SPOTlight / WesDe

Marc Elosua-Bayes

7/7/2021

```
library(SPOTlight)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

SPOTlight objects

Load data

```
spotlight1 <- readRDS(file = "~/Downloads/spotlight_ls.rds.gz")
spotlight2 <- readRDS(file = "~/Downloads/spotlight_ls_2.rds.gz")
```

Extract info

```
NMF1 <- spotlight1[[1]][[1]]
ct_labs1 <- spotlight1[[1]][[2]]
decon1 <- spotlight1[[2]]

NMF2 <- spotlight2[[1]][[1]]
ct_labs2 <- spotlight2[[1]][[2]]
decon2 <- spotlight2[[2]]
```

Explore these objects

```
colnames(decon1)
```

```
##  [1] "B.cells"                    "B.cells.memory"
##  [3] "B.cells.naive"              "Basophils"
##  [5] "Dendritic.cells"            "Distal.tubule.cells"
##  [7] "Gamma.delta.T.cells"        "Intercalated.cells"
##  [9] "Kidney.progenitor.cells"    "Loop.of.Henle.cells"
## [11] "Macrophages"                "Mast.cells"
## [13] "Megakaryocytes"             "Mesangial.cells"
```

```
## [15] "Monocytes"                    "Natural.killer.T.cells"
## [17] "Neutrophils"                  "NK.cells"
## [19] "Nuocytes"                     "Plasma.cells"
## [21] "Plasmacytoid.dendritic.cells" "Podocytes"
## [23] "Principal.cells"              "Proximal.tubule.cells"
## [25] "Red.pulp.macrophages"         "T.cells"
## [27] "T.follicular.helper.cells"    "T.helper.cells"
## [29] "T.regulatory.cells"           "Transient.cells"
## [31] "unknown"                      "res_ss"
```

```
colnames(decon2)
```

```
##  [1] "B.cells"                           "B.cells.memory"
##  [3] "B.cells.naive"                     "Basophils"
##  [5] "Collecting.duct.intercalated.cells" "Collecting.duct.principal.cells"
##  [7] "Collecting.duct.transient.cells"   "Cortical.cells"
##  [9] "Dendritic.cells"                   "Distal.convoluted.tubule.cells"
## [11] "Distal.tubule.cells"               "Endothelial.cells"
## [13] "Eosinophils"                       "Fibroblasts"
## [15] "Gamma.delta.T.cells"               "Kidney.progenitor.cells"
## [17] "Loop.of.Henle.cells"               "Macrophages"
## [19] "Mast.cells"                        "Medullary.cells"
## [21] "Megakaryocytes"                    "Mesangial.cells"
## [23] "Monocytes"                         "Myeloid.derived.suppressor.cells"
## [25] "Natural.killer.cells"              "Natural.killer.T.cells"
## [27] "Neutrophils"                       "NK.cells"
## [29] "Nuocytes"                          "Plasma.cells"
## [31] "Plasmacytoid.dendritic.cells"      "Podocytes"
## [33] "Principal.cells"                   "Proximal.tubule.cells"
## [35] "Red.pulp.macrophages"              "T.cells"
## [37] "T.cytotoxic.cells"                 "T.follicular.helper.cells"
## [39] "T.helper.cells"                    "T.memory.cells"
## [41] "T.regulatory.cells"                "Transient.cells"
## [43] "unknown"                           "Urothelial.cells"
## [45] "Vascular.endothelial.cells"        "res_ss"
```

Topic profiles Take a look at which genes are the most important for each topic and therefore get an insight into which genes are driving them.

Model 1 Make sure that all rows add up to 1

```
# -ncol(decon1) to remove res_ss
rowSums(decon1[, -ncol(decon1)])
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [260] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [297] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [334] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

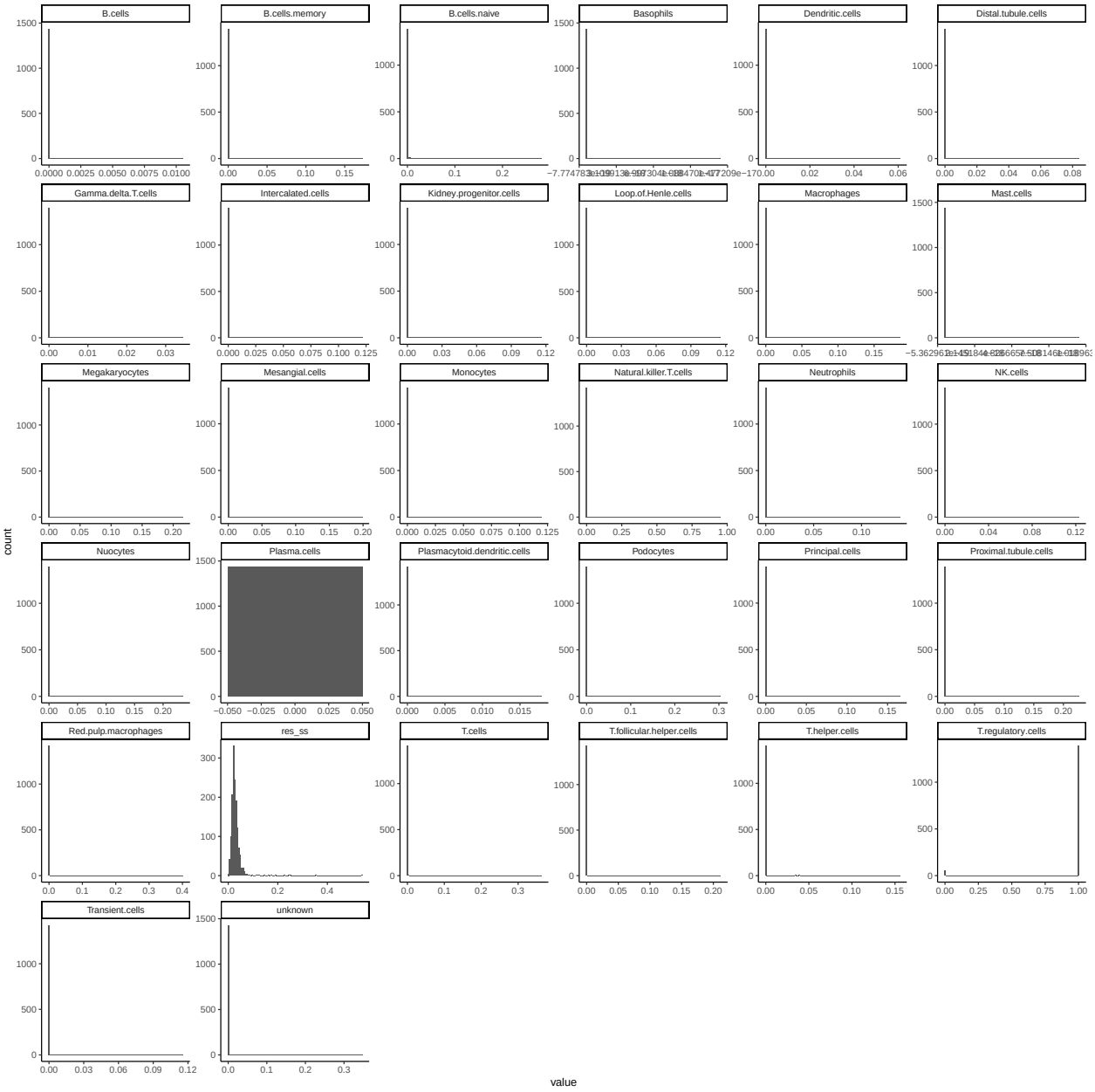
```
##  [371] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [408] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [445] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [482] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [519] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [556] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [593] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [630] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [667] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [704] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [741] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [778] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [815] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [852] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [889] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [926] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [963] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1000] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1037] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1074] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1111] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1148] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1185] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1222] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1259] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1296] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1333] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1370] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1407] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Look at summary statistics for each decomposed cell type

```
# Hmisc::describe(x = decon1)
```

Visualization of the predicted probabilities

```
data.frame(decon1) %>%
  tidyr::pivot_longer(cols = dplyr::everything()) %>%
  ggplot2::ggplot(ggplot2::aes(x = value)) +
  ggplot2::geom_histogram(bins = 100) +
  ggplot2::facet_wrap(. ~ name, scales = "free") +
  ggplot2::theme_classic()
```



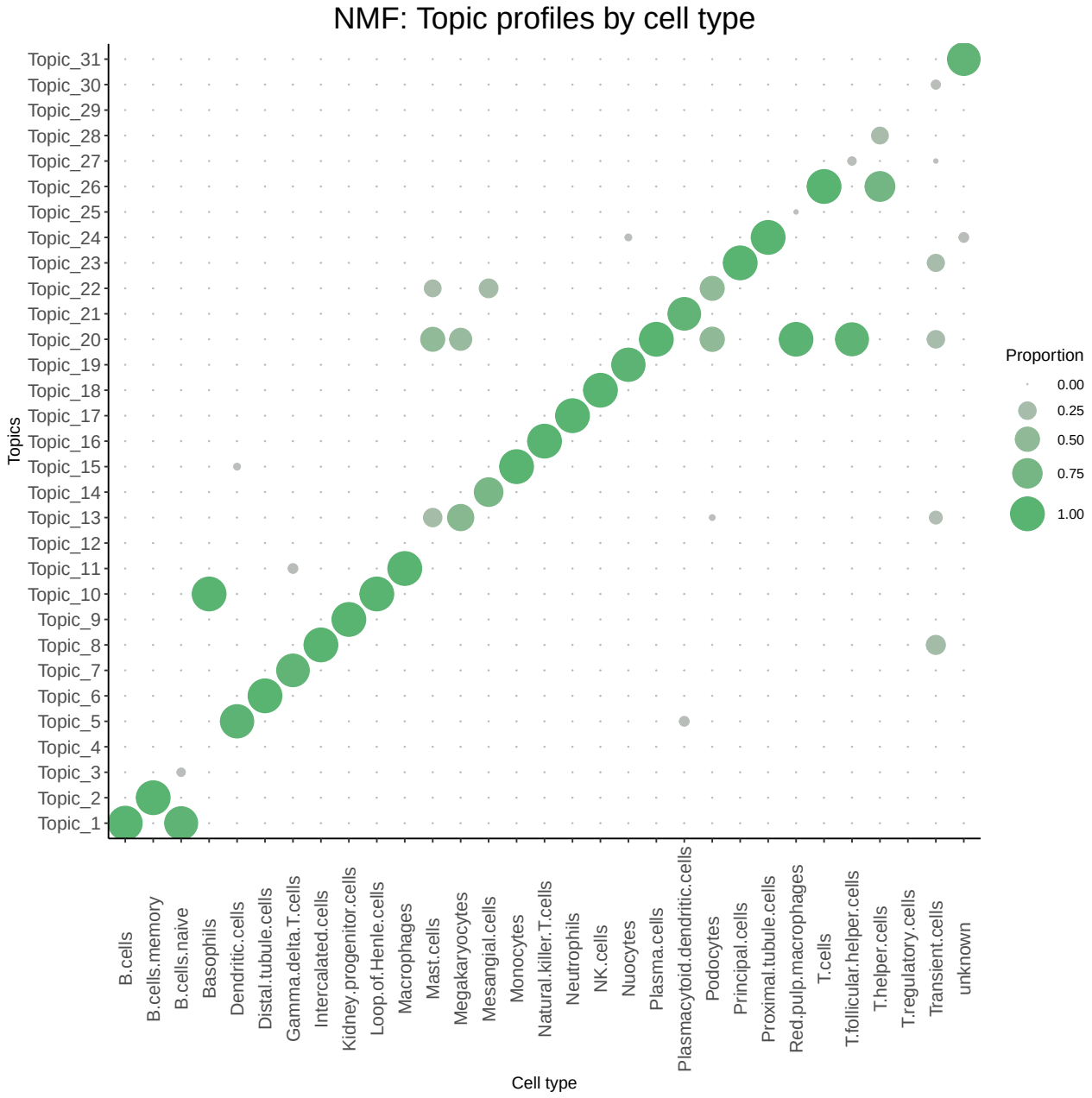
We see how regulatory T cells capture most of the signal. Lets look into this a bit further...

Look at the topic profiles:

```
h1 <- NMF::coef(NMF1)
rownames(h1) <- paste("Topic", 1:nrow(h1), sep = "_")
topic_profile_plts1 <- SPOTlight::dot_plot_profiles_fun(
  h = h1,
  train_cell_clust = ct_labs1)
```

Summarised topic profiles:

```
topic_profile_plts1[[2]] + ggplot2::theme(
  axis.text.x = ggplot2::element_text(angle = 90),
  axis.text = ggplot2::element_text(size = 12))
```



Above we can see how T reg doesn't have a representative topic profile, something that could be happening is that Tregs are defined by a topic profile with a low weight which would make it capture a lot of signal and overshadow other signal:

Lets take a look here

```
# Prep data
h_df <- data.frame(t(h1))
colnames(h_df) <- gsub(".", " ", colnames(h_df), fixed = TRUE)
h_ds <- h_df/rowSums(h_df)
h_ds[, "clust_vr"] <- ct_labs1

# Organize data
summarise_topic_df <- h_ds %>%
  dplyr::group_by(clust_vr) %>%
```

```
  dplyr::summarise_all(list(median)) %>%
  tibble::remove_rownames(.) %>%
  tibble::column_to_rownames("clust_vr")

summarise_topic_df[c("B.cells", "T.regulatory.cells"), ]
```

```
##                         Topic_1       Topic_2       Topic_3       Topic_4
## B.cells              1.000000e+00 1.473367e-132 1.061810e-104 2.979202e-140
## T.regulatory.cells 1.012583e-116 3.700013e-123 1.408013e-149 4.858607e-153
##                         Topic_5       Topic_6       Topic_7       Topic_8
## B.cells            3.264886e-128  0.000000e+00 6.417130e-152 6.401143e-283
## T.regulatory.cells 1.276929e-131 6.255418e-316 1.624871e-191 2.609962e-290
##                    Topic_9 Topic_10      Topic_11      Topic_12      Topic_13
## B.cells                  0        0 8.776509e-185 2.644161e-153 1.142498e-282
## T.regulatory.cells       0        0 1.402335e-155 1.004886e-153  0.000000e+00
##                        Topic_14      Topic_15      Topic_16      Topic_17
## B.cells            8.761477e-283 1.584300e-171 2.928760e-269 9.280821e-233
## T.regulatory.cells 6.580530e-287 3.821942e-147 3.586926e-276 8.619301e-119
##                        Topic_18      Topic_19      Topic_20      Topic_21
## B.cells            1.138082e-134 6.663898e-308 6.916919e-323 3.025777e-124
## T.regulatory.cells 7.403144e-175  0.000000e+00  0.000000e+00 5.590934e-127
##                        Topic_22 Topic_23 Topic_24      Topic_25      Topic_26
## B.cells            3.754541e-292        0        0 9.352009e-229 5.974090e-131
## T.regulatory.cells 1.236199e-314        0        0 2.009710e-276 1.236381e-176
##                        Topic_27      Topic_28      Topic_29      Topic_30
## B.cells            1.911883e-233 5.499163e-111 1.612499e-153 5.938242e-132
## T.regulatory.cells 1.721553e-270 2.430760e-135  1.842315e-87 9.133637e-168
##                        Topic_31
## B.cells            2.435566e-207
## T.regulatory.cells 1.366724e-193
```

We can see how B cells for example is dominated by topic_1 while T regs have no defined topic and it will try to fit noise until engulfing all of it.

Profiles for all cells

```
topic_profile_plts1[[1]] + theme(axis.text.x = element_text(angle = 90),
                                 axis.text = element_text(size = 12))
```

Regulatory T cells seem to have a large variablity, there is no one or 2 shared topics across all cells. Consider reclustering these cells or filter the gene set used to try to obtain a cleaner signal.

lastly, look at the genes for each topic

```
basis_spotlight1 <- data.frame(NMF::basis(NMF1))

colnames(basis_spotlight1) <- glue::glue("Topic-{1:ncol(basis_spotlight1)}")

basis_spotlight1 %>%
  round(., 5) %>%
  DT::datatable(., filter = "top")
```

```
## Warning in instance$preRenderHook(instance): It seems your data is too big
## for client-side DataTables. You may consider server-side processing: https://
```

rstudio.github.io/DT/server.html

Show 10 entries Search:

	Topic-1	Topic-2	Topic-3	Topic-4	Topic-5	Topic-6	Topic-7	Topic-8	Topic-9	Topic-10	Topic-11	Topic-12	Topic-13	Topic-14	Topic-15	Topic-16	Topic-17	Topic-18	Topic-19	Topic-20	Topic-21	Topic-22	Topic-23	Topic-24	Topic-25	Topic-26	Topic-27	Topic-28	Topic-29	Topic-30	Topic-31
Kap	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0013	0	0	0	0	0	0	0
Cd74	0.00069	0.00004	0	0	0.00178	0	0	0	0	0	0.00222	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ccl5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0026	0	0	0	0	0	0	0	0.00135	0	0	0	0	0
Lyz2	0	0	0	0	0	0	0	0	0	0	0.00055	0	0	0	0.00214	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Umod	0	0	0	0	0	0.00045	0	0	0	0.0042	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cxcl2	0	0	0	0	0	0	0	0	0	0	0.00119	0	0	0	0	0	0.00214	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Igkc	0	0.00234	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S100a8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00466	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Slc12a1	0	0	0	0	0	0	0	0	0	0.00382	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S100a9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.00441	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Showing 1 to 10 of 5,710 entries Previous 1 2 3 4 5 ... 571 Next

I do not see clear regulatory T cell markers in the basis matrix - *HAVCR2* and *FOXP3*.

Model 2 Make sure that all rows add up to 1

```
rowSums(decon2[, -ncol(decon2)])
```

```
##    [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##   [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##   [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [260] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [297] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [334] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [371] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [408] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [445] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [482] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [519] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [556] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [593] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [630] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [667] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [704] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [741] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [778] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [815] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [852] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [889] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [926] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [963] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1000] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1037] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1074] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1111] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1148] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1185] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1222] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1259] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1296] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1333] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1370] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1407] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Look at summary statistics for each decomposed cell type

```
# Hmisc::describe(x = decon2)
```

Visualization of the predicted probabilities

```
data.frame(decon2) %>%
  tidyr::pivot_longer(cols = dplyr::everything()) %>%
  ggplot2::ggplot(ggplot2::aes(x = value)) +
  ggplot2::geom_histogram(bins = 100) +
  ggplot2::facet_wrap(. ~ name, scales = "free") +
  ggplot2::theme_classic()
```

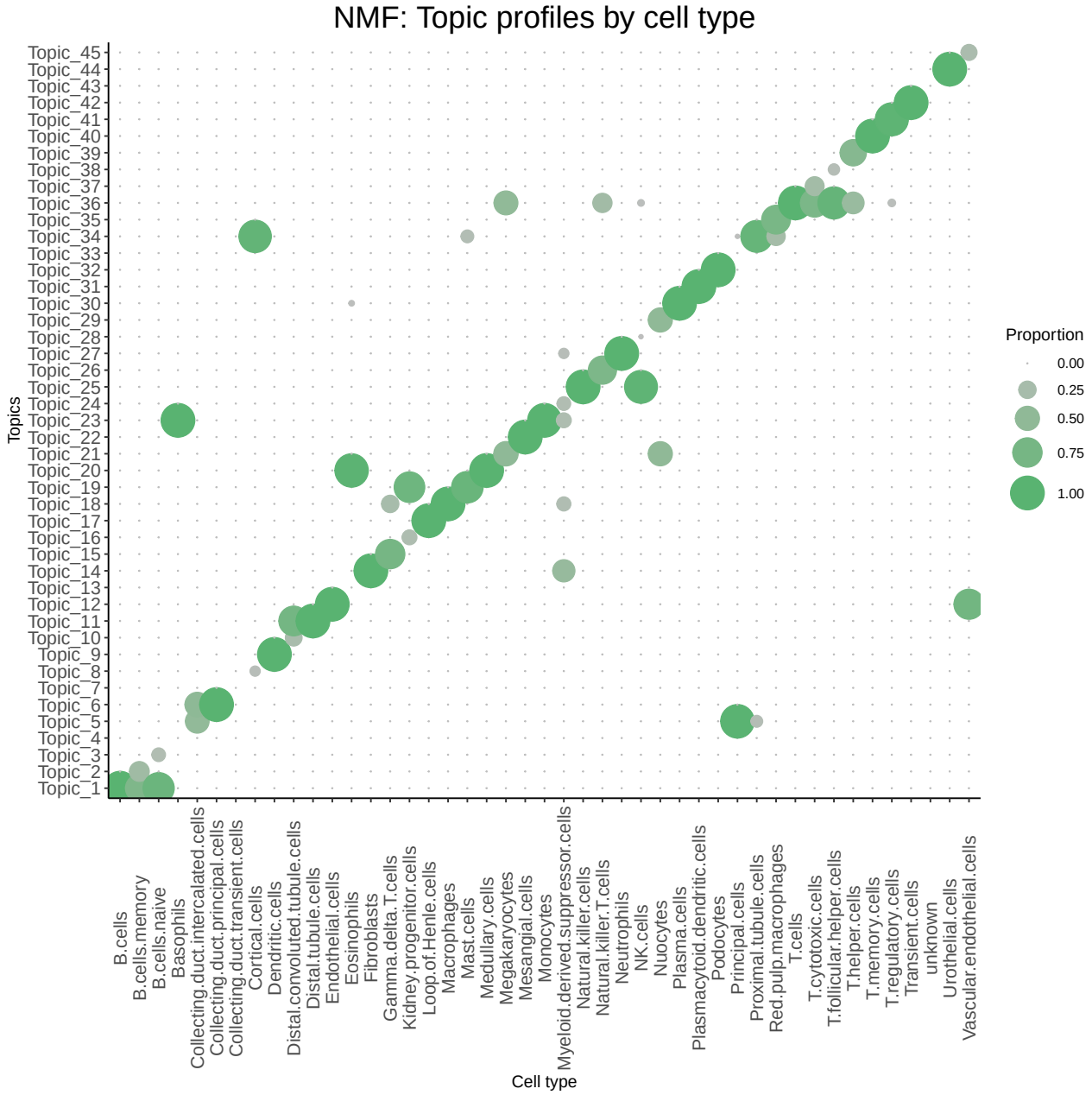
We can see how in this case the Unknown cell type captures all the biological signal.

Look at the topic profiles:

```
h2 <- NMF::coef(NMF2)
rownames(h2) <- paste("Topic", 1:nrow(h2), sep = "_")
topic_profile_plts2 <- SPOTlight::dot_plot_profiles_fun(
  h = h2,
  train_cell_clust = ct_labs2)
```

Summarised topic profiles:

```
topic_profile_plts2[[2]] +
  ggplot2::theme(
    axis.text.x = ggplot2::element_text(angle = 90),
    axis.text = ggplot2::element_text(size = 12))
```



We see the unkown cell type has the same issue as before with T regs... lets follow the same process as

before:

```
# Prep data
h_df <- data.frame(t(h2))
colnames(h_df) <- gsub(".", " ", colnames(h_df), fixed = TRUE)
h_ds <- h_df/rowSums(h_df)
h_ds[, "clust_vr"] <- ct_labs2

# Organize data
summarise_topic_df <- h_ds %>%
  dplyr::group_by(clust_vr) %>%
  dplyr::summarise_all(list(median)) %>%
  tibble::remove_rownames(.) %>%
  tibble::column_to_rownames("clust_vr")

summarise_topic_df[c("B.cells", "unknown"), ]
```

```
##               Topic_1       Topic_2       Topic_3       Topic_4       Topic_5
## B.cells  9.569699e-01  6.398193e-88  4.998908e-83 8.667823e-133 1.159149e-190
## unknown 1.059014e-252 8.687149e-168 9.143148e-168 6.816725e-290 1.199423e-227
##               Topic_6       Topic_7       Topic_8       Topic_9      Topic_10
## B.cells  0.000000e+00 3.836642e-208  1.802978e-98 3.952849e-129 2.048662e-126
## unknown 1.905537e-271 3.115399e-275 6.011495e-178 1.091470e-220 1.107063e-249
##              Topic_11 Topic_12      Topic_13      Topic_14      Topic_15
## B.cells 6.195972e-292        0 7.900458e-125 5.412527e-173 2.527358e-127
## unknown 1.482197e-322        0 2.343032e-239 1.702626e-300 3.127862e-247
##              Topic_16      Topic_17      Topic_18      Topic_19 Topic_20
## B.cells 6.175962e-122 2.478869e-258 1.790457e-195 8.073985e-271        0
## unknown 1.601762e-195 4.515188e-296 1.152906e-257 1.369043e-297        0
##              Topic_21      Topic_22      Topic_23      Topic_24      Topic_25
## B.cells 8.178214e-130 8.625639e-288 8.579625e-140  4.302269e-96 2.957696e-139
## unknown 5.836843e-258 6.990716e-304 2.092529e-281 2.915914e-191 2.151044e-305
##              Topic_26      Topic_27      Topic_28      Topic_29      Topic_30
## B.cells 1.935595e-102 1.685337e-234  2.224527e-95  3.292801e-84 6.459372e-124
## unknown 1.514614e-178 2.555155e-308 1.155338e-174 6.414014e-189 1.066195e-273
##              Topic_31      Topic_32      Topic_33      Topic_34      Topic_35
## B.cells 1.454772e-109 8.952524e-215 2.217329e-168 8.100327e-293 2.972661e-146
## unknown 1.364786e-183 6.618049e-280 1.320265e-237  0.000000e+00 4.387390e-216
##              Topic_36      Topic_37      Topic_38      Topic_39      Topic_40
## B.cells 2.163817e-125  1.220318e-90  6.694623e-93  7.322614e-96 2.863484e-111
## unknown 7.308671e-284 1.204820e-189 5.537398e-180 1.064330e-206 1.217565e-237
##              Topic_41      Topic_42      Topic_43      Topic_44      Topic_45
## B.cells 2.510611e-114 4.335872e-261 2.474174e-142 1.122988e-262 1.495680e-102
## unknown 5.816953e-247 2.044599e-282 1.222245e-145 8.162808e-306 1.526617e-200
```

Same as before, the B cells associated topic is dominated by topic_1 while unknowns have no defined topic and therefore it will try to fit noise to signal until engulfing all of it.

Profiles for all cells

```
topic_profile_plts2[[1]] +
  ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 90),
                 axis.text = ggplot2::element_text(size = 12))
```

Again, the cells within the Unkown cluster have very variable topic profiles. I can also see that you used a bunch of cluster with number of cells < 5. I would recommend having clusters with $>$ ~20 cells since that allows the model to correctly capture the biological signal, if there are clusters with low number of cells the biology learned will be specific to those ~5 cells. Due to the sparsity of scRNAseq data the model cannot fully learn the shared biology across cells.

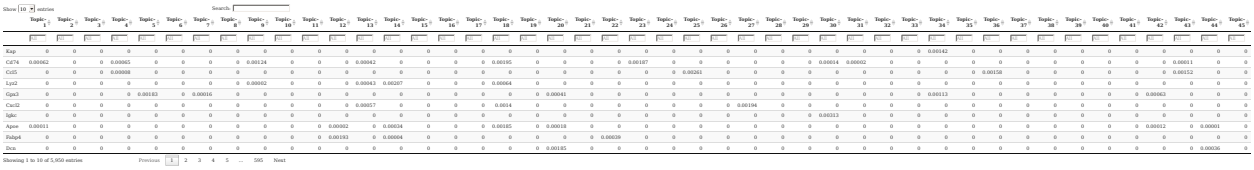
Lastly look at the weight of each gene for each topic.

```
basis_spotlight2 <- data.frame(NMF::basis(NMF2))

colnames(basis_spotlight2) <- glue::glue("Topic-{1:ncol(basis_spotlight2)}")

basis_spotlight2 %>%
  round(., 5) %>%
  DT::datatable(., filter = "top")
```

```
## Warning in instance$preRenderHook(instance): It seems your data is too big
## for client-side DataTables. You may consider server-side processing: https://
## rstudio.github.io/DT/server.html
```



Session Info

```
sessionInfo()
```

```
## R version 4.0.4 (2021-02-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.5 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=es_ES.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=es_ES.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=es_ES.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=es_ES.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] ggplot2_3.3.3       tidyr_1.1.3         tibble_3.1.2
## [4] stringr_1.4.0       Biobase_2.50.0      BiocGenerics_0.36.1
## [7] SPOTlight_0.1.6     dplyr_1.0.6
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.6           ps_1.6.0             assertthat_0.2.1
##  [4] digest_0.6.27        foreach_1.5.1        utf8_1.2.1
##  [7] gridBase_0.4-7       R6_2.5.0             plyr_1.8.6
## [10] evaluate_0.14        highr_0.9            pillar_1.6.1
## [13] rlang_0.4.11         callr_3.7.0          jquerylib_0.1.4
## [16] DT_0.18              rmarkdown_2.8        labeling_0.4.2
## [19] webshot_0.5.2        htmlwidgets_1.5.3    munsell_0.5.0
## [22] compiler_4.0.4       xfun_0.23            pkgconfig_2.0.3
## [25] pkgmaker_0.32.2.900 htmltools_0.5.1.1    tidyselect_1.1.1
## [28] codetools_0.2-18     fansi_0.4.2          crayon_1.4.1
## [31] withr_2.4.2          grid_4.0.4           jsonlite_1.7.2
## [34] xtable_1.8-4         gtable_0.3.0         lifecycle_1.0.0
## [37] registry_0.5-1       DBI_1.1.1            magrittr_2.0.1
```

```
## [40] scales_1.1.1        stringi_1.6.2       farver_2.1.0
## [43] reshape2_1.4.4      doParallel_1.0.16   bslib_0.2.5.1
## [46] ellipsis_0.3.2      generics_0.1.0      vctrs_0.3.8
## [49] RColorBrewer_1.1-2  NMF_0.23.0          iterators_1.0.13
## [52] tools_4.0.4         glue_1.4.2          purrr_0.3.4
## [55] crosstalk_1.1.1     rngtools_1.5        processx_3.5.2
## [58] yaml_2.2.1          colorspace_2.0-1    cluster_2.1.1
## [61] knitr_1.33          sass_0.4.0
```