

# Package: `optim.functions` (via `r-universe`)

October 31, 2024

**Title** Standard Benchmark Optimization Functions

**Version** 0.1

**Description** A set of standard benchmark optimization functions for R and a common interface to sample them.

**Depends** R (>= 3.0)

**License** MIT + file LICENSE

**LazyData** true

**Suggests** testthat, covr

**Imports** lhs, randtoolbox, stats, stringr

**Collate** 'optim.functions.R' 'vectorize.R' 'sampling.R'  
'single.objective.R' 'functions.R'

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Thomas Torsney-Weir [aut, cre]

**Maintainer** Thomas Torsney-Weir <torsneyt@gmail.com>

**Date/Publication** 2017-03-30 11:28:25 UTC

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

**RemoteUrl** <https://github.com/cran/optim.functions>

**RemoteRef** HEAD

**RemoteSha** efc63eb325c270502401726af486f69163824452

## Contents

<code>get_info</code>	2
<code>optim.functions</code>	2
<code>sample.func</code>	2
<code>samplingf</code>	3

<b>Index</b>	<b>4</b>
--------------	----------

---

get_info	<i>Lookup information about a function by name</i>
----------	--

---

**Description**

Lookup information about a function by name

**Usage**

```
get_info(fun.name)
```

**Arguments**

fun.name	Function name to lookup
----------	-------------------------

---

optim.functions	<i>optim.functions: A collection of standard optimization functions along with a standard interface to call and sample those functions.</i>
-----------------	---

---

**Description**

optim.functions: A collection of standard optimization functions along with a standard interface to call and sample those functions.

**Sampling functions**

The sampling functions all need to have a standard interface. They all must take 2 parameters: n, the number of samples to generate and k, the number of dimensions to sample. They all return a matrix-like object containing n rows of k columns of samples.

The main interface to this library is through the [sample.func](#) function.

---

sample.func	<i>Unified function sampling interface.</i>
-------------	---

---

**Description**

Provides a unified interface to sampling all functions. One just needs to specify the name of the function to sample, the number of samples, the dimensionality, and the sampling method and the library will take care of the rest.

**Usage**

```
sample.func(FUN.name, n, k, method = "lh.sample", ...)
```

**Arguments**

FUN.name	The name of the function to sample.
n	The number of sample points.
k	The number of dimensions to sample.
method	The sampling method.
...	Additional arguments passed to FUN.name.

**Value**

A data frame consisting of n rows with columns x1..xk and y1..yz where z is the number of outputs of the function.

---

samplingf	<i>The sampling functions take (n, k) where n is the number of samples and k is the number of dimensions. The sampling functions should return samples in a 0-1 hypercube. sampling.func &lt;- function(n, k)</i>
-----------	---

---

**Description**

The sampling functions take (n, k) where n is the number of samples and k is the number of dimensions. The sampling functions should return samples in a 0-1 hypercube. sampling.func <- function(n, k)

**Usage**

```
lh.sample(n, k, ...)
cartesian.sample(n, k, ...)
random.sample(n, k, ...)
hexagonal.sample(n, k, ...)
torus.sample(n, k, ...)
sobol.sample(n, k, ...)
halton.sample(n, k, ...)
```

**Arguments**

n	The number of samples.
k	The number of dimensions.
...	Additional arguments for the sampling (e.g. seed).

# Index

`cartesian.sample (samplingf)`, 3

`get_info`, 2

`halton.sample (samplingf)`, 3

`hexagonal.sample (samplingf)`, 3

`lh.sample (samplingf)`, 3

`optim.functions`, 2

`optim.functions-package`  
(`optim.functions`), 2

`random.sample (samplingf)`, 3

`sample.func`, 2, 2

`samplingf`, 3

`sobol.sample (samplingf)`, 3

`torus.sample (samplingf)`, 3