Estimation of Sample Size and Power for Dunnett's Testing Setups with Unequal Effect Sizes

Marcus Vollmer

Institute of Bioinformatics, University Medicine Greifswald

Adaptive Designs and Mutiple Testing Procedures

04 October 2018, Bremen

Content

1. Sample Size Estimation

Humane Animal Research

Universitätsmedizin GREIFSWALD

The three Rs

Principles were developed over 50 years ago as a framework for humane animal research

Replacement

Methods which avoid or replace the use of animals

Reduction

Methods which minimise the number of animals used per experiment

Refinement

Methods which minimise the suffering and improve animal welfare

www.nc3rs.org.uk

Humane Animal Research

The three Rs

Principles were developed over 50 years ago as a framework for humane animal research

Replacement

Methods which avoid or replace the use of animals

Reduction

Methods which minimise the number of animals used per experiment

Refinement

Methods which minimise the suffering and improve animal welfare

www.nc3rs.org.uk

Humane Animal Research

German animal protection law (came into force in 1972)

- *duty of disclosure* 13%
  prescribed by law (e.g. approval of pharmaceuticals, routine test of vaccines)
- *subject to approval* 87%
  Administrative regulation for execution of animal protection law
  8.3 The animal welfare officer should ensure, that appropriate biometrical methods will be deployed during the planning of the experimental project.

Administrative regulation

Humane Animal Research

German animal protection law (came into force in 1972)

- *duty of disclosure* 13%
  prescribed by law (e.g. approval of pharmaceuticals, routine test of vaccines)
- *subject to approval* 87%
  Administrative regulation for execution of animal protection law
  14.1.3.1 The commission have to support the competent authority about their decision to approve animal experiments; in their statement they should comment in particular, whether it is scientifically substantiated, that [. . .] no more animals were included in panning of the experiment than essential, to answer the question in consideration of biometrical methods.

▸ Administrative regulation

Sample Size Estimation in R

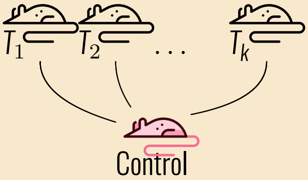
R:pwr

Function	Coverage
`pwr.2p.test`	Two proportions (equal n)
`pwr.2p2n.test`	Two proportions (unequal n)
`pwr.anova.test`	Balanced one way ANOVA
`pwr.chisq.test`	Chi-square test
`pwr.f2.test`	General linear model
`pwr.p.test`	Proportion (one sample)
`pwr.r.test`	Correlation
`pwr.t.test`	T-tests (one sample, 2 sample, paired)
`pwr.t2n.test`	T-test (two samples with unequal n)

▸ R:pwr ▸ statmethods.net

2. Dunnett’s Test

A niece of ANOVA



- statisticshowto.com
- OPAC: 703/SQ 1247-1
- Icons by Freepik from Flaticon

Multiple comparison test by Charles Dunnett (1955)

- Post-hoc-Test after ANOVA
- Compare k treatment arms against a control group
  $H_{0i} : \mu_i = \mu_0$
- Similar to performing multiple t-tests
- Designed to hold the family-wise error rate
  FWER=P(number of falsely rejected $H_0 \geq 1) \leq \alpha$
- General rule (same effect size, equal variance):
  $\frac{n_0}{n} \approx \sqrt{k}\frac{\sigma_0}{\sigma}$

Available methods in R

`R:multcomp` and `R:DTK`

Performing the special testing problem with unequal group sizes. The computation of the p-values includes the consideration of a multidimensional t-distribution and the adjustment for multiple testing. A procedure for sample size estimation is missing.

`R:DunnettTests`

Conducting sample size calculation, but only with identical treatment effect size and pre-specified sample allocation ratio. In other situations, simulation-based evaluation is suggested, which needs great computational effort.

Available methods in R

Power calculation with R:DunnettTests

```
library(DunnettTests)

#Compare group means of four treatment arms to a control arm (upper one-sided
    tests)
k     = 4   # Number of treatment arms
mu    = 2   # Assumed mean of each treatment arm
mu0 = 1   # Assumed mean of the control arm
n     = 20
n0    = 20
sigma = 1
df  = n*k+n0-k-1

# get power of the test
(power = powDT(r=k, k, mu, mu0, n, n0, "means", sigma, df, testcall="SD"))
```

```
[1] 0.7999448
```

▸ R:DunnettTests

Available methods in R

Sample size calculation with R:DunnettTests

```r
# calculate sample sizes to achive the power
nvDT(ratio=n/n0, power=0.8, r=k, k, mu, mu0, "means", sigma, dist="zdist",
    testcall="SD")
```

```
$`least sample size required in each treatment groups`
[1] 20
$`least sample size required in the control group`
[1] 20
```

▸ R:DunnettTests

Available methods in R

Dunnett's Test with R:multcomp

Implemented methods control the family-wise error rate

FWER=P(number of falsely rejected $H_0 \geq 1) \leq \alpha$

```
x = c(rnorm(n0,mu0,sigma), rnorm(n,mu,sigma), rnorm(n,mu,sigma), rnorm(n,mu,
    sigma), rnorm(n,mu,sigma))
f = gl.unequal(n=k+1, k=c(n0,n,n,n,n))

library(multcomp)
Dunnet = glht(aov(x~f, data.frame(f,x)), linfct=mcp(f="Dunnett"))
summary(Dunnet)
```

▸ R:multcomp

Available methods in R

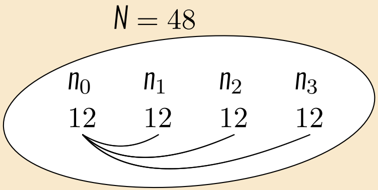
Dunnett's Test with R:multcomp

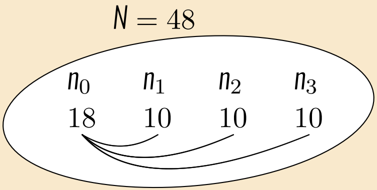
```
Simultaneous Tests for General Linear Hypotheses
Multiple Comparisons of Means: Dunnett Contrasts

Fit: aov(formula = x ~ f, data = data.frame(f, x))

Linear Hypotheses:
  Estimate Std. Error t value Pr(>|t|)
  2 - 1 == 0   1.0462     0.3121   3.352  0.00449 **
  3 - 1 == 0   1.0637     0.3121   3.408  0.00357 **
  4 - 1 == 0   1.4444     0.3121   4.628  < 0.001 ***
  5 - 1 == 0   1.1007     0.3121   3.527  0.00243 **
  ---
  Signif. codes:  0 ‘’*** 0.001 ‘’** 0.01 ‘’* 0.05 ‘’. 0.1 ‘’  1
(Adjusted p values reported -- single-step method)
```

R:multcomp

3. Idea of unbalanced testing

Balanced vs. unbalanced sample sizes

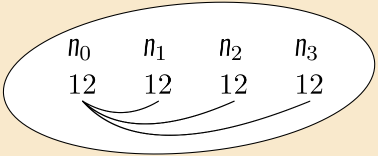




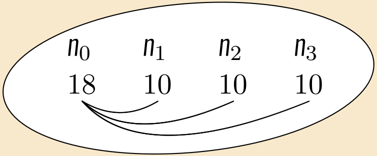
Balanced vs. unbalanced sample sizes

Universitätsmedizin GREIFSWALD

$N = 48$



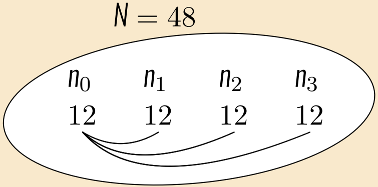
$N = 48$

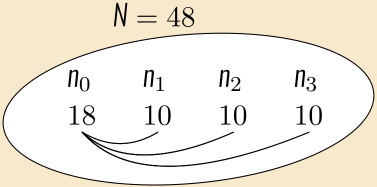


$$t = \frac{\bar{x}_0 - \bar{x}_i}{\sigma\sqrt{\frac{1}{n_0} + \frac{1}{n_i}}} = \frac{\Delta\bar{x}}{\sigma}\frac{1}{\sqrt{\frac{2}{12}}} = 2.45\frac{\Delta\bar{x}}{\sigma}$$

$$t = \frac{\bar{x}_0 - \bar{x}_i}{\sigma\sqrt{\frac{1}{18} + \frac{1}{10}}} = 2.54\frac{\Delta\bar{x}}{\sigma}$$

Balanced vs. unbalanced sample sizes





$$t = \frac{\bar{x}_0 - \bar{x}_i}{\sigma\sqrt{\frac{1}{n_0} + \frac{1}{n_i}}} = \frac{\Delta\bar{x}}{\sigma}\frac{1}{\sqrt{\frac{2}{12}}} = 2.45\frac{\Delta\bar{x}}{\sigma}$$

$$t = \frac{\bar{x}_0 - \bar{x}_i}{\sigma\sqrt{\frac{1}{18} + \frac{1}{10}}} = 2.54\frac{\Delta\bar{x}}{\sigma}$$

$$\sqrt{\frac{1}{12} + \frac{1}{12}} = \sqrt{\frac{1}{6}} = \sqrt{\frac{1}{18} + \frac{1}{9}}$$

saved

Balanced vs. unbalanced sample sizes

$N = 48$

n_0	n_1	n_2	n_3
12	12	12	12

$$t = \frac{\bar{x}_0 - \bar{x}_i}{\sigma\sqrt{\frac{1}{n_0} + \frac{1}{n_i}}} = \frac{\Delta\bar{x}}{\sigma}\frac{1}{\sqrt{\frac{2}{12}}} = 2.45\frac{\Delta\bar{x}}{\sigma}$$

$N = 48$

n_0	n_1	n_2	n_3
18	10	10	10

$$t = \frac{\bar{x}_0 - \bar{x}_i}{\sigma\sqrt{\frac{1}{\cdot} + \frac{1}{\cdot}}} = 2.54\frac{\Delta\bar{x}}{\sigma}$$

$$\sqrt{\frac{1}{12} + \frac{1}{12}} = \sqrt{\frac{1}{6}} = \sqrt{\frac{1}{18} + \frac{1}{9}}$$

saved

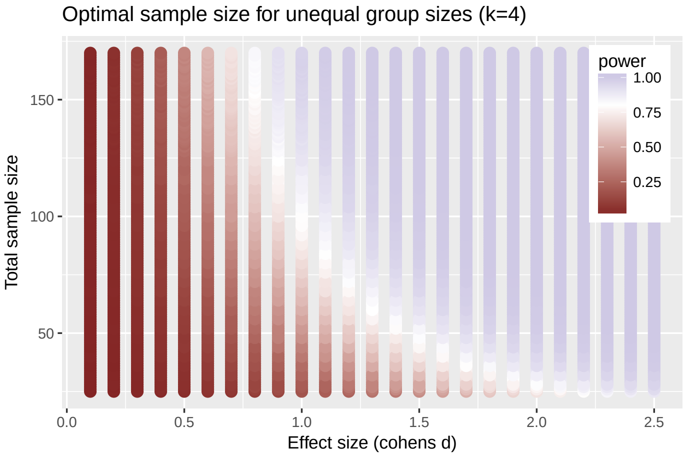
Reduction

Methods which minimise the number of animals used per experiment

4. What is the optimal set of sample sizes?

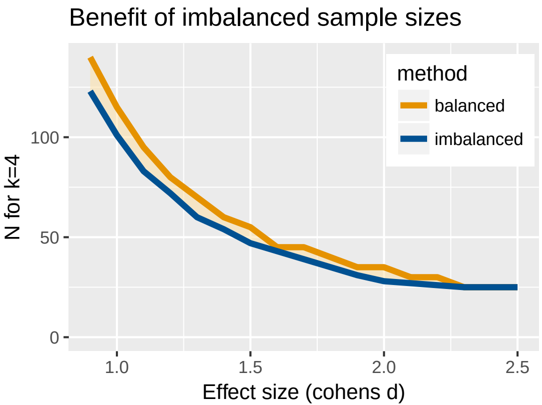
Results

- Maximal power estimated by random sampling
- All possible partition sets $\{n_0, n\}$ with $N = n_0 + k \cdot n$ included



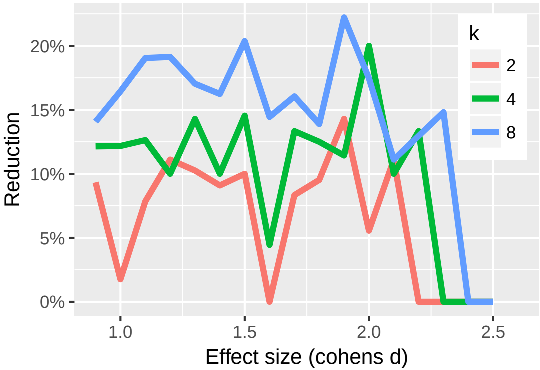
Results

- Balanced partitions sets compared against imbalanced partitions
- Difference in total sample size at 80% power



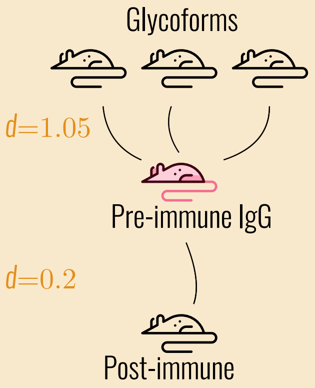
Results

- Balanced partitions sets compared against imbalanced partitions
- Difference in total sample size at 80% power
- Reduction increases with number of treatment groups k



5. A case example for animal test proposals

Passive immunization with glycoforms of IgG



Immunoglobulin G immunization of pneumococcal infected mice
Measurements from IVIS Spectrum Imaging

Assumptions

- Negative control - Pre-immune IgG: $\mu_0 = 4.58$
- Negative control - Post-immune: $\mu_1 = 5.73$
- 3 Glycoforms: $\mu_{2,3,4} = 3.57$
- Equal variance: $\sigma = 0.96$

Passive immunization with glycoforms of IgG

```
list.a = seq(32,34,1)
list.b = seq(10,12,1)
list.c = seq(18,20,1)
Power = expand.grid(a=list.a, b=list.b, c=list.c)

Power$n = Power$a + Power$b + 3*Power$c
Power$p2 = NA
Power$p3 = NA
Power$p4 = NA
Power$p5 = NA

rep = 1000
```

⋮

Passive immunization with glycoforms of IgG

Universitätsmedizin GREIFSWALD

```
for (j in 1:NROW(Power)) {
  a = Power$a[j]
  b = Power$b[j]
  c = Power$c[j]
  ng = c(a,b,c,c,c)

  p = matrix(0,rep,length(ng)-1);
  for (i in 1:rep) {
    x = c(rnorm(ng[1], mu_IgG, sd_IgG), rnorm(ng[2], mu_Negativ_PBS, sd_
        Negativ_PBS),  rnorm(ng[3], mu_glycoIgG, sd_glycoIgG), rnorm(ng[3],
        mu_glycoIgG, sd_glycoIgG), rnorm(ng[3], mu_glycoIgG, sd_glycoIgG))
    f = gl.unequal(n=5, k=ng)
    Dunnet = glht(aov(x~f, data.frame(f,x)), linfct=mcp(f="Dunnett"))
    S = summary(Dunnet)

    p[i,] = S$test$pvalues
  }
  Power[j, 5:(5+NCOL(p)-1)] = colSums(p<.05)/rep
}
```

Passive immunization with glycoforms of IgG

```
Power$n = Power$a + Power$b + 3*Power$c
Power$power = rowMeans(Power[,5:8])
View(Power[order(Power$power),])
```

	a	b	c	n	p2	p3	p4	p5	power
11	33	10	19	100	0.812	0.788	0.787	0.795	0.79550
9	34	12	18	100	0.851	0.785	0.780	0.768	0.79600
18	34	12	19	103	0.868	0.772	0.762	0.783	0.79625
22	32	11	20	103	0.798	0.809	0.800	0.789	0.79900
20	33	10	20	103	0.776	0.808	0.807	0.812	0.80075
17	33	12	19	102	0.843	0.779	0.797	0.785	0.80100
21	34	10	20	104	0.822	0.813	0.793	0.777	0.80125
12	34	10	19	101	0.811	0.791	0.784	0.827	0.80325
15	34	11	19	102	0.829	0.802	0.798	0.785	0.80350
23	33	11	20	104	0.814	0.804	0.803	0.812	0.80825
16	32	12	19	101	0.862	0.792	0.800	0.785	0.80975
26	33	12	20	105	0.864	0.803	0.795	0.806	0.81700
24	34	11	20	105	0.812	0.807	0.825	0.831	0.81875
25	32	12	20	104	0.842	0.814	0.828	0.810	0.82350
27	34	12	20	106	0.853	0.794	0.815	0.833	0.82375

Passive immunization with glycoforms of IgG

Universitätsmedizin GREIFSWALD

```
Power$n = Power$a + Power$b + 3*Power$c
Power$power = rowMeans(Power[,5:8])
View(Power[order(Power$power),])
```

	a	b	c	n	p2	p3	p4	p5	power
11	33	10	19	100	0.812	0.788	0.787	0.795	0.79550
9	34	12	18	100	0.851	0.785	0.780	0.768	0.79600
18	34	12	19	103	0.868	0.772	0.762	0.783	0.79625
22	32	11	20	103	0.798	0.809	0.800	0.789	0.79900
20	33	10	20	103	0.776	0.808	0.807	0.812	0.80075
17	33	12	19	102	0.843	0.779	0.797	0.785	0.80100
21	34	10	20	104	0.822	0.813	0.793	0.777	0.80125
12	34	10	19	101	0.811	0.791	0.784	0.827	0.80325
15	34	11	19	102	0.829	0.802	0.798	0.785	0.80350
23	33	11	20	104	0.814	0.804	0.803	0.812	0.80825
16	32	12	19	101	0.862	0.792	0.800	0.785	0.80975
26	33	12	20	105	0.864	0.803	0.795	0.806	0.81700
24	34	11	20	105	0.812	0.807	0.825	0.831	0.81875
25	32	12	20	104	0.842	0.814	0.828	0.810	0.82350
27	34	12	20	106	0.853	0.794	0.815	0.833	0.82375

Passive immunization with glycoforms of IgG

```
Power$n = Power$a + Power$b + 3*Power$c
Power$power = rowMeans(Power[,5:8])
View(Power[order(Power$power),])
```

	a	b	c	n	p2	p3	p4	p5	power
11	33	10	19	100	0.812	0.788	0.787	0.795	0.79550
9	34	12	18	100	0.851	0.785	0.780	0.768	0.79600
18	34	12	19	103	0.868	0.772	0.762	0.783	0.79625
22	32	11	20	103	0.798	0.809	0.800	0.789	0.79900
20	33	10	20	103	0.776	0.808	0.807	0.812	0.80075
17	33	12	19	102	0.843	0.779	0.797	0.785	0.80100
21	34	10	20	104	0.822	0.813	0.793	0.777	0.80125
12	34	10	19	101	0.811	0.791	0.784	0.827	0.80325
15	34	11	19	102	0.829	0.802	0.798	0.785	0.80350
23	33	11	20	104	0.814	0.804	0.803	0.812	0.80825
16	32	12	19	101	0.862	0.792	0.800	0.785	0.80975
26	33	12	20	105	0.864	0.803	0.795	0.806	0.81700
24	34	11	20	105	0.812	0.807	0.825	0.831	0.81875
25	32	12	20	104	0.842	0.814	0.828	0.810	0.82350

6. How to organize an intelligent search for an optimal set of group sizes?

Concepts

The aim is to determine the sample sizes for multiple treatment groups with different effect sizes (different means and unequal variances). A necessary statistical power of 80% is expected.

Ideas for finding the minimal set of group sizes in Monte Carlo experiments:

Random search

Start with initial size

Sample new position based on derived statistical power

Modified grid search

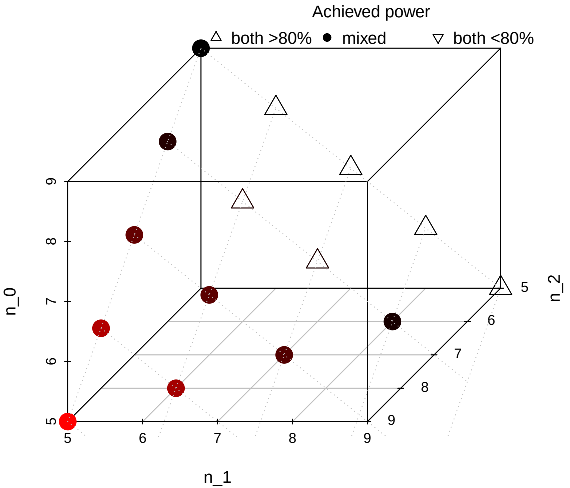
Evaluate given parameter sets – start with coarse grid – refine grid – increase accuracy (number of simulations)

Topological concept

Bottom-up or top-down procedure using the topology of different sets of sample sizes

Topological Concept

- Illustration of integer partitions with $N = n_0 + n_1 + n_2 = 19$
- (n_0, n_1, n_2) are individual group sizes

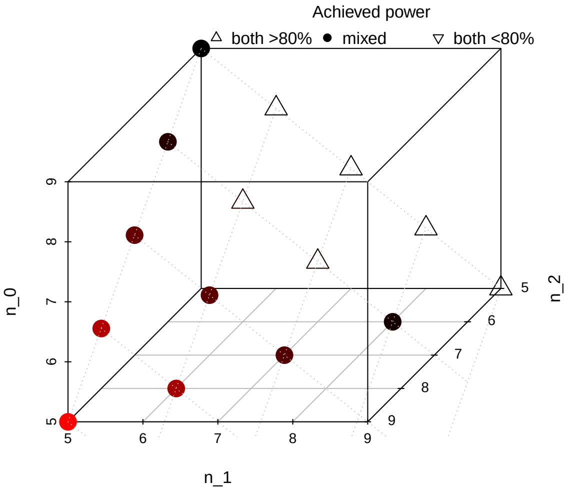


Topological Concept

- Illustration of integer partitions with $N = n_0 + n_1 + n_2 = 19$
- (n_0, n_1, n_2) are individual group sizes
- $(6, 5, 8)$ has the following neighbors in the lower integer partition with $N = 18$:

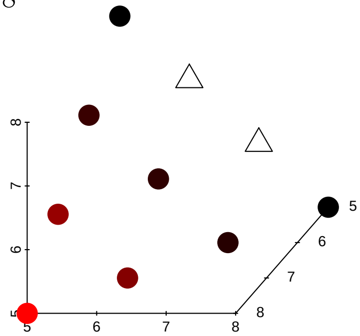
$$(5, 5, 8), (6, 4, 8), (6, 5, 7)$$

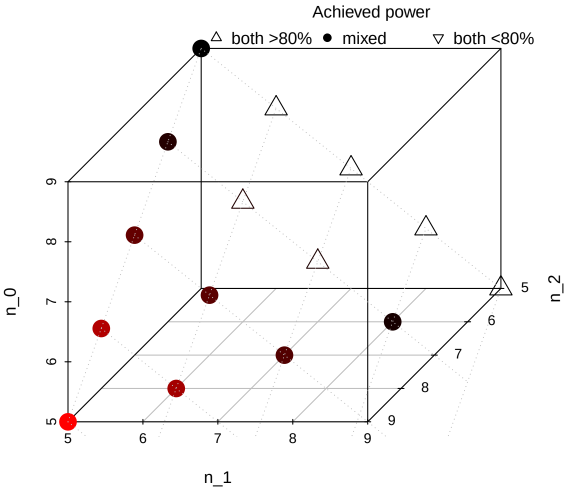
For those partitions we already know that the power is less than the power for $(6, 5, 8)$.



Topological Concept

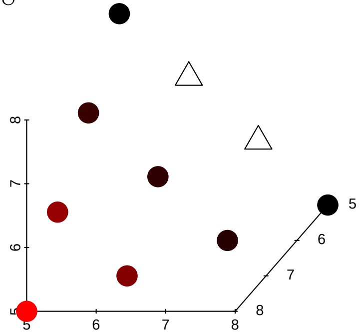
$N = 18$



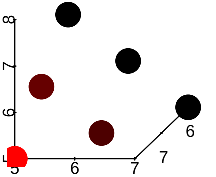


Topological Concept

$N = 18$



$N = 17$



Topological Concept

Closing remarks

- It needs roughly 500,000 samples to estimate the power precisely up to the first decimal place
- An intelligent search at different levels of integer partitions of size *N* can massively reduce the computational size
- It is the objective of constructing a random search with the use of topological relations and precision levels

... work in progress ...

Thank You for Your Attention!

7. Appendix

Simulation study on a computing cluster

```
dunnett_multisize_power <- function(n0, n, cohens_d, m=1e5, wr=FALSE){
  require(multcomp)
  require(DTK)

  mu_ctr = 0
  sd_ctr = 1
  sd_trm = 1
  mu_trm = cohens_d

  ng = c(n0,n)
  p = matrix(0,m,length(ng)-1)
```

⋮

Simulation study on a computing cluster

⋮

```
for (i in 1:m) {
  x = rnorm(ng[1], mu_ctr, sd_ctr)
  for (j in 1:(length(ng)-1)) {
    x = c(x, rnorm(ng[j+1], mu_trm[j], sd_trm))
  }
  f = gl.unequal(n=length(ng), k=ng)

  Dunnet = glht(aov(x~f, data.frame(f,x)), linfct=mcp(f="Dunnett"))
  S = summary(Dunnet)

  p[i,] = S$test$pvalues
}
```

⋮

Simulation study on a computing cluster

⋮

```
  if (wr==TRUE) {
    save(p, file=paste0("R_data/",
      "p_n0_",
      formatC(n0, width=3, format="d", flag="0"),
      "_n_",
      paste(formatC(n, width=3, format="d", flag="0"), collapse="_"),
      "_e_",
      paste(formatC(cohens_d*100, width=3, format="d", flag="0"), collapse="_")
          ,
      ".Rda")
    )
  }
}
```

Simulation study on a computing cluster: The bash!R

```
#!/usr/bin/Rscript
#### To define a name for the job (will be displayed in qstat, pbstop output):
#PBS -N equal_sizes
#PBS -m abe
#PBS -M marcus.vollmer@uni-greifswald.de,jan.zude@uni-greifswald.de
### Ressources requested: Memory and Time
#PBS -l nodes=1:ppn=40,cput=3500:00:00
###,mem=40gb,pmem=1gb
### Following are the R-commands to execute.

setwd("/mnt/staff/vollmer/dunnett")
source("dunnett_multisize_power.R")
```

⋮

Simulation study on a computing cluster: The bash!R

Universitätsmedizin GREIFSWALD

⋮

```
list.n = seq(5,50,1)
list.n_treat = 2^(1:3)
list.cohens_d = seq(.1,4,.1)

S = expand.grid(n=list.n, n_treat=list.n_treat, cohens_d=list.cohens_d)

require(doParallel)
registerDoParallel(cores=detectCores(all.tests=FALSE, logical=TRUE))
foreach(i=1:NROW(S)) %dopar% {
  m = 1e4
  n = rep(S$n[i], S$n_treat[i])
  n0 = S$n[i]
  cohens_d = rep(S$cohens_d[i], S$n_treat[i])
  }
}
```