

An R-based genetic algorithm

Egon Willighagen 

Published November 19, 2007

Citation

Willighagen, E. (2007). An R-based genetic algorithm. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/hfw6p-d6p02>

Keywords

Rstats, Chemometrics

Abstract

During my PhD I wrote a simple but effective genetic algorithm package for R. Because there was a bug recently found, and there is interest in extending the functionality, I have set up a SourceForge project called *genalg*.

Copyright

Copyright © Egon Willighagen 2007. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

chem-bla-ics

During my PhD I wrote a simple but [effective genetic algorithm](#) package for R. Because there was a bug recently found, and there is interest in extending the functionality, I have set up a [SourceForge](#) project called [genalg](#).

The package provides GA support for binary and real-value chromosomes (and integer chromosomes is something that will be added soon), and allows to use custom evaluation functions. Here is some example code:

```
# optimize two values to match pi and sqrt(50)
evaluate <- function(string=c()) {
  returnVal = NA;
  if (length(string) == 2) {
    returnVal = abs(string[1]-pi) + abs(string[2]-sqrt(50));
  } else {
    stop("Expecting a chromosome of length 2!");
  }
  returnVal
}

monitor <- function(obj) {
  # plot the population
  xlim = c(obj$stringMin[1], obj$stringMax[1]);
  ylim = c(obj$stringMin[2], obj$stringMax[2]);
  plot(obj$population, xlim=xlim, ylim=ylim, xlab="pi", ylab="sqrt(50)");
}

rbga.results = rbga(c(1, 1), c(5, 10), monitorFunc=monitor,
  evalFunc=evaluate, verbose=TRUE, mutationChance=0.01)
plot(rbga.results)
plot(rbga.results, type="hist")
plot(rbga.results, type="vars")
```