

# Package: LinkedGASP (via r-universe)

August 24, 2024

**Type** Package

**Title** Linked Emulator of a Coupled System of Simulators

**Version** 1.0

**Date** 2018-11-24

**Author** Ksenia N. Kzyurova, kseniak.ucoz.net

**Maintainer** Ksenia N. Kzyurova <ksenia.kzyurova@gmail.com>

**Depends** nloptr, spBayes

**Suggests** MASS

**Description** Prototypes for construction of a Gaussian Stochastic Process emulator (GASP) of a computer model. This is done within the objective Bayesian implementation of the GASP. The package allows for construction of a linked GASP of the composite computer model. Computational implementation follows the mathematical exposition given in publication: Ksenia N. Kzyurova, James O. Berger, Robert L. Wolpert. Coupling computer models through linking their statistical emulators. SIAM/ASA Journal on Uncertainty Quantification, 6(3): 1151-1171, (2018).<DOI:10.1137/17M1157702>.

**License** GPL (>= 3)

**NeedsCompilation** no

**Date/Publication** 2018-12-09 16:50:03 UTC

**Repository** <https://ksenia-kzyurova.r-universe.dev>

**RemoteUrl** <https://github.com/cran/LinkedGASP>

**RemoteRef** HEAD

**RemoteSha** 362d27efda12b62d2ab21eb726481833d5b8179c

## Contents

emp_GASP_plot	2
eval_GASP_RFP	4
eval_TGASP	6

eval_type1_GASP . . . . .	7
eval_type2_GASP . . . . .	8
GASP_plot . . . . .	9
link . . . . .	10
NGASPmetrics . . . . .	12
TGASPmetrics . . . . .	14
TGASP_plot . . . . .	15

<b>Index</b>	<b>17</b>
--------------	-----------

---

emp_GASP_plot	<i>Empirical linked GASP plot</i>
---------------	-----------------------------------

---

### Description

Function plots the empirical true linked emulator in case of one-dimensional input.

### Usage

```
emp_GASP_plot(em, fun, data, emul_type, exp.ql, exp.qu, labels, ylab, xlab, ylim,
col_CI_area, col_points, col_fun, col_mean, points)
```

### Arguments

em	the returned output from the function eval_type1_GASP(...) or eval_type2_GASP(...).
fun	Simulator function. Currently only one-dimensional input is supported.
data	Training data and smoothness. The same as supplied to eval_GASP_RFP(...) for construction of the GASP.
emul_type	A text string which provides description of an emulator.
exp.ql	Quantile 0.025
exp.qu	Quantile 0.975
labels	As in standard R plot.
ylab	As in standard R plot.
xlab	As in standard R plot.
ylim	As in standard R plot.
col_CI_area	Color of a credible area.
col_points	Color of the training points.
col_fun	Color of a simulator function.
col_mean	Color of the emulator of the GASP mean.
points	Default is FALSE. To plot or not the training points.

### Value

Plot

**Author(s)**

Ksenia N. Kzyurova, kseniak.ucoz.net

**Examples**

```
## Function f1 is a simulator
f1<-function(x){sin(pi*x)}
## Function f2 is a simulator
f2<-function(x){cos(5*x)}

## Function f2(f1) is a simulator of a composite model
f2f1 <- function(x){f2(f1(x))}

## One-dimensional inputs are x1
x1 <- seq(-1,1,.37)

## The following contains the list of data inputs (training) and outputs (fD) together with the
## assumed fixed smoothness of a computer model output.
data.f1 <- list(training = x1,fD = f1(x1), smooth = 1.99)

## Evaluation of GASP parameters
f1_MLEs = eval_GASP_RFP(data.f1,list(function(x){x^0},function(x){x^1}),1,FALSE)

## Evaluate the emulator
xn = seq(-1,1,.01)
GASP_type2_f1 <- eval_type2_GASP(as.matrix(xn),f1_MLEs)

par(mfrow = c(1,1))
par(mar = c(6.1, 6.1, 5.1, 2.1))
ylim = c(-1.5,1.5)
GASP_plot(GASP_type2_f1,f1,data.f1,"Type 2 GASP",ylab = " f",xlab = "x",
ylim = ylim, plot_training = TRUE)

s = GASP_type2_f1$mu
s.var = diag(GASP_type2_f1$var)

x2 = seq(-0.95,0.95,length = 6)#f1(x1)
data.f2 <- list(training = x2,fD = f2(x2), smooth = 2) # linking requires this emulator
## to have smoothness parameter equal to 2

f2_MLEs = eval_GASP_RFP(data.f2,list(function(x){x^0},function(x){x^1}),1,FALSE)

GASP_type1_f2 <- eval_type1_GASP(as.matrix(seq(-3.5,3.5,.01)),f2_MLEs)
GASP_type2_f2 <- eval_type2_GASP(as.matrix(seq(-1,1,.01)),f2_MLEs)
TGASP_f2 <- eval_TGASP(as.matrix(seq(-1,1,.01)),f2_MLEs)

ylim = c(-1.5,1.5)
# labels = c(expression(phantom(x)*phantom(x)*phantom(x)*f(x[1])),
# expression(f(x[2])*phantom(x)*phantom(x)*phantom(x)),
# expression(f(x[3])),expression(f(x[4])),
# expression(f(x[5])),expression(f(x[6])))
```

```

par(mar = c(6.1, 6.1, 5.1, 2.1))
GASP_plot(GASP_type2_f2,f2,data.f2, "Type 2 GASP",labels = x2,xlab= "z",ylab = " g",
ylim = ylim,plot_training = TRUE)

le <- link(f1_MLEs, f2_MLEs, as.matrix(xn))

## Construct an empirical emulator
n.samples = 100
em2.runs<-mat.or.vec(n.samples,length(s))
library(MASS)
for(i in 1:n.samples) {
  GASP = eval_type2_GASP(as.matrix(mvrnorm(1,s,diag(s.var))),f2_MLEs)
  em2.runs[i,] <- mvrnorm(1,GASP$mu, GASP$var)
}

## Plot the empirical GASP emulator
data.f2f1 <- list(training = x1,fD = f2f1(x1), smooth = 2)

par(mar = c(6.1, 6.1, 5.1, 2.1))
emp_GASP_plot(le$em2,f2f1,data.f2f1,"Linked",apply(em2.runs,2,quantile,probs = 0.025),
  apply(em2.runs,2,quantile,probs = 0.975),
  ylab = expression("g" ~ scriptscriptstyle(0) ~ "f"),xlab = "x, input",ylim = ylim)

```

---

eval\_GASP\_RFP

*Evaluation of parameters of a Gaussian stochastic process emulator of a computer model.*


---

## Description

This function evaluates parameters of a Gaussian stochastic process emulator of a computer model based on a few observations which are available from the simulator of a computer model.

## Usage

```
eval_GASP_RFP(data, basis, corr.cols, nugget)
```

## Arguments

data	list which consists of three objects: training input values (which may be multivariate, along several dimensions), corresponding output values of a simulator (scalar) and a vector of smoothness parameter(s) along each input direction.
basis	A set of functions in the mean of a Gaussian process. Typically assumed to be linear in one or several dimensions.
corr.cols	specifies which input directions must be included in the specification of a correlation function.
nugget	Parameter which accounts for possible small stochasticity in the output of a computer model. Default is FALSE.

**Details**

See examples which illustrate inputs specification to the function.

**Value**

Function returns a list of objects, including estimates of parameters, which is subsequently may be used for construction of a GASP approximation with the estimated parameters and the data involved.

delta	Estimates of range parameters in the correlation function.
eta	Estimates of a nugget.
sigma.sq	Estimates of variance.
data	Input parameter returned for convenience.
nugget	Input parameter returned for convenience.
basis	Input parameter returned for convenience.
corr.cols	Input parameter returned for convenience.

**Author(s)**

Ksenia N. Kzyurova, kseniak.ucoz.net.

**References**

Ksenia N. Kzyurova, James O. Berger, and Robert L. Wolpert. Coupling computer models through linking their statistical emulators. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3): 1151-1171, 2018

Gu, M., Wang, X., Berger, J. O. et al. (2018) Robust Gaussian stochastic process emulation. *The Annals of Statistics*, 46, 3038-3066.

**Examples**

```
## Function f1 is a simulator
f1<-function(x){sin(pi*x)}

## One-dimensional inputs are x1
x1 <- seq(-1,1,.37)

## data.f1 contains the list of data inputs (training) and outputs (fD) together with the assumed
## fixed smoothness of a computer model output. This corresponds to the smoothness in a product
## power exponential correlation function used for construction of the emulator.
data.f1 <- list(training = x1,fD = f1(x1), smooth = 1.99)

## Evaluation of GASP parameters
f1_MLEs = eval_GASP_RFP(data.f1,list(function(x){x^0},function(x){x^1}),1,FALSE)
```

---

 eval\_TGASP

*T-GASP emulator*


---

### Description

This function evaluates the third GASP of a computer model within objective Bayesian (OB) implementation of the GASP, resulting in T-GASP.

### Usage

```
eval_TGASP(input, GASPparams)
```

### Arguments

input	Input values (the same dimension as training input data in the next argument GASPparams)
GASPparams	The output of the function eval_GASP_RFP.

### Value

Function returns a list of three objects

x	Inputs.
mu	Mean of an emulator.
var	Covariance matrix of an emulator.

### Author(s)

Ksenia N. Kzyurova, kseniak.ucoz.net.

### Examples

```
## Function f2 is a simulator
f2<-function(x){cos(5*x)}

## One-dimensional inputs x2
x2 = seq(-0.95,0.95,length = 6)
data.f2 <- list(training = x2,fd = f2(x2), smooth = 2)

## Evaluation of GASP parameters
f2_MLEs = eval_GASP_RFP(data.f2,list(function(x){x^0},function(x){x^1}),1,FALSE)

## Evaluation of a T-GASP emulator
TGASP_f2 <- eval_TGASP(as.matrix(seq(-1,1,.01)),f2_MLEs)
```

---

eval_type1_GASP	<i>The first type of an emulator of a computer model</i>
-----------------	--

---

### Description

This function evaluates the first GASP of a computer model using maximum a posteriori estimates (MAP) of parameters of the GASP.

### Usage

```
eval_type1_GASP(input, GASPparams)
```

### Arguments

input	input values (the same dimension as training input data in the next argument GASPparams)
GASPparams	The output of the function eval_GASP_RFP.

### Details

See examples which illustrate inputs specification to the function.

### Value

Function returns a list of three objects

x	Inputs.
mu	Mean of an emulator.
var	Covariance matrix of an emulator.

### Author(s)

Ksenia N. Kyzurova, kseniak.ucoz.net.

### Examples

```
## Function f1 is a simulator
f1<-function(x){sin(pi*x)}

## One-dimensional inputs are x1
x1 <- seq(-1,1,.37)

## The following contains the list of data inputs (training) and outputs (fD) together with the
## assumed fixed smoothness of a computer model output.
data.f1 <- list(training = x1,fD = f1(x1), smooth = 1.99)

## Evaluation of GASP parameters
f1_MLEs = eval_GASP_RFP(data.f1,list(function(x){x^0},function(x){x^1}),1,FALSE)
```

```
## Evaluate the emulator
xn = seq(-1,1,.01)
GASP_type1_f1 <- eval_type1_GASP(as.matrix(xn),f1_MLEs)
```

---

eval\_type2\_GASP      *The second type of an emulator of a computer model*

---

### Description

This function evaluates the second GASP of a computer model within partial objective Bayesian (POB) implementation of the GASP.

### Usage

```
eval_type2_GASP(input, GASPparams)
```

### Arguments

input	input values (the same dimension as training input data in the next argument GASPparams)
GASPparams	The output of the function eval_GASP_RFP.

### Details

See examples which illustrate inputs specification to the function.

### Value

Function returns a list of three objects

x	Inputs.
mu	Mean of an emulator.
var	Covariance matrix of an emulator.

### Author(s)

Ksenia N. Kzyurova, kseniak.ucoz.net.

### Examples

```
## Function f2 is a simulator
f2<-function(x){cos(5*x)}

## One-dimensional inputs x2
x2 = seq(-0.95,0.95,length = 6)
data.f2 <- list(training = x2, fD = f2(x2), smooth = 2)
```

```
## Evaluation of GASP parameters
f2_MLEs = eval_GASP_RFP(data.f2,list(function(x){x^0},function(x){x^1}),1,FALSE)

## Evaluation of a second type GASP emulator
GASP_type2_f2 <- eval_type2_GASP(as.matrix(seq(-1,1,.01)),f2_MLEs)
```

---

GASP\_plot

*Plot of the GASP*


---

### Description

Function allows to plot the GASP in case of one-dimensional input.

### Usage

```
GASP_plot(em, fun, data, emul_type, labels, yax, ylab, xlab,ylim,
col_CI_area,col_points,col_fun,col_mean,plot_training = FALSE, plot_fun = TRUE)
```

### Arguments

em	the returned output from the function <code>eval_type1_GASP(...)</code> or <code>eval_type2_GASP(...)</code> .
fun	Simulator function. Currently only one-dimensional input is supported.
data	Training data and smoothness. The same as supplied to <code>eval_GASP_RFP(...)</code> for construction of the GASP.
emul_type	A text string which provides description of an emulator.
labels	As in standard R plot.
yax	As in standard R plot.
ylab	As in standard R plot.
xlab	As in standard R plot.
ylim	As in standard R plot.
col_CI_area	Color of a credible area.
col_points	Color of the training points.
col_fun	Color of a simulator function.
col_mean	Color of the emulator of the GASP mean.
plot_training	(Not) to plot the training points. Default is FALSE.
plot_fun	(Not) to plot the simulator function. Default is TRUE.

### Value

Plot

**Note**

The function requires further development to be automated for visualization along a single dimension out of multiple dimensions and along two dimensions out of multiple dimensions.

**Author(s)**

Ksenia N. Kzyurova, kseniak.ucoz.net

**Examples**

```
## Function f1 is a simulator
f1<-function(x){sin(pi*x)}

## One-dimensional inputs are x1
x1 <- seq(-1,1,.37)

## The following contains the list of data inputs (training) and outputs (fD) together with the
## assumed fixed smoothness of a computer model output.
data.f1 <- list(training = x1,fD = f1(x1), smooth = 1.99)

## Evaluation of GASP parameters
f1_MLEs = eval_GASP_RFP(data.f1,list(function(x){x^0},function(x){x^1}),1,FALSE)

## Evaluate the emulator
xn = seq(-1,1,.01)
GASP_type1_f1 <- eval_type1_GASP(as.matrix(xn),f1_MLEs)

## Plot the emulator
par(mfrow = c(1,1))
par(mar = c(6.1, 6.1, 5.1, 2.1))
ylim = c(-1.5,1.5)
GASP_plot(GASP_type1_f1,fun = f1,data = data.f1,"",ylim = ylim, plot_training = TRUE)
```

---

[link](#)

*Linking two emulators*

---

**Description**

Function constructs a linked GASP emulator of a composite computer model  $f_2(f_1)$ .

**Usage**

```
link(f1_MLEs, f2_MLEs, test_input)
```

**Arguments**

f1_MLEs	Parameters of the emulator of a simulator f1.
f2_MLEs	Parameters of the emulator of a simulator f2.
test_input	Testing inputs.

**Details**

See examples which illustrate inputs specification to the function.

**Value**

Four types of the linked GASP.

em1	Type 1 emulator, which uses MAP estimates of parameters.
em2	Type 2 emulator within partial objective Bayesian (POB) implementation.
emT	T-GASP emulator within objective Bayesian (OB) implementation.
em3	Approximated T-GASP emulator with the Gaussian distribution.

**Author(s)**

Ksenia N. Kzyurova, kseniak.ucoz.net

**References**

Ksenia N. Kzyurova, James O. Berger, and Robert L. Wolpert. Coupling computer models through linking their statistical emulators. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3): 1151-1171, 2018

**Examples**

```
## Function f1 is a simulator
f1<-function(x){sin(pi*x)}
## Function f2 is a simulator
f2<-function(x){cos(5*x)}

## Function f2(f1) is a simulator of a composite model
f2f1 <- function(x){f2(f1(x))}

## One-dimensional inputs are x1
x1 <- seq(-1,1,.37)

## The following contains the list of data inputs (training) and outputs (fD) together with
## the assumed fixed smoothness of a computer model output.
data.f1 <- list(training = x1,fD = f1(x1), smooth = 1.99)

## Evaluation of GASP parameters
f1_MLEs = eval_GASP_RFP(data.f1,list(function(x){x^0},function(x){x^1}),1,FALSE)

## Evaluate the emulator
xn = seq(-1,1,.01)
GASP_type2_f1 <- eval_type2_GASP(as.matrix(xn),f1_MLEs)

par(mfrow = c(1,1))
par(mar = c(6.1, 6.1, 5.1, 2.1))
ylim = c(-1.5,1.5)
GASP_plot(GASP_type2_f1,f1,data.f1,"Type 2 GASP",ylab = " f",xlab = "x",
```

```

ylim = ylim, plot_training = TRUE)

s = GASP_type2_f1$mu
s.var = diag(GASP_type2_f1$var)

x2 = seq(-0.95,0.95,length = 6)#f1(x1)
data.f2 <- list(training = x2,fD = f2(x2), smooth = 2) # linking requires this emulator
# to have smoothness parameter equal to 2

f2_MLEs = eval_GASP_RFP(data.f2,list(function(x){x^0},function(x){x^1}),1,FALSE)

GASP_type1_f2 <- eval_type1_GASP(as.matrix(seq(-3.5,3.5,.01)),f2_MLEs)
GASP_type2_f2 <- eval_type2_GASP(as.matrix(seq(-1,1,.01)),f2_MLEs)
TGASP_f2 <- eval_TGASP(as.matrix(seq(-1,1,.01)),f2_MLEs)

ylim = c(-1.5,1.5)
# labels = c(expression(phantom(x)*phantom(x)*phantom(x)*f(x[1])),
# expression(f(x[2])*phantom(x)*phantom(x)*phantom(x)),
# expression(f(x[3])),expression(f(x[4])),
# expression(f(x[5])),expression(f(x[6])))

par(mar = c(6.1, 6.1, 5.1, 2.1))
GASP_plot(GASP_type2_f2,f2,data.f2, "Type 2 GASP",labels = x2,xlab= "z",ylab = " g",
ylim = ylim,plot_training = TRUE)

le <- link(f1_MLEs, f2_MLEs, as.matrix(xn))

## Plot second type of the linked GASP
data.f2f1 <- list(training = x1,fD = f2f1(x1), smooth = 2)

par(mar = c(6.1, 6.1, 5.1, 2.1))
GASP_plot(le$em2,f2f1,data.f2f1,"Linked",labels = x1,
ylab = expression("g" ~ scriptscriptstyle(0) ~ "f"),xlab = "x",ylim = ylim)

```

---

 NGASPmetrics

*GASP performance assessment measures*


---

## Description

Evaluates frequentist performance of the GASP.

## Usage

```
NGASPmetrics(GASP, true_output, ref_output)
```

## Arguments

GASP	GASP emulator.
true_output	Output from the simulator.
ref_output	Heuristic emulator output.

**Value**

List of performance measures.

RMSPE_base	Root mean square predictive error with respect to the heuristic emulator output.
RMSPE	Root mean square predictive error for the emulator output
ratio	ratio of RMSPE_base to RMSPE. Ratio = RMSPE_base/RMSPE
CI	95% central credible intervals
emp_cov	95% empirical coverage within the CI
length_CIs	Average length of 95% central credible intervals

**Author(s)**

Ksenia N. Kzyurova, ksenia.ucoz.net

**References**

Ksenia N. Kzyurova, James O. Berger, and Robert L. Wolpert. Coupling computer models through linking their statistical emulators. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3): 1151-1171, 2018

**Examples**

```
## Function f1 is a simulator
f1<-function(x){sin(pi*x)}

## One-dimensional inputs are x1
x1 <- seq(-1,1,.37)

## The following contains the list of data inputs (training) and outputs (fD) together with
## the assumed fixed smoothness of a computer model output.
data.f1 <- list(training = x1,fD = f1(x1), smooth = 1.99)

## Evaluation of GASP parameters
f1_MLEs = eval_GASP_RFP(data.f1,list(function(x){x^0},function(x){x^1}),1,FALSE)

## Evaluate the emulator
xn = seq(-1,1,.01)
GASP_type2_f1 <- eval_type2_GASP(as.matrix(xn),f1_MLEs)

## Plot the emulator
par(mar = c(6.1, 6.1, 5.1, 2.1))
GASP_plot(GASP_type2_f1,data = data.f1,emul_type = "",ylim = ylim, plot_training = TRUE)

## Measure performance of an emulator
NGASPmetrics(GASP_type2_f1,f1(xn),mean(f1(xn)))
```

---

 TGASPmetrics

*Performance measurement of a T-GASP*


---

**Description**

Evaluates frequentist performance of a T-GASP.

**Usage**

```
TGASPmetrics(TGASP, true_output, ref_output)
```

**Arguments**

TGASP	TGASP emulator (in the paper this is done within an objective Bayesian implementation - OB emulator.)
true_output	Output from the simulator.
ref_output	Heuristic emulator output.

**Details**

See examples which illustrate the use of the function.

**Value**

List of performance measures.

RMSPE_base	Root mean square predictive error with respect to the heuristic emulator output.
RMSPE	Root mean square predictive error for the emulator output
ratio	ratio of RMSPE_base to RMSPE. Ratio = RMSPE_base/RMSPE
CI	95% central credible intervals
emp_cov	95% empirical coverage within the CIs
length_CIs	Average length of 95% central credible intervals

**Author(s)**

Ksenia N. Kzyurova, ksenia.ucoz.net

**References**

Ksenia N. Kzyurova, James O. Berger, and Robert L. Wolpert. Coupling computer models through linking their statistical emulators. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3): 1151-1171, 2018

**Examples**

```

## Function f1 is a simulator
f1<-function(x){sin(pi*x)}

## One-dimensional inputs are x1
x1 <- seq(-1,1,.37)

## The following contains the list of data inputs (training) and outputs (fD) together with
## the assumed fixed smoothness of a computer model output.
data.f1 <- list(training = x1,fD = f1(x1), smooth = 1.99)

## Evaluation of GASP parameters
f1_MLEs = eval_GASP_RFP(data.f1,list(function(x){x^0},function(x){x^1}),1,FALSE)

## Evaluate the emulator
xn = seq(-1,1,.01)
TGASP_f1 <- eval_TGASP(as.matrix(xn),f1_MLEs)

## Plot the emulator
par(mfrow = c(1,1))
par(mar = c(6.1, 6.1, 5.1, 2.1))
ylim = c(-1.5,1.5)
TGASP_plot(TGASP_f1,f1,data.f1,ylim = ylim)

## Measure the performance of the emulator
TGASPmetrics(TGASP_f1,f1(xn),mean(f1(xn)))

```

---

TGASP\_plot

*T-GASP plot*


---

**Description**

Function allows to plot the TGASP in case of one-dimensional input. Black-and-white version.

**Usage**

```
TGASP_plot(tem, fun, data, labels, ylim, points)
```

**Arguments**

tem	TGasP emulator.
fun	Simulator function.
data	Training data and smoothness. The same as supplied to eval_GASP_RFP(...) for construction of a GASP.
labels	As in standard R plot.
ylim	As in standard R plot.
points	(Not) to plot the training points.

**Details**

See examples.

**Value**

Plot

**Note**

The function requires further development to be automated for visualization along a single dimension out of multiple dimensions and along two dimensions out of multiple dimensions.

This function needs to be automated to allow for fast visualization of a single emulator (with no comparison to the actual simulator function), etc.

**Author(s)**

Ksenia N. Kzyurova, kseniak.ucoz.net

**Examples**

```
## Function f1 is a simulator
f1<-function(x){sin(pi*x)}

## One-dimensional inputs are x1
x1 <- seq(-1,1,.37)

## The following contains the list of data inputs (training) and outputs (fD) together with
## the assumed fixed smoothness of a computer model output.
data.f1 <- list(training = x1,fD = f1(x1), smooth = 1.99)

## Evaluation of GASP parameters
f1_MLEs = eval_GASP_RFP(data.f1,list(function(x){x^0},function(x){x^1}),1,FALSE)

## Evaluate the emulator
xn = seq(-1,1,.01)
TGASP_f1 <- eval_TGASP(as.matrix(xn),f1_MLEs)

## Plot the emulator
par(mfrow = c(1,1))
par(mar = c(6.1, 6.1, 5.1, 2.1))
ylim = c(-1.5,1.5)
TGASP_plot(TGASP_f1,f1,data.f1,ylim = ylim)
```

# Index

## \* **models**

- eval\_GASP\_RFP, [4](#)
- eval\_TGASP, [6](#)
- eval\_type1\_GASP, [7](#)
- eval\_type2\_GASP, [8](#)
- link, [10](#)

## \* **print**

- emp\_GASP\_plot, [2](#)
- GASP\_plot, [9](#)
- TGASP\_plot, [15](#)

## \* **utilities**

- NGASPmetrics, [12](#)
- TGASPmetrics, [14](#)

- emp\_GASP\_plot, [2](#)
- eval\_GASP\_RFP, [4](#)
- eval\_TGASP, [6](#)
- eval\_type1\_GASP, [7](#)
- eval\_type2\_GASP, [8](#)

- GASP\_plot, [9](#)

- link, [10](#)

- NGASPmetrics, [12](#)

- TGASP\_plot, [15](#)
- TGASPmetrics, [14](#)