

# Package ‘soptdmaeA’

July 23, 2025

**Version** 1.0.1

**Date** 2025-05-24

**Title** Sequential Optimal Designs for Two-Colour cDNA Microarray Experiments

**Author** Dibaba Bayisa Gemechu [aut, cre],  
Legesse Kassa Debushe [aut],  
Linda Haines [aut]

**Maintainer** Dibaba Bayisa Gemechu <diboobayu@gmail.com>

**Depends** R (>= 3.3.2), MASS, Matrix, igraph, tcltk

**License** GPL-2

**Description** Computes sequential A-, MV-, D- and E-optimal or near-optimal block and row-column designs for two-colour cDNA microarray experiments using the linear fixed effects and mixed effects models where the interest is in a comparison of all possible elementary treatment contrasts. The package also provides an optional method of using the graphical user interface (GUI) R package 'tcltk' to ensure that it is user friendly.

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2025-05-24 21:20:02 UTC

## Contents

ematbrcd.mae . . . . .	2
fixparsoptd.mae . . . . .	3
graphsoptd.mae . . . . .	4
mmenusoptd.mae . . . . .	5
soptdmae . . . . .	6
soptdmaeA-internal . . . . .	9

<b>Index</b>	<b>12</b>
--------------	-----------

---

`cmatbrcd.mae`*Computes the treatment information matrix*

---

### Description

Computes the information matrix (C-matrix) for treatment effects under either the linear fixed effects model or the linear mixed effects model setting for a given block or row-column design.

### Usage

```
cmatbrcd.mae(trt.N, blk.N, theta, des, dtype)
```

### Arguments

<code>trt.N</code>	integer, specifying number of treatments $v$ of design <code>des</code> .
<code>blk.N</code>	integer, specifying number of arrays (blocks or columns) $b$ of design <code>des</code> .
<code>theta</code>	numeric, representing a function of the ratio of random array variance and random error variance. It takes any value between 0 and 1, inclusive.
<code>des</code>	matrix, a $2 \times b$ block or row-column design with $b$ arrays and $v$ treatments.
<code>dtype</code>	character, specifying the design type. For block designs, <code>dtype = "blkd"</code> and for row-column designs, <code>dtype = "rcd"</code> .

### Value

Returns a  $v \times v$  treatment information matrix (C-matrix).

### Author(s)

Dibaba Bayisa Gemechu, Legesse Kassa Debusho, and Linda Haines

### References

Debusho, L. K., Gemechu, D. B. and Haines, L. (2018). Algorithmic construction of optimal block designs for two-colour cDNA microarray experiments using the linear mixed effects model. *Communications in Statistics - Simulation and Computation*, <https://doi.org/10.1080/03610918.2018.1429617>.

Gemechu D. B., Debusho L. K. and Haines L. M. (2014). A-optimal designs for two-colour cDNA microarray experiments using the linear mixed effects model. *Peer-reviewed Proceedings of the Annual Conference of the South African Statistical Association for 2014 (SASA 2014)*, Rhodes University, Grahamstown, South Africa. pp 33-40, ISBN: 978-1-86822-659-7.

### See Also

[soptdmaeA](#), [fixparsoptd.mae](#)

## Examples

```
##Information matrix

trt.N <- 4
blk.N <- 4
theta <- 0.3
dsgn <- rbind(1:4,c(2:4,1))
dtype <- "rcd"

cmatbrcd.mae(trt.N = 4, blk.N = 4, theta = 0.2, des = dsgn, dtype = "rcd")
```

---

fixparsoptd.mae	<i>Sets parametric values and initial design</i>
-----------------	--

---

## Description

Creates a tcltk window that allow the users to set or fix values for the parametric combinations and initial optimal or near-optimal design to compute sequential optimal or near-optimal block or row-column designs.

## Usage

```
fixparsoptd.mae(Optcrit)
```

## Arguments

Optcrit	character, specifying the optimality criteria to be used. Optcrit takes the letter "A", "MV", "D" and "E" for A-, MV-, D- and E-optimal or near-optimal designs, respectively.
---------	--

## Details

fixparsoptd.mae creates a pop-up tcltk window that allow the users to set the parametric combinations and initial optimal or near-optimal designs to compute optimal or near-optimal block or row-column designs. The parameters include the number of treatments `trt.N`, number arrays `blk.N`, theta value `theta`, number of replications of the optimization procedure `nrep`, number of added treatments `strt`, number of added arrays `sary` and a button that allows users to insert initial optimal or near-optimal design `des0` by typing into a pop-up sheet that will occur when clicking on the 'Insert' button. The users are not restricted to a specific dimension of the initial design, thus, the dimension of the initial design `des0` can be  $2 \times \text{blk.N}$  or  $\text{blk.N} \times 2$ . The default initial design `des0` is a  $3 \times 2$  loop/cyclic design.

Furthermore, on this window, the checkbox options that allow the users to choose whether or not they need to have the graphical layout of the resultant sequential optimal or near-optimal block or row-column design, to make a choice between the two design types (block design and row-column designs) and to print the summary of the resultant sequential optimal or near-optimal design on R-console directly are available.

After setting all the required parametric combinations and selecting the algorithm of interest, clicking on the search button on the set parametric combinations tcltk window, similar to the results that can be obtained when using the function [soptdmaeA](#), the summary of the resultant sequential optimal or near-optimal design will be saved in the current working directory in .csv format. The graphical layout of resultant sequential optimal or near-optimal design together with that of the initial optimal or near-optimal design will also be displayed on R console.

### Value

The `fixparsoptd.mae` function creates a pop-up tcltk window that allow the users to set the parametric combinations to compute sequential optimal or near-optimal block or row-column designs.

### Author(s)

Dibaba Bayisa Gemechu, Legesse Kassa Debusho, and Linda Haines

### See Also

[soptdmaeA](#), [mmenusoptd.mae](#)

---

<code>graphsoptd.mae</code>	<i>Creates the graphical layout of resultant sequential optimal design</i>
-----------------------------	--

---

### Description

Creates the graphical layout of resultant sequential A-, MV-, D- or E-optimal or near-optimal block or row-column design on a separate pop-up GUI tcltk window.

### Usage

```
graphsoptd.mae(trt.N, blk.N, theta, soptdesF, Optcrit, strt, sary, dtype)
```

### Arguments

<code>trt.N</code>	integer, specifying number of treatments $v$ of resultant sequential optimal design, <code>soptdesF</code> .
<code>blk.N</code>	integer, specifying number of arrays $b$ of resultant sequential optimal design, <code>soptdesF</code> .
<code>theta</code>	numeric, representing a function of the ratio of random array variance and random error variance. It takes any value between 0 and 1, inclusive.
<code>soptdesF</code>	matrix, a $2 \times b$ obtained sequential optimal or near-optimal block or row-column design.
<code>Optcrit</code>	character specifying the optimality criteria to be used. Thus, <code>Optcrit</code> takes the letter "A", "MV", "D" and "E" for A-, MV-, D- and E-optimal or near-optimal block or row-column designs, respectively.
<code>strt</code>	a non-negative integer, specifying number of added treatments/conditions to the initial design.

sary a non-negative integer, specifying number of added arrays to the initial design.  
 dtype character, specifying the design type. For block designs, dtype = "blkd" and for row-column designs, dtype = "rcd".

### Details

Detail discussions concerning the constructions of graph can be found in `igraph` R package.

### Value

Returns the graphical layout of the resultant sequential optimal or near-optimal block or row-column design 'soptdesF' on a separate pop-up window with the new edges (arrays) and vertices (treatments) added to the initial design coloured in red and brown, respectively, for separation purpose. Furthermore, the function `graphsoptd.mae` saves the graphical layout of the initial `des0` and resultant sequential optimal or near-optimal design in .pdf format in a working subdirectory.

### Note

When closing a pop-up window for graphical layout of the resultant designs (Graph plot), if the window is closed by clicking on the red button with "X" sign (top-right), the warning message "Warning message: In `rm(list = cmd, envir = .tkplot.env)`: object 'tkp ...' not found" will occur in R console irrespective of what command is executed next. To resolve this warning message, click on "close" menu that is located at the top-left of the graph plot pop-up window when closing this window.

### Author(s)

Dibaba Bayisa Gemechu, Legesse Kassa Debusho, and Linda Haines

---

mmenusoptd.mae	<i>Creates the main menu tcltk window</i>
----------------	---

---

### Description

Alternative to directly using the function `soptdmaeA` on the R console, the function `mmenusoptd.mae` creates the main menu tcltk window with the list of four optimality criteria (A, MV, D and E) for search of sequential optimal or near-optimal block or row-column designs for two-colour cDNA microarray experiments.

### Usage

```
mmenusoptd.mae()
```

### Details

The main menu window created using the function `mmenusoptd.mae` contains the list of four optimality criteria (the A-, MV -, D- and E) for sequential block or row-column designs that allows user to choose an optimality criterion of interest. The function will then call for a function `fixparsoptd.mae` for further option regarding the setup of parametric combinations, initial optimal or near-optimal design, and output, see `fixparsoptd.mae` documentation.

**Author(s)**

Dibaba Bayisa Gemechu, Legesse Kassa Debusho, and Linda Haines

**See Also**

[soptdmaeA](#), [fixparsoptd.mae](#)

---

soptdmaeA	<i>Sequential optimal designs for two-colour cDNA microarray experiments</i>
-----------	--

---

**Description**

Used to compute sequential A-, MV-, D- or E-optimal or near-optimal block and row-column designs for two-colour cDNA microarray experiments under either the linear fixed effects model or the linear mixed effects model settings using the array exchange algorithms of Debusho, Gemechu and Haines (2018) <[doi.org/10.1080/03610918.2018.1429617](https://doi.org/10.1080/03610918.2018.1429617)>.

**Usage**

```
soptdmaeA(trt.N, blk.N, theta, nrep, strt, sary, des0, dtype, Optcrit = "", ...)
```

```
## Default S3 method:
soptdmaeA(trt.N, blk.N, theta, nrep, strt, sary, des0, dtype, Optcrit = "",...)
## S3 method for class 'soptdmaeA'
print(x, ...)
## S3 method for class 'soptdmaeA'
summary(object, ...)
```

**Arguments**

trt.N	integer, specifying number of treatments $v$ of initial design, $des_0$ .
blk.N	integer, specifying number of arrays (blocks or columns) $b$ of initial design, $des_0$ .
theta	numeric, representing a function of the ratio of random array variance and random error variance. It takes any value between 0 and 1, inclusive.
nrep	integer, specifying number of replications of the optimization procedure.
strt	a non-negative integer, specifying number of added treatments/conditions to the initial design.
sary	a non-negative integer, specifying number of added arrays to the initial design.
des0	matrix, a $2 \times blk.N$ or $blk.N \times 2$ initial block or row-column design. The initial design must be treatment connected and the number of treatments and arrays should also coincides with $trt.N$ and $blk.N$ inserted by the user, if this conditions are not satisfied, the package will stop running with an error message.

<code>dtype</code>	character, specifying the design type. For block designs, <code>dtype = "blkd"</code> and for row-column designs, <code>dtype = "rcd"</code> .
<code>Optcrit</code>	character, specifying the optimality criteria to be used. <code>Optcrit</code> takes the letter "A", "MV", "D" and "E" for A-, MV-, D- and E-optimal or near-optimal block or row-column designs, respectively.
<code>x</code>	the object to be printed.
<code>object</code>	an object of class "soptdmaeA".
<code>...</code>	not used.

## Details

`soptdmaeA` computes sequential optimal or near-optimal block or row-column designs for the two-colour cDNA microarray experiments where the interest is in a comparison of all possible elementary treatment contrasts for a given initial optimal or near-optimal designs. The function computes sequential A-, MV-, D- and E-optimal or near optimal block or row-column designs via calling of four sub-functions `seqAoptbrcd.maeA`, `seqMVoptbrcd.maeA`, `seqDoptbrcd.maeA`, and `seqEoptbrcd.maeA`, respectively. These functions uses the array exchange algorithm of Debusho, Gemechu and Haines (2016). Thus, once the parametric combinations of interest are sated, these functions will first compute, randomly, a new connected initial design with a new number of arrays and, optionally, a new number of treatments. Then they perform the array exchange procedure through deletion and addition of candidate arrays at a time and selects a design with best array exchange with respect to the optimality criterion value. The candidate arrays are lists of possible arrays with different treatment combinations and their lists are dependent of the number of arrays and treatments added to the initial optimal or near-optimal design. For example, if only one treatment and one array are to be added to the initial optimal or near-optimal design, then the candidate arrays will be only those arrays that consists of a new treatment together with the old treatments in the initial optimal or near-optimal design with or without considering their position within the array for row-column or block designs, respectively.

The minimum value of `trt.N` and `blk.N` is 3 and `trt.N` should be less than or equal to `blk.N - 1`. Thus, the least initial design should be of a design with 3 number of treatments and number of arrays. The minimum number of `sary` and `strt` are 1 and 0, respectively, and `sary` should be greater than or equal to `strt`. The linear fixed effects model results for given parametric combinations and initial design are obtained by setting `theta = 0.0`.

`nrep` takes a value of greater than or equal to 1. However, to ensure optimality of the resultant design, for `sary - strt > 0`, the `nrep` should be greater than or equal to 10. In addition, as `trt.N` or `blk.N` or `sary` and/or `strt` or all of them increase, to ensure optimality of resultant design, it is advised to further increase the value of `nrep` up to greater than or equal to 50. However, it has to be noted that as `trt.N` or `blk.N` or `nrep` or all of them increase, computer time required to generate sequential optimal or near-optimal design increases.

## Value

Returns the initial and resultant sequential A-, MV-, D- or E-optimal or near-optimal block or row-column design with their corresponding score value and parametric combination saved in excel file in a working directory. In addition, the function `soptdmaeA` displays the graphical layout of the initial and resultant optimal or near-optimal block or row-column designs. Specifically:

`call`                    the method call.

v	number of treatments of obtained sequential design.
b	number of arrays of obtained sequential design.
theta	theta value.
nrep	number of replications of the optimization procedure.
strt	number of added treatments.
sary	number of added arrays.
Optcrit	optimality criteria.
optdes0	a $2 \times \text{blk.N}$ initial optimal or near-optimal block or row-column design.
optcrtsv0	score value of the optimality criteria 'Optcrit' of the initial optimal or near-optimal block or row-column design 'optdes0'.
soptdesF	a $2 \times \text{blk.N}$ obtained sequential optimal or near-optimal block or row-column design.
soptcrtsv	score value of the optimality criteria 'Optcrit' of the resultant sequential optimal or near-optimal block design 'soptdesF'.
file_loc, file_loc2	location where the summary of the resultant optimal or near-optimal block design is saved in .csv format.
equireplicate0	logical value indicating whether the initial optimal or near-optimal block or row-column design is equireplicate or not.
vtrtrep0	vector of treatment replication of the initial optimal or near-optimal block or row-column design.
equireplicate	logical value indicating whether the resultant sequential optimal or near-optimal block or row-column design is equireplicate or not.
vtrtrep	vector of treatment replication of the resultant sequential optimal or near-optimal block or row-column design.
Cmat	the C-matrix or treatment information matrix of the obtained sequential optimal or near-optimal block or row-column design.

The output also includes graphical layouts of the initial and resultant sequential optimal or near-optimal block or row-column design. The new edges (arrays) and vertices (treatments) added to the initial design are coloured in red and brown, respectively, for identification purpose.

NB: The function `soptdmaeA` also saves the summary of the initial and resultant sequential optimal or near-optimal block or row-column design in .csv format in the working directory. Furthermore, the function reports only one final sequential optimal or near-optimal block or row-column design, however, there is a possibility of more than one sequential optimal or near-optimal block or row-column designs for a given parametric combination. The function `graphsoptd.mae` can be used to view and rearrange the graphical layout of the resultant sequential optimal or near-optimal block or row-column design on tcltk window. Alternative to the function `soptdmaeA`, a GUI tcltk window can be used to generate sequential optimal or near-optimal block or row-column designs, see `mmenusoptd.mae` and `fixparsoptd.mae`.

### Author(s)

Dibaba Bayisa Gemechu, Legesse Kassa Debusho, and Linda Haines



## References

Debushe, L. K., Gemechu, D. B. and Haines, L. (2018). Algorithmic construction of optimal block designs for two-colour cDNA microarray experiments using the linear mixed effects model. *Communications in Statistics - Simulation and Computation*, <https://doi.org/10.1080/03610918.2018.1429617>.

Gemechu D. B., Debushe L. K. and Haines L. M. (2014). A-optimal designs for two-colour cDNA microarray experiments using the linear mixed effects model. *Peer-reviewed Proceedings of the Annual Conference of the South African Statistical Association for 2014 (SASA 2014)*, Rhodes University, Grahamstown, South Africa. pp 33-40, ISBN: 978-1-86822-659-7.

## See Also

[mmenusoptd.mae](#), [fixparsoptd.mae](#)

## Examples

```
##To obtain sequential A-optimal or near-optimal block design for a given
##initial A-optimal or near-optimal block design, set

trt.N <- 3 #Number of treatments
blk.N <- 3 #Number of blocks
theta <- 0 #theta value
nrep <- 10 #Number of replications
strt <- 2 #Number of added treatments
sary <- 3 #Number of added arrays
des0 <- rbind(1:3, c(2, 3, 1)) #Initial design
dtype = "blkd" #Design type
Optcrit <- "A" #Optimality criteria

seqAoptbd <- soptdmaeA(trt.N = 3, blk.N = 3, theta = 0, nrep = 10,
                      strt = 2, sary = 3, des0, dtype = "blkd", Optcrit = "A")

summary(seqAoptbd)

##To obtain sequential A-optimal or near-optimal row-column design for a given
##initial A-optimal or near-optimal row-column design des0 (stated above), set

dtype = "rcd" #Design type

seqAoptrcd <- soptdmaeA(trt.N = 3, blk.N = 3, theta = 0, nrep = 10,
                       strt = 2, sary = 3, des0, dtype = "rcd", Optcrit = "A")

summary(seqAoptrcd)
```

**Description**

Functions for internal usage only.

**Usage**

```
## Computes A-optimal or near-optimal block or row-column designs
## using array exchange algorithm
seqAoptbrcd.maeA(trt.N, blk.N, theta, nrep, strt, sary, des0, dtype)

## Computes MV-optimal or near-optimal block or row-column designs
## using array exchange algorithm
seqMVoptbrcd.maeA(trt.N, blk.N, theta, nrep, strt, sary, des0, dtype)

## Computes A-optimal or near-optimal block or row-column designs
## using array exchange algorithm
seqDoptbrcd.maeA(trt.N, blk.N, theta, nrep, strt, sary, des0, dtype)

## Computes A-optimal or near-optimal block or row-column designs
## using array exchange algorithm
seqEoptbrcd.maeA(trt.N, blk.N, theta, nrep, strt, sary, des0, dtype)
```

**Arguments**

trt.N	integer, specifying number of treatments $v$ of initial design, $des_0$ .
blk.N	integer, specifying number of arrays $b$ of initial design, $des_0$ .
theta	numeric, representing a function of the ratio of random array variance and random error variance. It takes any value between 0 and 1, inclusive.
nrep	integer, specifying number of replications of the optimization procedure.
strt	a non-negative integer, specifying number of added treatments/conditions to the initial design.
sary	a non-negative integer, specifying number of added arrays to the initial design.
des0	matrix, a $2 \times blk.N$ or $blk.N \times 2$ initial block or row-column design. The initial design must be treatment connected and the number of treatments and arrays should also coincides with trt.N and blk.N inserted by the user, if this conditions are not satisfied, the package will stop running with an error message.
dtype	character, specifying the design type. For block designs, dtype = "blkd" and for row-column designs, dtype = "rcd".

**Details**

These functions are handled via a generic function [soptdmaeA](#). Please refer to the [soptdmaeA](#) documentation for details.

**Author(s)**

Dibaba Bayisa Gemechu, Legesse Kassa Debusho, and Linda Haines

**References**

Debusho, L. K., Gemechu, D. B. and Haines, L. (2018). Algorithmic construction of optimal block designs for two-colour cDNA microarray experiments using the linear mixed effects model. *Communications in Statistics - Simulation and Computation*, <https://doi.org/10.1080/03610918.2018.1429617>.

Gemechu D. B., Debusho L. K. and Haines L. M. (2014). A-optimal designs for two-colour cDNA microarray experiments using the linear mixed effects model. *Peer-reviewed Proceedings of the Annual Conference of the South African Statistical Association for 2014 (SASA 2014)*, Rhodes University, Grahamstown, South Africa. pp 33-40, ISBN: 978-1-86822-659-7.

**See Also**

[sopdmaeA](#)

# Index

- \* **Array exchange algorithm**
    - soptdmaeA, [6](#)
    - soptdmaeA-internal, [9](#)
  - \* **C-matrix**
    - cmatbrcd.mae, [2](#)
  - \* **Graphical layout**
    - graphsoptd.mae, [4](#)
  - \* **Information matrix**
    - cmatbrcd.mae, [2](#)
  - \* **Microarray experiment**
    - soptdmaeA, [6](#)
    - soptdmaeA-internal, [9](#)
  - \* **Sequential A-optimal block designs**
    - soptdmaeA, [6](#)
    - soptdmaeA-internal, [9](#)
  - \* **Sequential A-optimal row-column designs**
    - soptdmaeA, [6](#)
    - soptdmaeA-internal, [9](#)
  - \* **Sequential D-optimal block designs**
    - soptdmaeA, [6](#)
    - soptdmaeA-internal, [9](#)
  - \* **Sequential D-optimal row-column designs**
    - soptdmaeA, [6](#)
    - soptdmaeA-internal, [9](#)
  - \* **Sequential E-optimal block designs**
    - soptdmaeA, [6](#)
    - soptdmaeA-internal, [9](#)
  - \* **Sequential E-optimal row-column designs**
    - soptdmaeA, [6](#)
    - soptdmaeA-internal, [9](#)
  - \* **Sequential MV-optimal block designs**
    - soptdmaeA, [6](#)
    - soptdmaeA-internal, [9](#)
  - \* **Sequential MV-optimal row-column designs**
    - soptdmaeA, [6](#)
    - soptdmaeA-internal, [9](#)
- cmatbrcd.mae, [2](#)
- fixparsoptd.mae, [2](#), [3](#), [6](#), [8](#), [9](#)
- graphsoptd.mae, [4](#), [8](#)
- mmenusoptd.mae, [4](#), [5](#), [8](#), [9](#)
- print.soptdmaeA (soptdmaeA), [6](#)
- print.summary.soptdmaeA (soptdmaeA), [6](#)
- seqAoptbrcd.maeA, [7](#)
- seqAoptbrcd.maeA (soptdmaeA-internal), [9](#)
- seqDoptbrcd.maeA, [7](#)
- seqDoptbrcd.maeA (soptdmaeA-internal), [9](#)
- seqEoptbrcd.maeA, [7](#)
- seqEoptbrcd.maeA (soptdmaeA-internal), [9](#)
- seqMVoigtbrcd.maeA, [7](#)
- seqMVoigtbrcd.maeA (soptdmaeA-internal), [9](#)
- soptdmaeA, [2](#), [4-6](#), [6](#), [10](#), [11](#)
- soptdmaeA-internal, [9](#)
- summary.soptdmaeA (soptdmaeA), [6](#)