Example workflow of fitting an MGPP model to tissue data

```r
library(tidyverse)
library(ppjsdm)
library(pals)
library(spatstat)
library(survival)
library(survminer)
library(kableExtra)

source("utils.R")
```

Welcome to the tutorial vignette for the paper "Investigating Ecological Interactions in the Tumor Microenvironment using Joint Species Distribution Models for Point Patterns"[1]. Here, I will be demonstrating how to run the analyses and create the figures that are demonstrated in the Example Analysis from that paper. To run all of the analyses in this vignette, you should install a fork of the fantastic `ppjsdm` package[2] from my Github using `devtools::install_github("jeliason/ppjsdm")`. My fork just adds one small piece of functionality to allow for calculation of the linear predictor of the conditional intensity function, as I show below.

DATA INPUT

Here, we load the data that is used in the paper, using `data_load_CRC()`. I am using a slightly cleaned-up and preprocessed version of the CODEX data from 35 colorectal cancer patients[3].

```r
set.seed(2023)

data_load_CRC()
spot <- "59_A"
```

```r
df <- df_raw %>%
  dplyr::filter(Spot == spot) %>%
  dplyr::mutate(type = forcats::fct_lump_min(type,20)) %>%
  dplyr::filter(type != "Other") %>%
  droplevels()
```

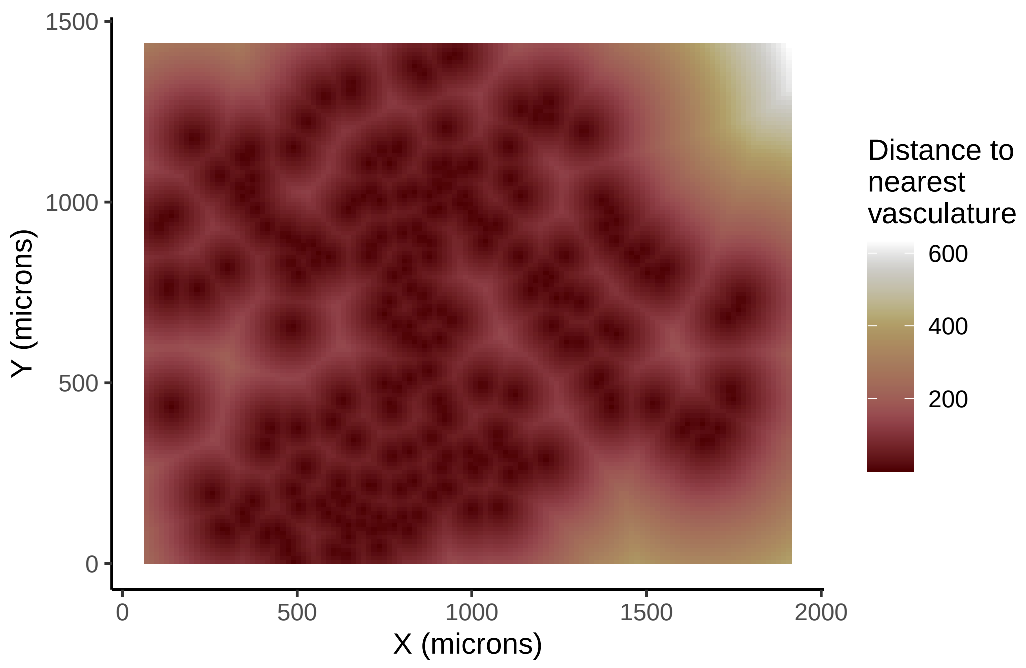
We will be demonstrating the MGPP method in this first part on spot 59_A. Here, we can see the abundance of each cell type from that spot:

```r
table(df$type)
```

```

      B cells CD163+ macros  CD8+ T cells  granulocytes memory CD4+ T
           71           637           200           599           173
 plasma cells        stroma         Tregs   tumor cells   vasculature
          307           367            34           176           235
```

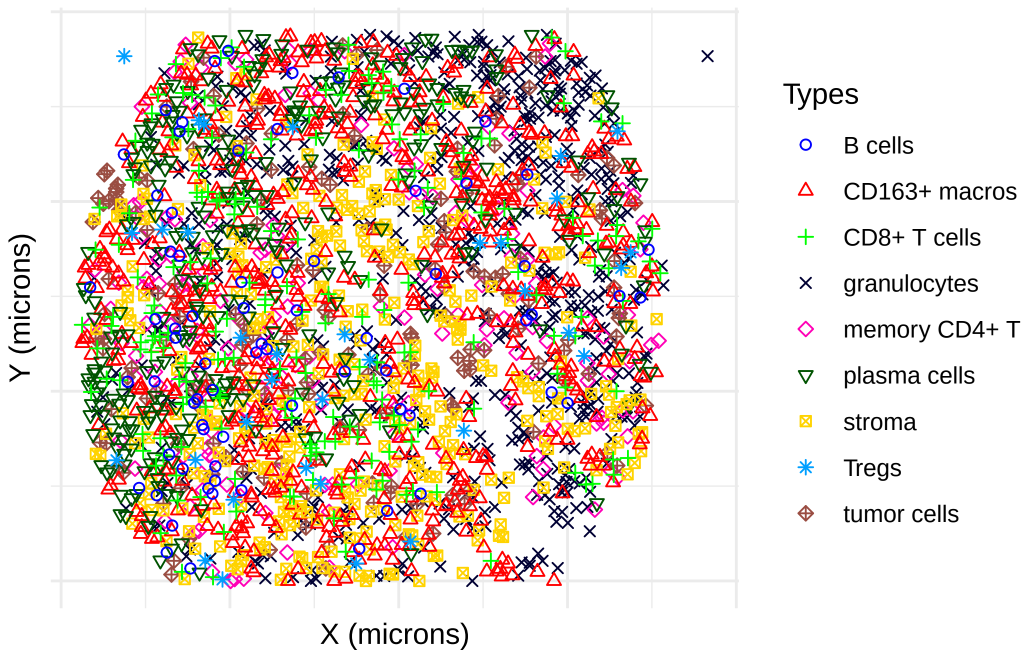
We can visualize the nearest-neighbor distance to vasculature cells at each point in space:

```r
nndist_vasculature <- nndist_cell_type(df,type = "vasculature")
nndist_vasculature %>%
  as.data.frame() %>%
  ggplot(aes(x,y,fill=value)) +
  geom_tile() +
  scico::scale_fill_scico(direction = 1) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.line = element_line(colour = "black")) +
  labs(x="X (microns)",y="Y (microns)",
       fill="Distance to \nnearest \nvasculature")
```



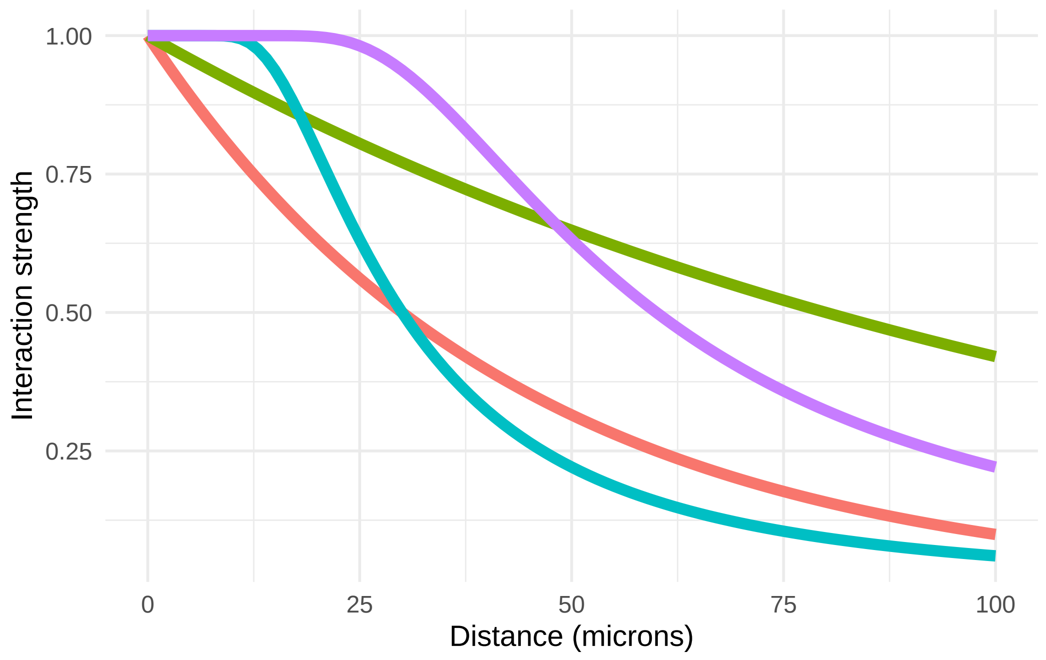
Additionally, we can plot the spatial distribution of each cell type as well:

```
region_var <- region_encoding(df)
x_range = c(min(df$X),max(df$X))
y_range = c(min(df$Y),max(df$Y))
df <- df %>%
  filter(type != "vasculature") %>%
  droplevels()
config <- Configuration(df$X,df$Y,df$type)
window <- Rectangle_window(x_range,y_range)

# point pattern plot

df %>%
  ggplot(aes(X,Y,color=type,shape=type)) +
  geom_point() +
  scale_color_manual(values = as.vector(glasbey(nlevels(df$type)))) +
  scale_shape_manual(values=1:nlevels(df$type)) +
  theme_minimal() +
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank()) +
  labs(x = "X (microns)",y = "Y (microns)",color="Types",shape="Types")
```



Finally, we can visualize the interaction potentials that we could use during model fitting to quantify the interaction strength between cell types:

```
r=seq(0,100,1)
tibble(r=r,exp1=exp(-r*log(2)/30),
       exp2=exp(-r*log(2)/80),
       sqb1=1-exp(-(30^2*log(2)/r^2)),
       sqb2=1-exp(-(60^2*log(2)/r^2))) %>%
  pivot_longer(-r) %>%
  ggplot(aes(r,value,color=name)) +
  geom_line(linewidth=2) +
  guides(color="none") +
  labs(x = "Distance (microns)",
       y = "Interaction strength") +
  theme_minimal()
```



MODEL FITTING

Now we turn to fitting the MGPP model that is detailed in our paper and the Flint et al. paper:

```r
R <- 30
saturation <- 5

number_of_species <- length(levels(df$type))
short_range <- matrix(R, number_of_species, number_of_species)
medium_range <- matrix(70, number_of_species, number_of_species)
long_range <- matrix(150, number_of_species, number_of_species)

covariates <- list()
covariates$outside = region_var$outside_var
covariates$vasc_dist <- nndist_vasculature
# covariates <- NULL
n_dummy <- 5e3

types <- levels(df$type)

args = list(configuration = config,
            window = window,
            covariates = covariates,
            model = "exponential",
            medium_range_model = "exponential",
            short_range = short_range,
            medium_range = medium_range,
            long_range = long_range,
            fitting_package = "glmnet",
            dummy_distribution = "binomial",
            saturation = saturation,
            min_dummy = n_dummy,
            max_dummy = n_dummy,
            debug = TRUE,
            nthreads = 2L)
tm <- Sys.time()
ppjsdm_fit <- do.call(gibbsm,args)
```

```
Starting computation of the regression matrix...
Starting pre-computation of the dispersions to put into the regression matrix...
Finished computing the dispersions. Time elapsed: 4 seconds, 556 milliseconds.
Finished computing the regression matrix. Time elapsed: 4 seconds, 584 milliseconds.

Calling glmnet...

Finished call to glmnet.

Time difference of 0.7099941 secs
```

```
print(Sys.time() - tm)
```

```
Time difference of 5.324512 secs
```

```
pots <- potentials(ppjsdm_fit)
```

as well as estimating the standard errors of each interaction estimate:

```
tm <- Sys.time()
ppjsdm_fit.summary <- summary(ppjsdm_fit, nthreads = 1, debug = FALSE)
print(Sys.time() - tm)
```

```
Time difference of 19.88283 secs
```

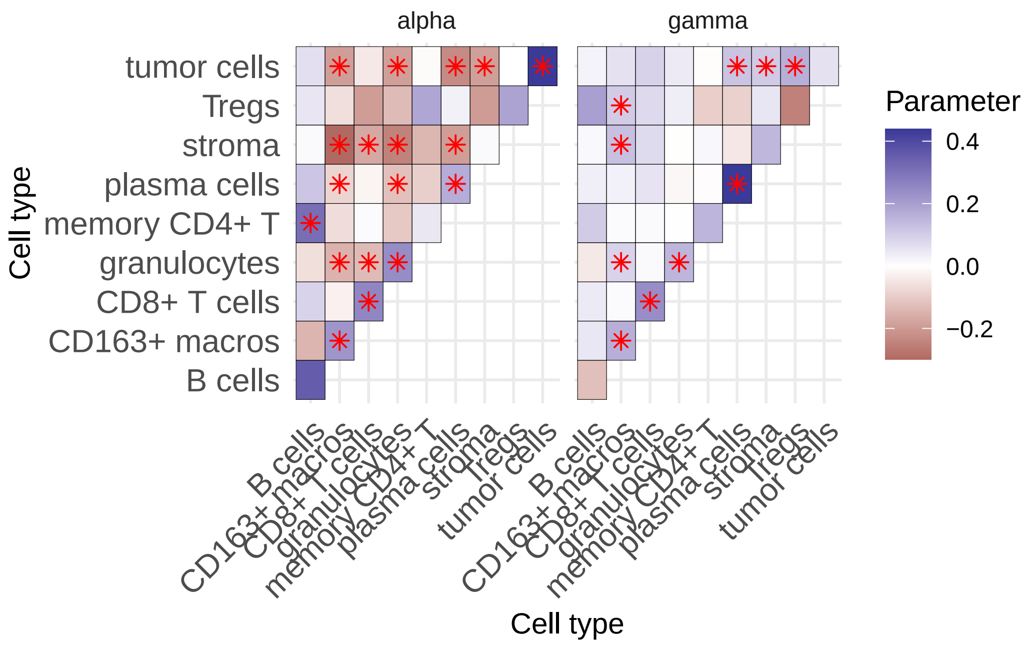
Here, we can see how much the intensity of each cell type is associated with the nearest-neighbor distance to vasculature:

```r
nms <- rownames(ppjsdm_fit.summary$se$alpha[[1]])

nms_tib <- tibble(type=nms,num_type=1:length(nms))
beta_coefs = c("vasc_dist")
d <- ppjsdm_fit.summary$coefficients %>%
  as_tibble(rownames = "coef_name") %>%
  filter(stringr::str_detect(coef_name,
                             stringr::str_c(beta_coefs,
                             collapse = "|"))) %>%
  separate(coef_name,into = c("coef","type1"),sep="_(?=[^_]+$)") %>%
  mutate(type1=as.numeric(type1)) %>%
  left_join(nms_tib,by=c("type1"="num_type")) %>%
  select(-type1)

d %>%
  select(type,coefficients,CI95_lo,CI95_hi) %>%
  mutate(estimate = formatC(coefficients,format = "e", digits=1)) %>%
  mutate(CI = paste0("(",formatC(CI95_lo,format = "e", digits =1),", ",
                     formatC(CI95_hi,format = "e",digits=1),")")) %>%
  select(-c(coefficients,CI95_lo,CI95_hi)) %>%
  kable(booktabs = T) %>%
  kable_paper(full_width=F)
```

type	estimate	CI
B cells	2.7e-05	(-6.1e-03, 6.1e-03)
CD163+ macros	-1.6e-03	(-3.1e-03, -8.0e-05)
CD8+ T cells	1.0e-03	(-1.8e-03, 3.9e-03)
granulocytes	-2.1e-03	(-3.3e-03, -8.0e-04)
memory CD4+ T	-2.0e-03	(-4.9e-03, 9.2e-04)
plasma cells	-1.3e-03	(-3.3e-03, 7.8e-04)
stroma	-6.4e-03	(-9.1e-03, -3.7e-03)
Tregs	4.4e-03	(-4.1e-03, 1.3e-02)
tumor cells	-2.3e-03	(-5.5e-03, 9.7e-04)

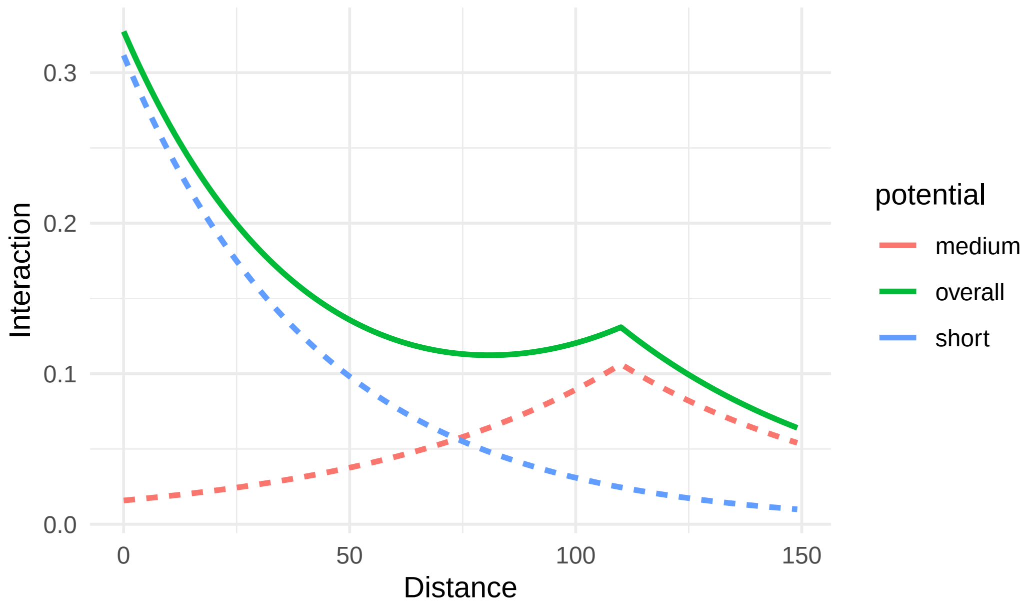
We can also visualize the interaction coefficients from short- and medium-range interactions as well. Here, short-range interactions are on the left, while medium-range are on the right. The fill color indicates the effect size, while the star indicates whether each estimated interaction is estimated to be statistically significant.

```r
a_df <- plot_coef(ppjsdm_fit.summary,coef="alpha",return_df = T)$df
g_df <- plot_coef(ppjsdm_fit.summary,coef="gamma",return_df = T)$df

bind_rows(mutate(a_df,name="alpha"),mutate(g_df,name="gamma")) %>%
  ggplot(aes(type1,type2,fill=coefficients)) +
  geom_tile(color="black") +
  anglex() +
  sig_stars(p_values="Pval") +
  scale_fill_gradient2() +
  labs(x = "Cell type",
       y = "Cell type",
       fill = "Parameter") +
  theme_minimal() +
  anglex() +
  theme(axis.text = element_text(size=12)) +
  facet_wrap(~name)
```



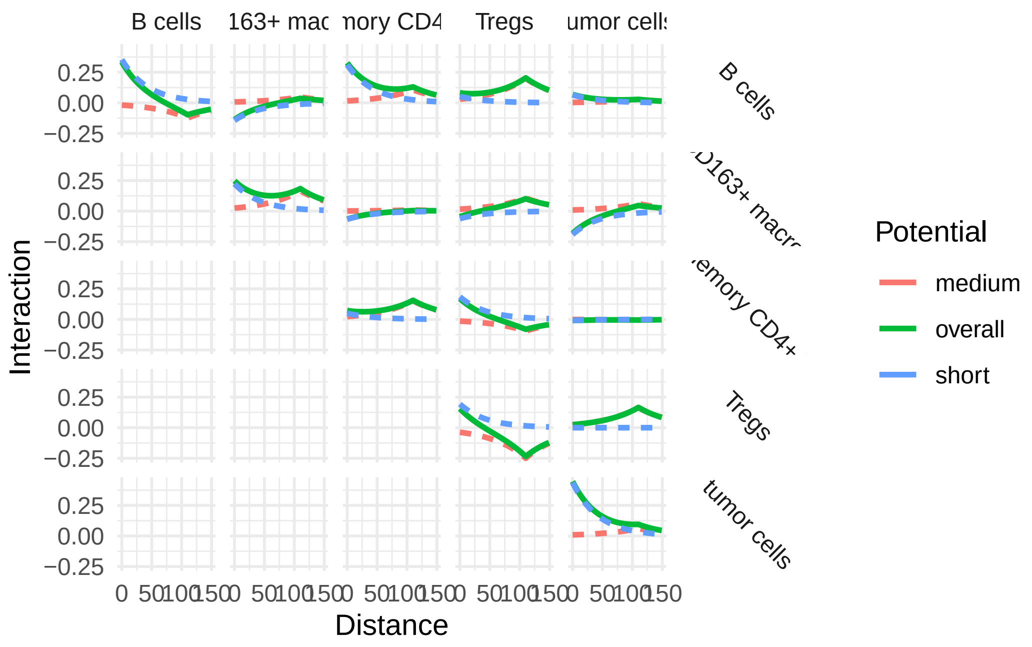
Multiplying each of these coefficients by the interaction potential that was used (here, the exponential for both short- and medium-range interactions) yields the actual interaction over spatial scales that exists between each pair of cell types. We can visualize these one at a time:

```r
plot_potentials(ppjsdm_fit,type1 = "B cells",type2 = "memory CD4+ T") +
  labs(title = "", x = "Distance",y = "Interaction") +
  theme_minimal()
```



Or in a pairwise fashion between a subset of cell types:

```r
plot_all_potentials(ppjsdm_fit,types = c("B cells","CD163+ macros",
                                          "memory CD4+ T","Tregs",
                                          "tumor cells")) +
  theme_minimal() +
  labs(x = "Distance",y = "Interaction",color="Potential") +
  theme(strip.text.y.right = element_text(angle=-45))
```



We can also recover the linear predictor of the conditional intensity for a given cell type. Here, we recover the linear predictor for tumor cells, and then plot, at each location in space, which of the cell types contributes the most to predicting the presence of tumor cells at that location.

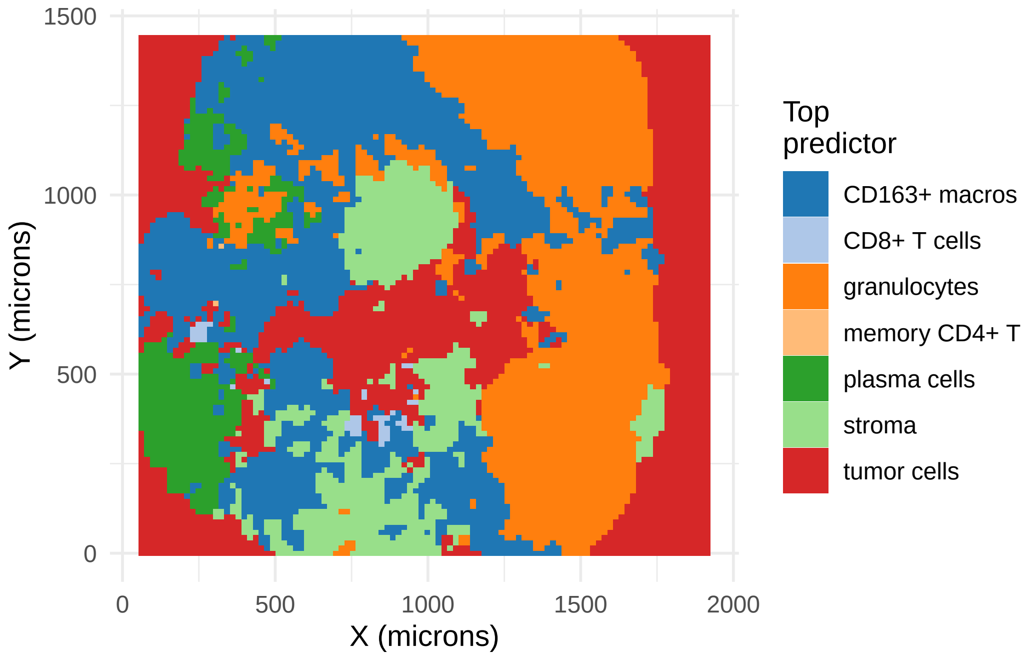
```
fit <- ppjsdm_fit
x <- seq(fit$window$x_range[1],fit$window$x_range[2],length.out=100)
y <- seq(fit$window$y_range[1],fit$window$y_range[2],length.out=100)

coords <- as.data.frame(expand.grid(x=x,y=y))
lin_pred <- ppjsdm:::model_matrix_papangelou.default(
   configuration = fit$configuration_list[[1]],
   x = coords$x,
   y = coords$y,
   type = "tumor cells",
   mark = 1.0,
   model = fit$parameters$model,
   medium_range_model = fit$parameters$medium_range_model,
   alpha = fit$coefficients$alpha,
   beta0 = fit$coefficients$beta0,
   beta = fit$coefficients$beta,
   gamma = fit$coefficients$gamma,
   covariates = fit$parameters$covariates,
   short_range = fit$coefficients$short_range,
```

```r
   medium_range = fit$coefficients$medium_range,
   long_range = fit$coefficients$long_range,
   nthreads = 2,
   saturation = fit$parameters$saturation)

lin_pred %>%
  dplyr::select(dplyr::contains("mr") | dplyr::contains("sr")) %>%
  tibble::rownames_to_column() %>%
  tidyr::pivot_longer(-1, names_to = c(".value", "set"),
                      names_sep = "[_]") -> d

sum_cols <- colnames(d)[-(1:2)]

d <- d %>%
  dplyr::mutate(overall = purrr::pmap_dbl(list(!!!rlang::parse_exprs(sum_cols)),
                                          sum)) %>%
  tidyr::pivot_wider(names_from = set,
                     values_from = c(!!!rlang::parse_exprs(sum_cols),overall))

lin_pred <- lin_pred %>%
  tibble::rownames_to_column() %>%
  dplyr::left_join(d) %>%
  dplyr::select(-rowname) %>%
  dplyr::relocate(x,y)
```

```
Joining with `by = join_by(rowname, `mr_B cells`, `mr_CD163+ macros`, `mr_CD8+
T cells`, mr_granulocytes, `mr_memory CD4+ T`, `mr_plasma cells`, mr_stroma,
mr_Tregs, `mr_tumor cells`, `sr1_B cells`, `sr1_CD163+ macros`, `sr1_CD8+ T
cells`, sr1_granulocytes, `sr1_memory CD4+ T`, `sr1_plasma cells`, sr1_stroma,
sr1_Tregs, `sr1_tumor cells`)`
```

```r
top_linpred <- lin_pred %>%
  select(x,y,contains("overall"),beta0) %>%
  pivot_longer(-c(x,y)) %>%
  group_by(x,y) %>%
  top_n(1,abs(value)) %>%
  ungroup() %>%
  select(-value) %>%
  mutate(name = factor(name)) %>%
  rename(top_pred = name)
```

```
top_linpred %>%
  filter(!(top_pred %in% c("outside"))) %>%
  mutate(top_pred = as.character(top_pred)) %>%
  mutate(top_pred = ifelse(top_pred == "beta0","tumor cells",top_pred)) %>%
  mutate(top_pred = gsub("overall_","",top_pred)) %>%
  ggplot(aes(x,y,fill=top_pred)) +
  geom_tile() +
  scale_fill_manual(values=as.vector(
    tableau20(n=length(unique(top_linpred$top_pred))))) +
  theme_minimal() +
  labs(x = "X (microns)", y = "Y (microns)",fill="Top \npredictor")
```



Next, we can estimate the AUC of each estimated conditional intensity, that is, how well the conditional intensity of each cell type correlates with the locations of each cell type.

```
# papangelou
start <- Sys.time()
papangelous <- get_papangelous(ppjsdm_fit,nthreads = 2)
```

```
[1] "B cells"
[1] "CD163+ macros"
[1] "CD8+ T cells"
```

```
[1] "granulocytes"
[1] "memory CD4+ T"
[1] "plasma cells"
[1] "stroma"
[1] "Tregs"
[1] "tumor cells"
```

```r
Sys.time() - start
```

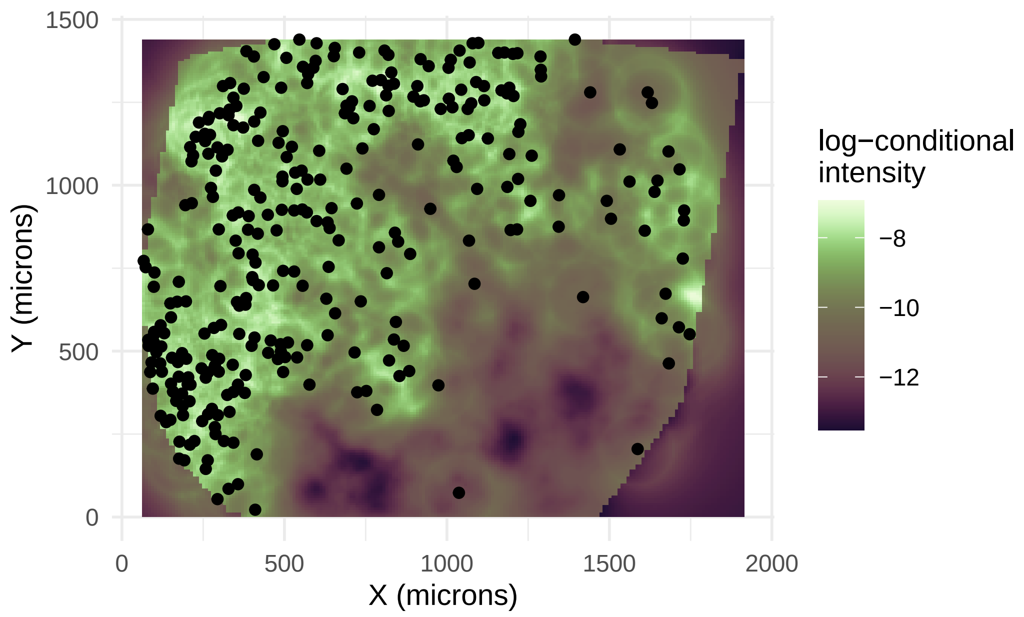
```
Time difference of 1.283638 mins
```

```r
aucs <- auc_papangelous(papangelous,ppjsdm_fit)

aucs %>%
  arrange(AUC)
```

```
# A tibble: 9 x 2
  type             AUC
  <chr>          <dbl>
1 memory CD4+ T 0.724
2 Tregs         0.747
3 stroma        0.760
4 CD163+ macros 0.812
5 CD8+ T cells  0.813
6 granulocytes  0.818
7 plasma cells  0.836
8 B cells       0.854
9 tumor cells   0.870
```

We can also visualize this conditional intensity for, eg, plasma cells:

```r
# plot papangelou
plot_papangelous(papangelous,ppjsdm_fit,types = c("plasma cells"))[[1]] +
  theme_minimal() +
  labs(title = "",x = "X (microns)",y = "Y (microns)",
       fill = "log-conditional \nintensity")
```



INTEGRATING ACROSS MULTIPLE PATIENTS

Here, I include model fits as estimated on each of the 140 images from the Schurch et al dataset[3]. The model estimation proceeds just as it did for the model fit on spot 59_A above. I have included code in the script “hpc.R” for fitting these models in parallel on a cluster.

```
models <- readRDS("data/fits_outside_vasc.rds")

slide_ids <- pt_data$Spot
```

We can also see the median AUC of each cell type in each patient group:

```
aucs <- lapply(models,\(m) enframe(m$aucs) %>% mutate(Spot=m$spot)) %>%
  bind_rows() %>%
  rename(type=name,AUC=value)

aucs %>%
  left_join(pt_data,by="Spot") %>%
  group_by(type) %>%
  summarise(All = round(median(AUC),2)) -> all_auc

aucs %>%
  left_join(pt_data,by="Spot") %>%
  group_by(type,Group) %>%
  summarise(AUC = round(median(AUC),2)) %>%
  pivot_wider(names_from = Group,values_from = AUC) %>%
  right_join(all_auc) %>%
  kable(booktabs = T) %>%
  kable_paper(full_width=F) %>%
  add_header_above(c(" ", "AUC" = 3))
```

```
`summarise()` has grouped output by 'type'. You can override using the
`.groups` argument.
Joining with `by = join_by(type)`
```

	AUC		
type	CLR	DII	All
B cells	0.83	0.82	0.82
CD163+ macros	0.72	0.69	0.70
CD4+ T cells	0.81	0.73	0.77
CD68+ macros	0.80	0.76	0.77
CD8+ T cells	0.77	0.74	0.75
NK cells	NA	0.72	0.72
Tregs	0.80	0.77	0.78
adipocytes	0.80	0.78	0.80
generic immune	0.76	0.76	0.76
granulocytes	0.77	0.73	0.75
memory CD4+ T	0.75	0.73	0.74
plasma cells	0.79	0.80	0.79
smooth muscle	0.77	0.75	0.76
stroma	0.72	0.70	0.70
tumor cells	0.72	0.76	0.75

```
df_raw %>%
  left_join(pt_data %>% select(Spot,Group)) %>%
  count(type,Spot,Group) %>%
  group_by(type,Group) %>%
  summarise(median = median(n)) %>%
  pivot_wider(names_from = Group,values_from = median)
```

```
Joining with `by = join_by(Spot)`
```

```
`summarise()` has grouped output by 'type'. You can override using the
`.groups` argument.
```

```
# A tibble: 16 x 3
# Groups:   type [16]
   type              CLR   DII
   <fct>           <dbl> <dbl>
 1 adipocytes        8.5    4
 2 B cells          14     17
 3 CD163+ macros   172.   316.
 4 CD4+ T cells      5.5    2
 5 CD68+ macros      9.5   20
 6 CD8+ T cells     62    109
 7 generic immune   12      9
 8 granulocytes     27    124.
 9 memory CD4+ T    53     80
10 NK cells          1.5    1
11 plasma cells     28     22
12 smooth muscle    77    124
13 stroma          104    158.
14 Tregs             8     21
15 tumor cells     102.   222
16 vasculature      61.5   75
```

as well as the correlation of the abundance of each cell type with the AUC of that cell type:

```
aucs %>%
  left_join(df_raw %>% count(type,Spot)) %>%
  group_by(type) %>%
  summarise(correlation = round(cor(AUC,n),2)) %>%
  kable(booktabs = T) %>%
  kable_paper(full_width=F)
```

```
Joining with `by = join_by(type, Spot)`
```

type	correlation
B cells	-0.18
CD163+ macros	-0.39
CD4+ T cells	-0.30
CD68+ macros	-0.41

CD8+ T cells	-0.49
NK cells	-0.99
Tregs	-0.35
adipocytes	-0.42
generic immune	-0.59
granulocytes	-0.54
memory CD4+ T	-0.29
plasma cells	-0.41
smooth muscle	-0.58
stroma	-0.49
tumor cells	-0.16

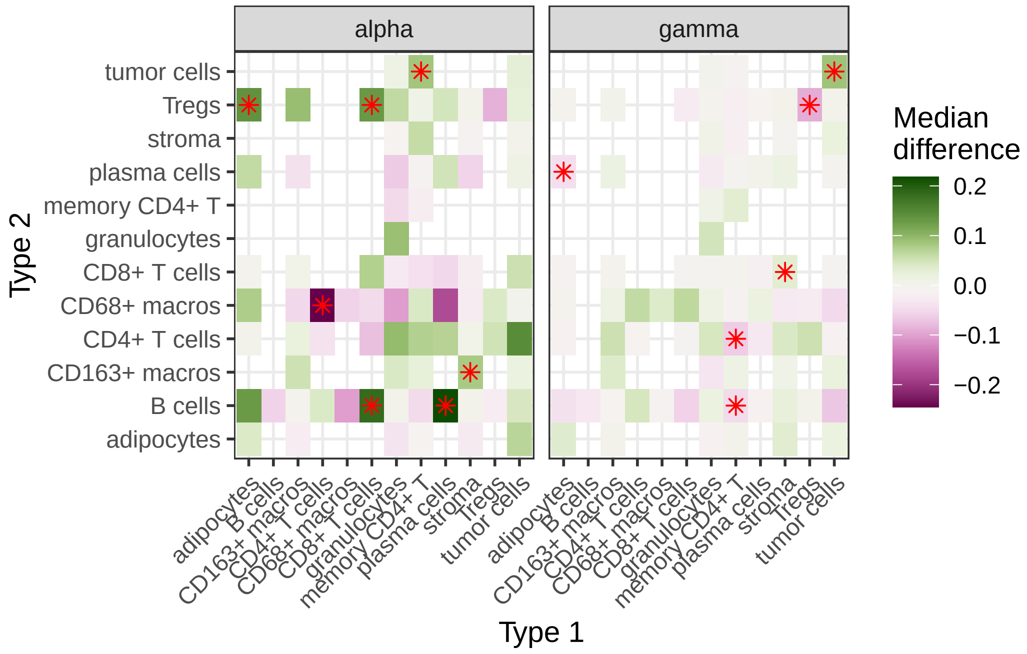
Let's collect all of the coefficients for short- and medium-range interactions into dataframes.

```r
gamma <- map(1:length(models),\(i) {
  model <- models[[i]]
  types <- df_raw %>%
    filter(Spot == model$spot) %>%
    filter(type != "vasculature") %>%
    pull(type) %>%
    unique() %>%
    as.vector()

  coefs_from_summary(model$summary,types) %>%
    filter(str_detect(name,"gamma")) %>%
    mutate(Spot = model$spot) %>%
    rename(gamma=coefficients) %>%
    select(-name)
}) %>%
  bind_rows()

alpha <- map(1:length(models),\(i) {
  model <- models[[i]]
  types <- df_raw %>%
    filter(Spot == model$spot) %>%
    filter(type != "vasculature") %>%
    pull(type) %>%
    unique() %>%
    as.vector()

  coefs_from_summary(model$summary,types) %>%
    filter(str_detect(name,"alpha")) %>%
    mutate(Spot = model$spot) %>%
    rename(alpha=coefficients) %>%
    select(-name)
}) %>%
  bind_rows()
```

```r
pt_data %>%
  filter(Group == "CLR") %>%
  group_by(Patient) %>%
  slice(3:4) %>%
  select(Patient,Group,Spot) %>% pull(Spot) -> throwout_spots
```

```r
alpha %>%
  left_join(gamma) %>%
  left_join(pt_data %>% select(Spot,Group)) %>%
  pivot_longer(c(alpha,gamma)) %>%

  group_by(type1,type2,name) %>%
  group_modify(~{
    tryCatch({
      wilcox.test(value ~ Group,data=.x,conf.int = T)
      },error=\(e) {
        NULL
      },warning=\(w) NULL) %>%
      broom::tidy()
  }) %>%
  ungroup() %>%
  mutate(p.adj = p.adjust(p.value)) %>%
  ggplot(aes(type1,type2,fill=estimate)) +
  geom_tile() +
  facet_wrap(~name) +
  theme_bw() +
  sig_stars(p_values = "p.value") +
  anglex() +
  scico::scale_fill_scico(palette="bam") +
  labs(x="Type 1",y="Type 2",fill="Median\ndifference")
```

```
Joining with `by = join_by(type1, type2, Spot)`
Joining with `by = join_by(Spot)`
```

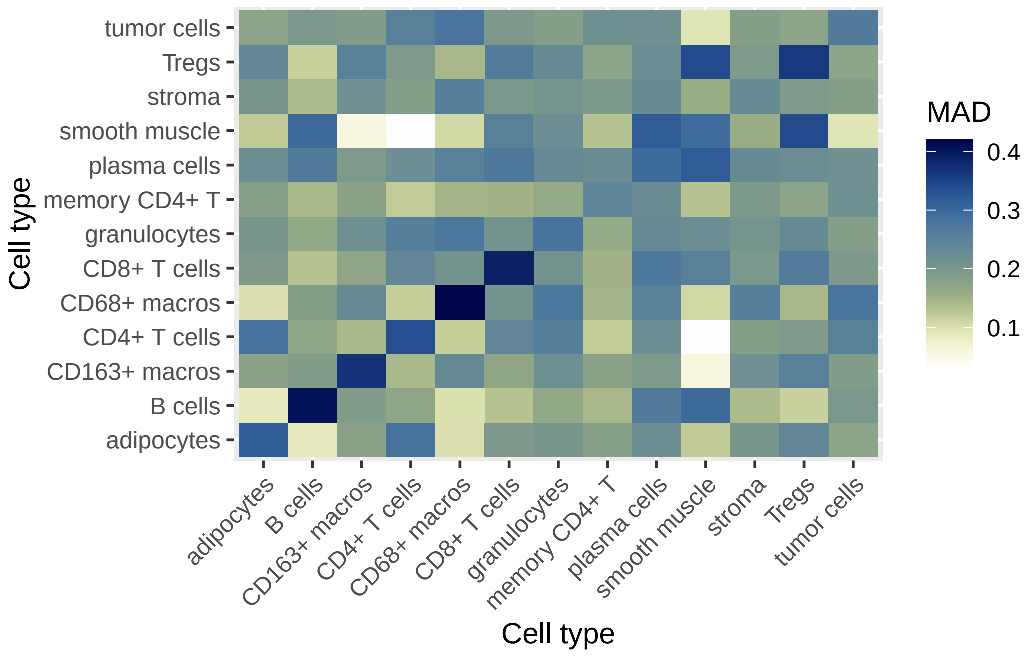


Next, let's examine the heterogeneity in interaction effect estimates across all patients by estimating the median absolute deviation (MAD) of each coefficient:

```r
alpha %>%
  left_join(pt_data) %>%
  group_by(type1,type2) %>%
  summarise(sd_coef = sd(alpha,na.rm = T),
            MAD = mad(alpha,na.rm = T)) %>%
  ungroup() -> d
```

```
Joining with `by = join_by(Spot)`
`summarise()` has grouped output by 'type1'. You can override using the
`.groups` argument.
```

```r
d %>%
  mutate(temp = type2,
         type2 = type1,
         type1 = temp) %>%
  select(-temp) -> flipped

bind_rows(d,flipped) %>%
  drop_na() %>%
  ggplot(aes(type1,type2,fill=MAD)) +
  geom_tile() +
  anglex() +
  scico::scale_fill_scico(palette = "davos",direction = -1) +
  labs(x = "Cell type",y = "Cell type")
```

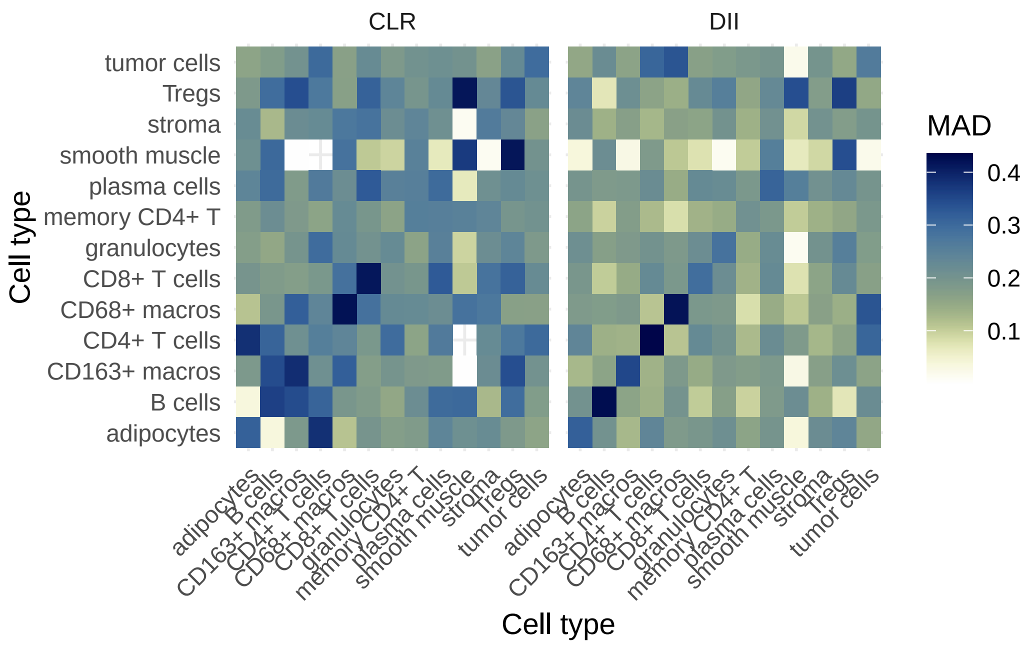


Let's do the same within each patient group:

```
alpha %>%
  left_join(pt_data) %>%
  group_by(type1,type2,Group) %>%
  summarise(sd_coef = sd(alpha,na.rm = T),
            MAD = mad(alpha,na.rm = T)) %>%
  ungroup() -> d
```

```
Joining with `by = join_by(Spot)`
`summarise()` has grouped output by 'type1', 'type2'. You can override using
the `.groups` argument.
```

```
d %>%
  mutate(temp = type2,
         type2 = type1,
         type1 = temp) %>%
  select(-temp) -> flipped

bind_rows(d,flipped) %>%
  drop_na() %>%
  ggplot(aes(type1,type2,fill=MAD)) +
  geom_tile() +
  theme_minimal() +
  anglex() +
  scico::scale_fill_scico(palette= "davos",direction = -1) +
  facet_wrap(~Group,ncol = 2) +
  labs(x = "Cell type",y = "Cell type")
```



Lastly, let's see how well each interaction predicts patient outcomes. We do that by fitting univariate Cox PH models to each interaction coefficient, as detailed in the paper:

```
gamma_df <- gamma %>%
  left_join(pt_data) %>%
  filter(!is.na(gamma)) %>%
  group_by(type1,type2) %>%
  group_modify(~{
    tryCatch({
```

```r
      if(nrow(.x) < 10) stop("Sample size is too small!")
      coxph(Surv(OS,OS_Censor)~gamma+Age+Sex,data=.x,cluster=Patient)
    },error=\(e) {
      # print(e)
      NULL
      },
    warning = \(w) {
      print(w)
      NULL
    }) %>%
      broom::tidy(conf.int=TRUE)
  }) %>%
  ungroup() %>%
  filter(term == "gamma") %>%
  filter(!str_detect(term,"Intercept")) %>%
  mutate(p.adj = p.adjust(p.value,method="fdr"))
```

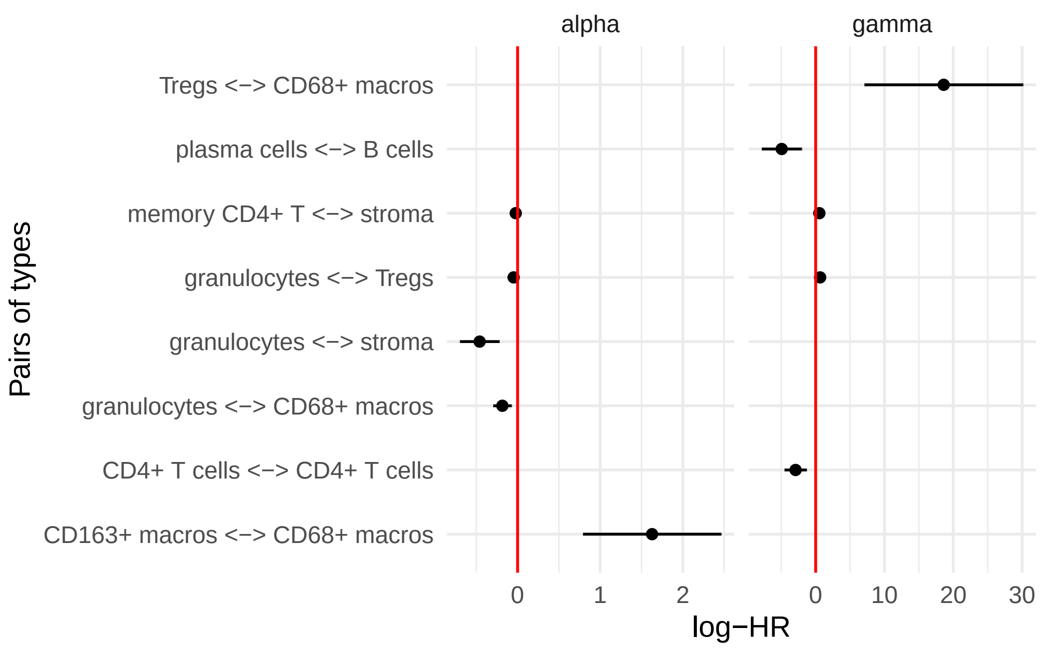
```
Joining with `by = join_by(Spot)`
```

```r
alpha_df <- alpha %>%
  left_join(pt_data) %>%
  group_by(type1,type2) %>%
  group_modify(~{
    tryCatch({
      if(nrow(.x) < 10) stop("Sample size is too small!")
      coxph(Surv(OS,OS_Censor)~alpha+Age+Sex,data=.x,cluster=Patient)
    },
    error=\(e) {
      # print(e)
      NULL
    },
    warning=\(w) {
      print(w)
      NULL
    }) %>%
      broom::tidy(conf.int=TRUE)
  }) %>%
  ungroup() %>%
  filter(!str_detect(term,"Intercept")) %>%
  mutate(p.adj = p.adjust(p.value,method="fdr"))
```

```
Joining with `by = join_by(Spot)`
```

We can plot these coefficients in a forest plot:

```r
gamma_df <- gamma_df %>%
  mutate(name = "gamma")
alpha_df %>%
  mutate(name = "alpha") %>%
  bind_rows(gamma_df) %>%
  mutate(p.adj = p.adjust(p.value,method = "fdr")) %>%
  filter(p.adj < 0.05) %>%
  unite(t1_t2,type1,type2,sep = " <-> ") %>%
  ggplot(aes(estimate,t1_t2)) +
  geom_point() +
  geom_errorbarh(aes(xmin=conf.low,xmax=conf.high,y=t1_t2),height=0) +
  geom_vline(xintercept = 0,color="red") +
  labs(x="log-HR",y="Pairs of types") +
  facet_wrap(~name,scales = "free_x") +
  theme_minimal()
```



1. Eliason, J. & Rao, A. Investigating Ecological Interactions in the Tumor Microenvironment Using Joint Species Distribution Models for Point Patterns. *The New England Journal of Statistics in Data Science* **2**, 296–310 (2024).

2. Flint, I., Golding, N., Vesk, P., Wang, Y. & Xia, A. The Saturated Pairwise Interaction Gibbs Point Process as a Joint Species Distribution Model. *Journal of the Royal Statistical Society Series C: Applied Statistics* **71**, 1721–1752 (2022).
3. Schürch, C. M. *et al.* Coordinated Cellular Neighborhoods Orchestrate Antitumoral Immunity at the Colorectal Cancer Invasive Front. *Cell* **182**, 1341–1359.e19 (2020).