette_colorblind <- c("#E69F00", "#56B4E9", "#009E73","#F0E442",
                        "#0072B2", "#D55E00", "#CC79A7", "#999999")
palette_cb_ext <- c("#ebac23", "#b80058",  "#008cf9",
                    "#006e00", "#00bbad", "#d163e6", "#b24502",
                    "#ff9287", "#5954d6", "#00c6f8",
                    "#878500", "#00a76c", "#979797", "#1e1e1e")
palette_npg <- c("#E64B35", "#4DBBD5", "#00A087", "#3C5488",
                 "#F39B7F", "#8491B4", "#91D1C2", "#DC0000",
                 "#7E6148", "#B09C85")
theme_set(theme_bw(base_size = 12))
```

Functions:

```
# function for cleaning data
clean_go_cc <- function(df, exp){

  clean_df <- df %>%
    janitor::clean_names() %>%
    filter(qualifier == "located_in") %>%
    mutate(Taxon = exp) %>%
    select(gene_product_id:go_name, Taxon) %>%
    unique()

    return(clean_df)

}
```

Read in and format data:

```
human <- read_tsv("leca/localization_ml/data/quickgo/human_qgo_all.tsv")
```

```
## Rows: 223960 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (13): GENE PRODUCT DB, GENE PRODUCT ID, SYMBOL, QUALIFIER, GO TERM, GO N...
## dbl  (1): TAXON ID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
yeast <- read_tsv("leca/localization_ml/data/quickgo/yeast_qgo_all.tsv")
```

```
## Rows: 35905 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (13): GENE PRODUCT DB, GENE PRODUCT ID, SYMBOL, QUALIFIER, GO TERM, GO N...
## dbl  (1): TAXON ID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

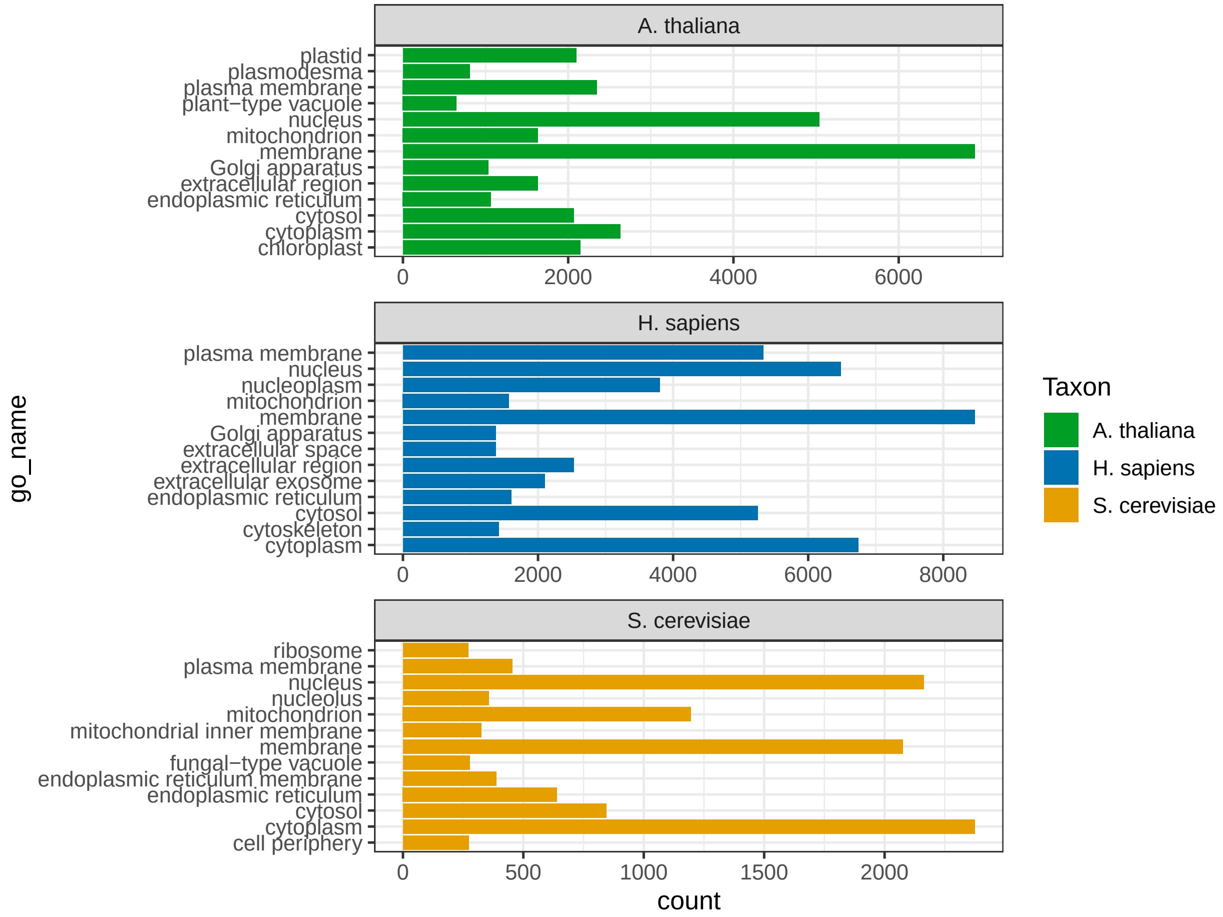
```
arath <- read_tsv("leca/localization_ml/data/quickgo/arath_qgo_all.tsv")
```

```
## Rows: 69779 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (12): GENE PRODUCT DB, GENE PRODUCT ID, SYMBOL, QUALIFIER, GO TERM, GO N...
## dbl  (1): TAXON ID
## lgl  (1): ANNOTATION EXTENSION
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
hclean <- clean_go_cc(human, "H. sapiens")
yclean <- clean_go_cc(yeast, "S. cerevisiae")
aclean <- clean_go_cc(arath, "A. thaliana")

all_data <- rbind(hclean, yclean, aclean)

summarized <- all_data %>%
  group_by(go_name, Taxon) %>%
  tally() %>%
  arrange(desc(Taxon), desc(n))

write_csv(summarized, "leca/localization_ml/results/summarized_all-quickgo_counts_xspecies.csv"
          )
```

Visualize the top hits from the raw data (by species):

```r
pdata <- summarized %>%
  ungroup %>%
  group_by(Taxon) %>%
  rename(count = n) %>%
  slice_max(count, n=13) %>%
  arrange(desc(go_name))

ggplot(pdata, aes(x = go_name, y = count, fill = Taxon)) +
  geom_col() +
  scale_fill_manual(values = palette_pretty) +
  facet_wrap(~Taxon, scales = "free", ncol = 1) +
  coord_flip()
```



Ed's suggested label list:

- GO:0005737 cytoplasm
- GO:0097708 intracellular vesicle
- GO:0009986 cell surface
- GO:0000785 chromatin
- GO:0005773 vacuole
- GO:0005929 cilium
- GO:0005856 cytoskeleton
- GO:0005634 nucleus
- GO:0016020 membrane
- GO:0005886 plasma membrane
- GO:0005739 mitochondrion
- GO:0031982 vesicle
- GO:0005794 Golgi apparatus
- GO:0005783 endoplasmic reticulum
- GO:0009986 cell surface

My final label list:

- GO:0042995 cell projection
- GO:0005856 cytoskeleton
- GO:0005829 cytosol
- GO:0031410 cytoplasmic vesicle
- GO:0005773 vacuole*

- GO:0005794 Golgi apparatus
- GO:0005783 endoplasmic reticulum
- GO:0005840 ribosome
- GO:0005634 nucleus
- GO:0005739 mitochondrion
- GO:0005886 plasma membrane
- GO:0005576 extracellular region

Not many vacuole labels in humans; also about half of the vacuoles in yeast & Arabidopsis are specifically labeled "fungal-type vacuole" or "plant-type vacuole." These labels are NOT child terms of vacuole, so these labels are lost with the final set listed above.

```
label_list <- c("cell projection", "cytoskeleton", "cytosol", "cytoplasmic vesicle", "vacuole", "Golgi a

pruned_labels <- all_data %>%
  filter(go_name %in% label_list)



# how many genes with labels did we retain?
uniq_genes <- unique(pull(all_data, gene_product_id)) # 42,835
uniq_genes_pruned <- unique(pull(pruned_labels, gene_product_id)) # 34,258
perc_retained_pruned <- (length(uniq_genes_pruned)/
                    length(uniq_genes))*100 # ~80%
```

Figure out which genes do not have labels in pruned list:

```
`%!in%` = Negate(`%in%`)

missing_genes <- all_data %>%
  filter(gene_product_id %!in% pruned_labels$gene_product_id)

missing_labels <- missing_genes %>%
  group_by(go_name) %>%
  tally() %>%
  mutate(percent = 100*(n/sum(n))) %>%
  arrange(desc(n))

# the vast majority of what we're missing falls into these 2 cases:
# 1. 'membrane' or 'cytoplasm' lacking a child term (~44%)
# 2. 'chloroplast' or 'plastid' (~16%)
```

Map pruned labels to KOG groups:

```
hogs <- read_tsv("leca/localization_ml/data/nog_mapping/human.euNOG.diamond.mapping.2759")
```

```
## Rows: 20504 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (2): ProteinID, ID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
yogs <- read_tsv("leca/localization_ml/data/nog_mapping/yeast.euNOG.diamond.mapping.2759")
```

```
## Rows: 5614 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (2): ProteinID, ID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

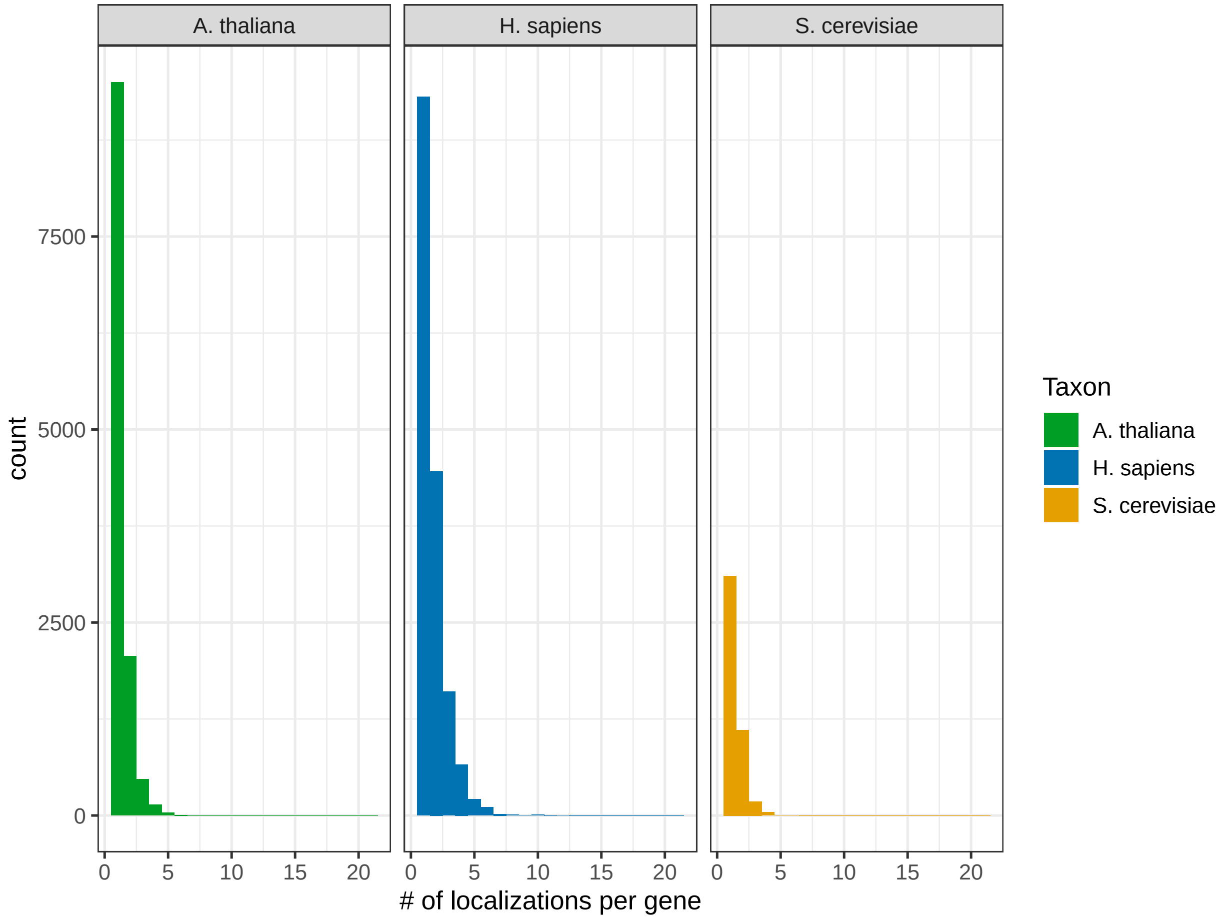
```
aogs <- read_tsv("leca/localization_ml/data/nog_mapping/arath.euNOG.diamond.mapping.2759")
```

```
## Rows: 25602 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (2): ProteinID, ID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
all_ogs <- bind_rows(hogs, yogs, aogs) %>%
  mutate(ProteinID = str_extract(ProteinID,'(?<=\\|)(.*)(?=\\|)')) # 51,720 OGs for human+yeast+arath

joined <- all_ogs %>%
  left_join(pruned_labels, by = c("ProteinID" = "gene_product_id")) %>%
  drop_na(go_term)

uniq_genes_joined <- unique(pull(joined, ProteinID)) # 33,062 genes
perc_retained_joined <- (length(uniq_genes_joined)/
                    length(uniq_genes_pruned))*100 # ~96.5% of these genes map to eggNOG groups

# created weighted KOG labels
weighted <- joined %>%
  group_by(ID, go_name) %>%
  tally() %>%
  rename(weight = n) %>%
  arrange(desc(weight))  # 24,093 orthogroups

# write out results:
write_csv(pruned_labels, "leca/localization_ml/results/pruned_quickgo_labels.csv")
write_csv(joined, "leca/localization_ml/results/pruned_quickgo-orthogroup_labels.csv")
write_csv(weighted, "leca/localization_ml/results/weighted_quickgo-orthogroup_labels.csv")
```

Evaluate the new labeling strategy:

```
# what is the distribution of the number of labels per gene?
joined %>%
  group_by(ProteinID, Taxon) %>%
  tally %>%
  ggplot(aes(x = n, fill = Taxon)) +
    geom_histogram(binwidth = 1) +
    facet_wrap(~Taxon, nrow = 1) +
    scale_fill_manual(values = palette_pretty) +
    labs(x = "# of localizations per gene")
```



```
# what is the distribution for the number of labels per orthogroup? (e.g. how does it shift from the pr

# make these same plots for the first approach (i.e. using the UniProt labels and the semi-manual regex

# which localizations have the most labels? (i.e. make a bar chart where each label is on the x-axis an

# which localizations are the most highly weighted?
```