Contrasts
All vs. Control
Dunnet vs. Holm vs. TukeyHSD

Lukas Graz

2023-05-04

Contents

TLDR **1**

Purpose and Structure of this Document **1**

Help functions **2**

Simulations **3**

Results **4**

Plot **5**

Results **5**

TLDR

Even though Dunnet is proven to be optiomal (in all vs comparison) for a reasonable group number (around 10) Holm seems to yield basically the same result, while being a more general method. When increasing the group number (e.g. to 30) Dunnet is more powerful.

Purpose and Structure of this Document

In many cases, we may want to compare new alternatives to a standard treatment or control. To do this, we use a linear model and perform hypothesis testing of all alternatives versus the control. The purpose of this document is to compare the power of different multiple testing techniques.

The document is structured as follows:

1. Define helper functions, including:
    - data generation
    - multiple testing functions
    - wrapper for simulations
2. Perform simulations for various settings.
3. Plot the results.

Note that there are different definitions of "power" in this context. We will be using the following definitions:

- The expectation that any of the non-placebos will be rejected.

- The expectation of the fraction of non-placebos that will be rejected.
- The expectation that all non-placebos will be rejected.

Help functions

```
library(multcomp)
library(PMCMRplus)
library(ggplot2)

set.seed(123)

# g groups, of which one is the control, n_effect of which have the effect `effect`
get_data <- function(g = 10, n_t = 5, n_c = 5, effect = 1, n_effect = 3) {
  fac <- as.factor(rep(c("ctrl", c(LETTERS, letters)[1:(g - 1)]), c(n_c, rep(n_t, g - 1))))
  groups <- relevel(fac, ref = "ctrl")
  y <- rnorm(groups, c(rep(0, length(groups) - n_t * n_effect), rep(effect, n_t * n_effect)))
  data.frame(group = groups,
    y = y
  )
}

holm <- function(data) {
  fit <- lm(y ~ group, data)
  p_vals <- summary(fit)$coefficients[-1, "Pr(>|t|)"]
  p.adjust(p_vals)
}

none <- function(data) {
  fit <- lm(y ~ group, data)
  summary(fit)$coefficients[-1, "Pr(>|t|)"]
}

# dunn <- function(data) {
#   fit <- aov(y ~ group, data)
#   PMCMRplus::dunnettTest(fit)$p.value[, ]
# }

tukey_hsd <- function(data) {
  fit <- aov(y ~ group, data)
  a <- TukeyHSD(fit)
  contrasts <- grep("ctrl", rownames(a$group), value = TRUE)
  a$group[contrasts, "p adj"]
}

# multcomp implementation of "dunnet"
dunn <- function(data) {
  n <- table(data$group)
  names(n) <- levels(data$group)
  c(summary(
    glht(aov(y ~ group, data), linfct = mcp(group = "Dunnett"))
  )$test$pvalues)
}
```

```
all <- function(data) as.matrix(data.frame(none = none(data),
  holm = holm(data), dunn = dunn(data), tukeyHSD = tukey_hsd(data))
)

dosim <- function(n_replicate = 200, n_goups = 10, n_treatmentgroup = 5,
                  n_controlgroup = 5, effect = 1, n_non_placebo_treatments = 3) {
  obj <- mcreplicate::mc_replicate(n_replicate, all(get_data(
    g = n_goups, n_t = n_treatmentgroup, n_c = n_controlgroup, effect = effect,
    n_effect = n_non_placebo_treatments
  )))
  stopifnot(effect > 0)
  if (n_non_placebo_treatments == 0) {
    p_val_non_placebo <- obj
    p_val_non_placebo[, , ] <- 1   # this is used for power calc. But in this case we have no effect
  } else {
    p_val_non_placebo <- obj[(n_goups - n_non_placebo_treatments):(n_goups - 1), , ]
  }
  p_val_no_effect <- obj[1:(n_goups - n_non_placebo_treatments - 1), , ]

  # The fraction of rejected tests within the non-placebo treatments
  MeanPower <- apply(p_val_non_placebo, 2, function(x) mean(x < 0.05))
  names(MeanPower) <- paste0("MeanPower_", names(MeanPower))

  # The fraction where all non-placebo treatments where detected
  AllPower  <- apply(p_val_non_placebo, c(2, 3), function(x) base::all(x < 0.05))
  AllPower  <- apply(AllPower, 1, function(x) mean(x))
  names(AllPower) <- paste0("AllPower_", names(AllPower))

  # The fraction where ANY non-placebo treatment was detected
  AnyPower  <- apply(p_val_non_placebo, c(2, 3), function(x) base::any(x < 0.05))
  AnyPower  <- apply(AnyPower, 1, function(x) mean(x))
  names(AnyPower) <- paste0("AnyPower_", names(AnyPower))

  # The FWER under the NULL (i.e. for placebo treatments)
  any_positive <- apply(p_val_no_effect, c(2, 3), function(x) any(x < 0.05))
  alpha <- apply(any_positive, 1, function(x) mean(x))
  names(alpha) <- paste0("alpha_", names(alpha))

  as.matrix(c(
    n_replicate = n_replicate, n_goups = n_goups, n_treatmentgroup = n_treatmentgroup,
    n_controlgroup = n_controlgroup, effect = effect,
    n_non_placebo_treatments = n_non_placebo_treatments,
    MeanPower, AllPower, AnyPower, alpha
  ), ncol = 1)
}
```

Simulations

```
n_rep <- 20000
a <- dosim(n_replicate = n_rep, n_non_placebo_treatments = 0) # NULL
```

```
g <- dosim(n_replicate = n_rep)
```

```
b <- dosim(n_replicate = n_rep, effect = .5)
```

```
c <- dosim(n_replicate = n_rep, effect = 2)
```

```
# 12=floor(n_t * sqrt(g-1)) ==> taking 14 yields the same total sample size
d <- dosim(n_replicate = n_rep, n_treatmentgroup = 4, n_controlgroup = 14)
```

```
e <- dosim(n_replicate = n_rep, n_treatmentgroup = 4, n_controlgroup = 14, effect = 2)
```

```
f <- dosim(n_replicate = n_rep, n_treatmentgroup = 4, n_controlgroup = 14,
           n_non_placebo_treatments = 7)
```

```
h <- dosim(n_replicate = n_rep, n_goups = 30, n_non_placebo_treatments = 3)
```

```
i <- dosim(n_replicate = n_rep, n_goups = 30, n_non_placebo_treatments = 10)
```

```
j <- dosim(n_replicate = n_rep, n_goups = 30, n_non_placebo_treatments = 20)
```

```
k <- dosim(n_replicate = n_rep, n_treatmentgroup = 20, n_controlgroup = 20)
```

```
# optimal sample allocation with equal total
l <- dosim(n_replicate = n_rep, n_treatmentgroup = 17, n_controlgroup = 47)
```

Results

```
result <- cbind(
  null=a,
  default=g,
  `-effect`=b,
  `+effect`=c,
  `+alloc` = d,
  `+alloc+effect` = e,
  `+nonplacebo_grps`=f,
  `+grps`=h,
  `+grps+nonplacebo`=i,
  `+grps++nonplacebo`=j,
  `+groupsize`=k,
  `+grpsize+alloc`=l
)
colnames(result) <- c("null", "default", "`-effect`", "`+effect`", "`+alloc`",
  "`+alloc+effect`", "`+nonplacebo_grps`", "`+grps`", "`+grps+nonplacebo`",
  "`+grps++nonplacebo`", "`+groupsize`", "`+grpsize+alloc`"
)
options(digits=4)
as.data.frame(result)[1:6,]
```

	null	default	-effect	+effect	+alloc	+alloc+effect	+nonplacebo_grps	+grps	+grps+nonplacebo	+grps++nonplacebo	+groupsize	+grpsize+alloc
n__replicate	20000	20000	2e+04	20000	20000	20000	20000	20000	20000	20000	20000	20000
n__goups	10	10	1e+01	10	10	10	10	30	30	30	10	10
n__treatmentgroup	5	5	5e+00	5	4	4	4	5	5	5	20	17
n__controlgroup	5	5	5e+00	5	14	14	14	5	5	5	20	47
effect	1	1	5e-01	2	1	2	1	1	1	1	1	1
n__non__placebo__treatments	0	3	3e+00	3	3	3	7	3	10	20	3	3

```
as.data.frame(result)[7:nrow(result),]
```

	null	default	-effect	+effect	+alloc	+alloc+effect	+nonplacebo_grps	+grps	+grps+nonplacebo	+grps++nonplacebo	+groupsize	+grpsize+alloc
MeanPower__none	0.0000	0.3400	0.1191	0.8701	0.4053	0.9297	0.4043	0.3462	0.3468	0.3515	0.8832	0.9394
MeanPower__holm	0.0000	0.1050	0.0215	0.6181	0.1415	0.7395	0.1489	0.0570	0.0596	0.0634	0.6548	0.7868
MeanPower__dunn	0.0000	0.1225	0.0272	0.6423	0.1444	0.7320	0.1420	0.0751	0.0760	0.0763	0.6714	0.7776
MeanPower__tukeyHSD	0.0000	0.0510	0.0088	0.4447	0.0691	0.5735	0.0693	0.0144	0.0147	0.0145	0.4828	0.6318
AllPower__none	0.0000	0.1298	0.0221	0.7345	0.1216	0.8221	0.0240	0.1316	0.0348	0.0155	0.7521	0.8424
AllPower__holm	0.0000	0.0254	0.0025	0.4009	0.0170	0.4880	0.0045	0.0080	0.0013	0.0010	0.4308	0.5550
AllPower__dunn	0.0000	0.0263	0.0024	0.4122	0.0143	0.4640	0.0008	0.0118	0.0012	0.0004	0.4350	0.5289
AllPower__tukeyHSD	0.0000	0.0067	0.0006	0.2165	0.0040	0.2739	0.0001	0.0008	0.0000	0.0000	0.2370	0.3254
AnyPower__none	0.0000	0.5726	0.2527	0.9717	0.7178	0.9964	0.8989	0.5847	0.7962	0.8825	0.9780	0.9982
AnyPower__holm	0.0000	0.2136	0.0515	0.8186	0.3189	0.9422	0.5112	0.1283	0.2617	0.3626	0.8586	0.9645
AnyPower__dunn	0.0000	0.2532	0.0660	0.8492	0.3322	0.9468	0.5307	0.1667	0.3232	0.4336	0.8802	0.9666
AnyPower__tukeyHSD	0.0000	0.1162	0.0228	0.6795	0.1732	0.8535	0.3137	0.0373	0.0891	0.1316	0.7326	0.9031
alpha__none	0.2600	0.2022	0.2077	0.2019	0.2424	0.2387	0.0965	0.4606	0.4093	0.2676	0.2034	0.2435
alpha__holm	0.0382	0.0309	0.0306	0.0366	0.0322	0.0429	0.0152	0.0304	0.0275	0.0174	0.0333	0.0393
alpha__dunn	0.0497	0.0367	0.0384	0.0385	0.0326	0.0345	0.0128	0.0466	0.0382	0.0213	0.0338	0.0326
alpha__tukeyHSD	0.0122	0.0110	0.0092	0.0100	0.0095	0.0107	0.0032	0.0046	0.0032	0.0015	0.0076	0.0086

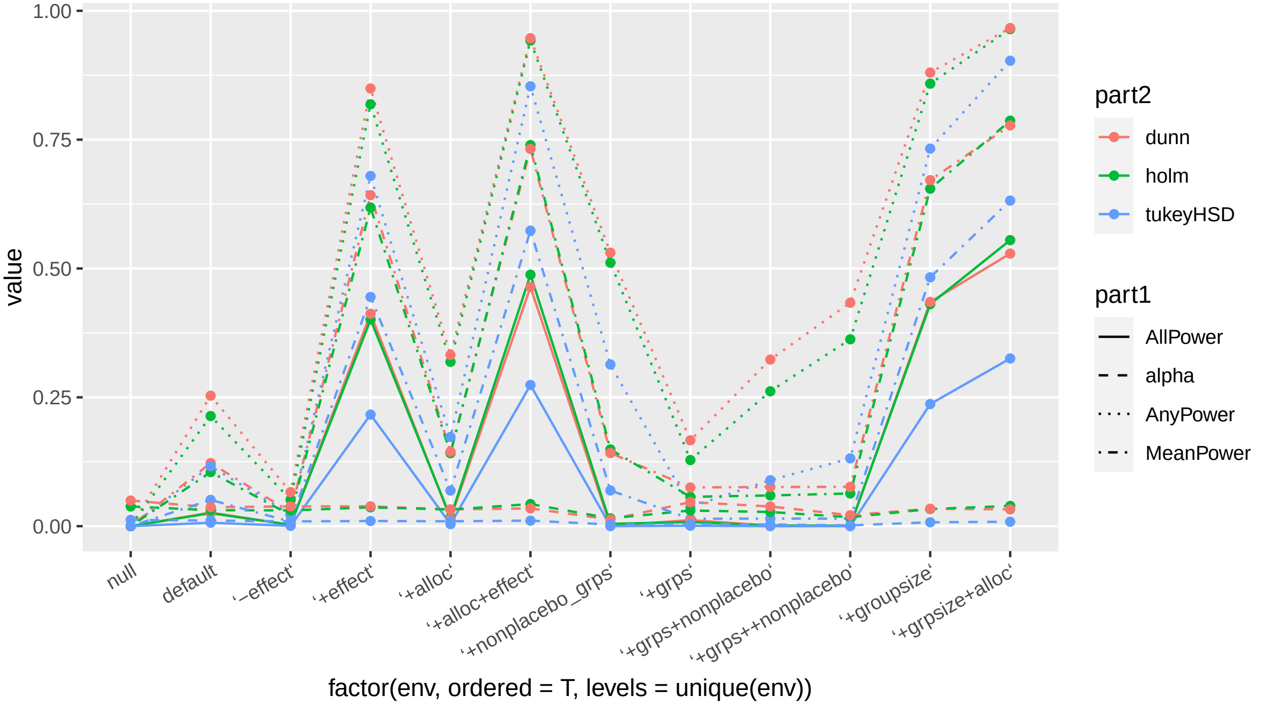
Plot

```r
X <- result[-c(1:7, 11, 15, 19),] # remove "none" rows, and parameters

# convert matrix to data frame
df <- data.frame(env = rep(colnames(X), each = nrow(X)),
                 strategy = rep(rownames(X), times = ncol(X)),
                 value = as.vector(X))

# split the strategy names into two parts
df$part1 <- gsub("^(.*?)_.*$", "\\1", df$strategy)
df$part2 <- gsub("^.*?_(.*)$", "\\1", df$strategy)

# plot
plt <- ggplot(df, aes(x = factor(env, ordered=T, levels=unique(env)), y = value,
  group = strategy, linetype = part1, color = part2)) +
  geom_line() +
  geom_point() +
  scale_linetype_manual(values = c("solid", "dashed", "dotted", "dotdash")) +  # set the line types
  theme(axis.text.x = element_text(angle = 30, hjust = 1)) # set x-axis labels to vertical
plt
```



Results

1. Dunnet and Holm yield very similar results when considering 10 groupos (while varying the number of non-placebo groups, the effect, and the samplesize allocation).
2. If we increase the number of groups Dunnet shows a slight advantage over Holm.
3. The optimal allocation ($n_{control} = n_{treatments}\sqrt{n_{groups} - 1}$) yields an improvement as well in both cases.