mantelNSGA vignette

```
library(mantelNSGA2)
library(vegan)
```

1 Introduction

bla (Clarke and Warwick 1994). Clarke and Warwick (1994) total combinations tested by sinkr::bioEnv() `2^ncol(var.mat)-1 =` {r 2^ncol(var.mat)-1}

The following tutorial outlines the use of the ELEFAN functions available in TropFishR (`ELEFAN`, `ELEFAN_GA`, and `ELEFAN_SA`). These functions can be used to estimate growth model parameters from length-frequency (lfq) data. The ability to convert length to relative age is the initial step in length-based stock assessment, and underpins subsequent analyses (see the sinkr package for a broader overview of length-based stock assessment).

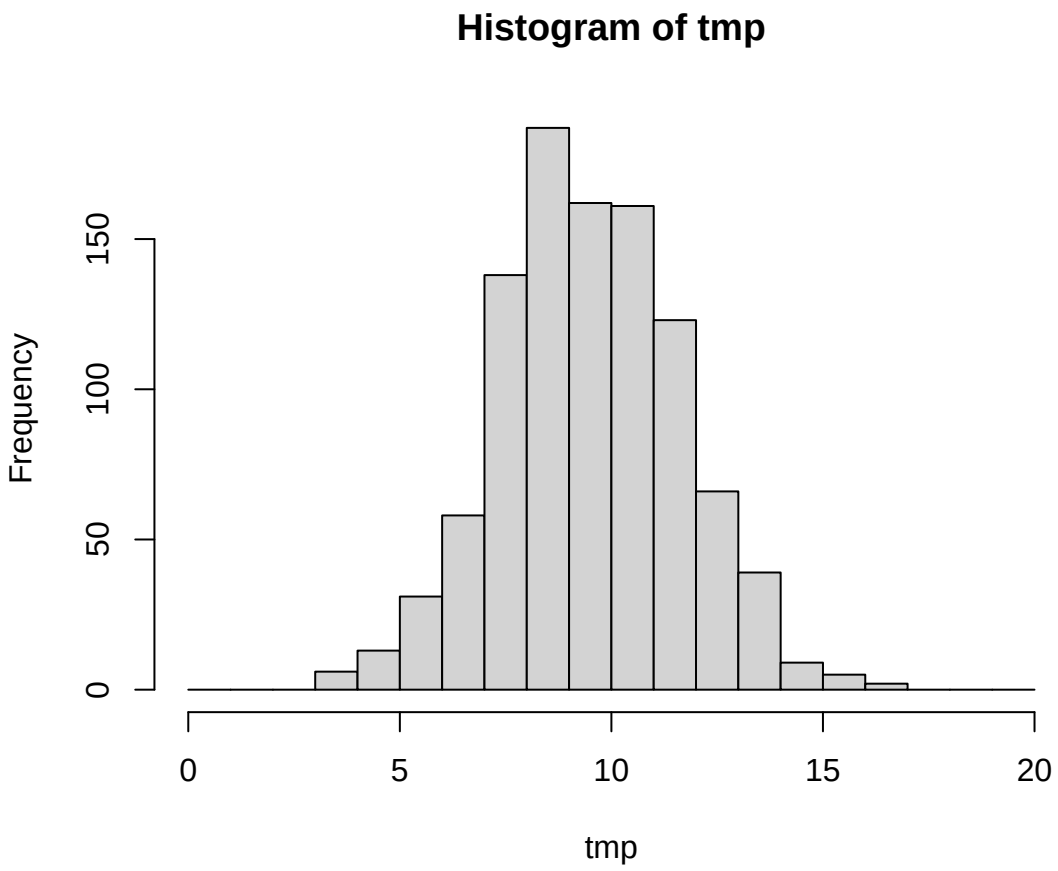
The growth model presently implemented within TropFishR is the von Bertalanffy growth function (VBGF),

$$L_t = L_\infty(1 - \exp(-K(t - t_0))),$$

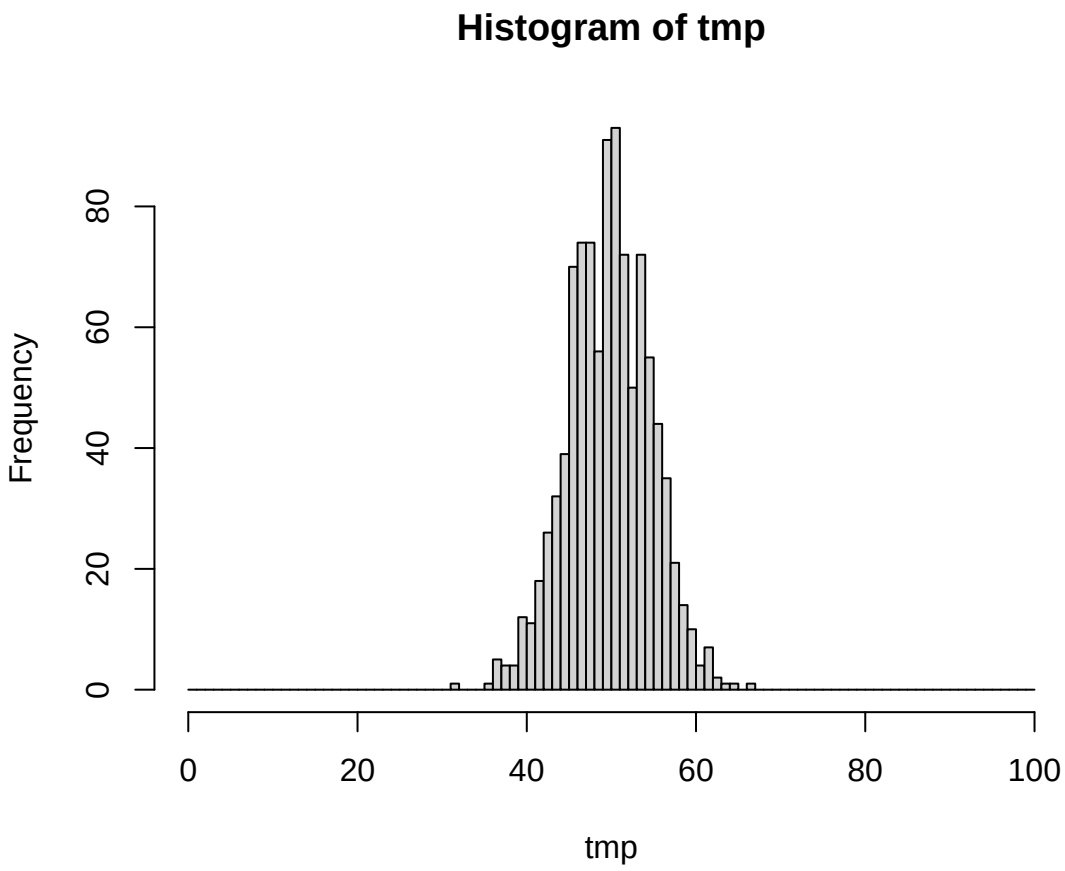
where L_t is length-at-age t, L_∞ is asymptotic length, K is the von Bertalanffy growth constant, and t_0 is the theoretical age when length equals zero. t_0 is usually negative, resulting in a positive length at the time of recruitment (t = 0). In TropFishR, these parameters are referred to as `Linf`, `K`, and `t0`.

The following plot shows the basic form of the VBGF under varying K settings:

```
# chance of having few or many "genes" turned on is low, so one ends up
# mainly exploring the middle
nvar <- 20
tmp <- replicate(1000, expr = sum(sample(x = c(0,1), nvar, replace = TRUE)))
hist(tmp, breaks = seq(0, nvar))
```



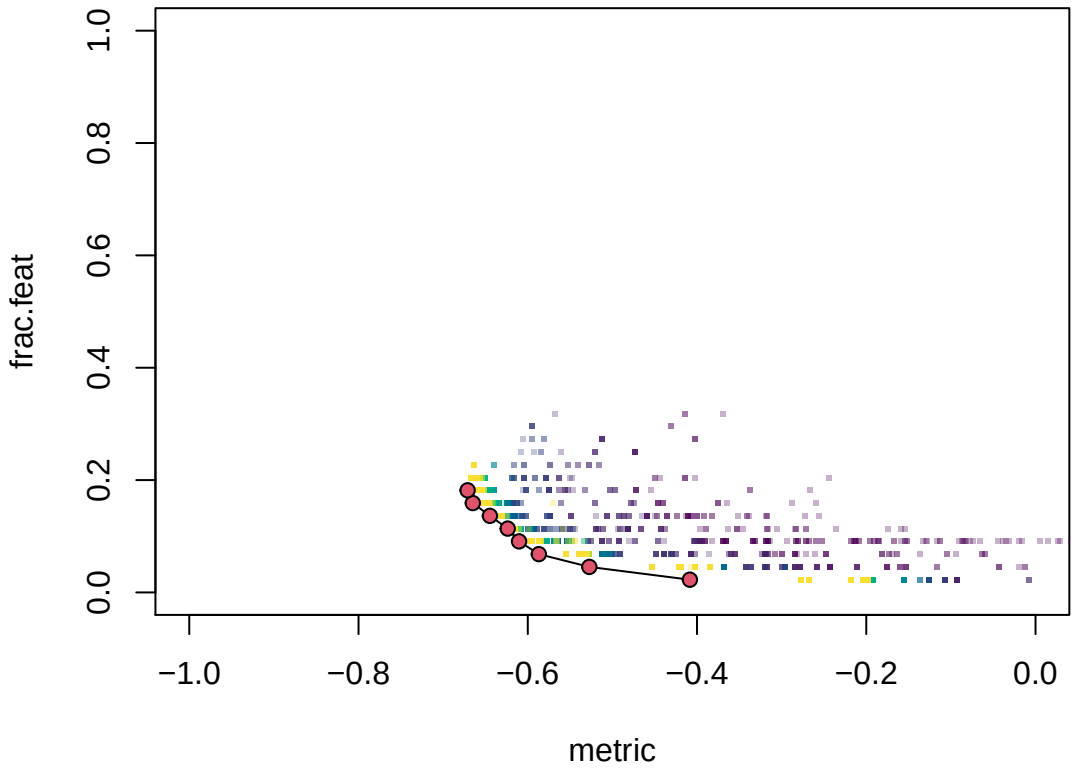

```
# more extreme when number of genes is high
nvar <- 100
tmp <- replicate(1000, expr = sum(sample(x = c(0,1), nvar, replace = TRUE)))
hist(tmp, breaks = seq(0, nvar))
```



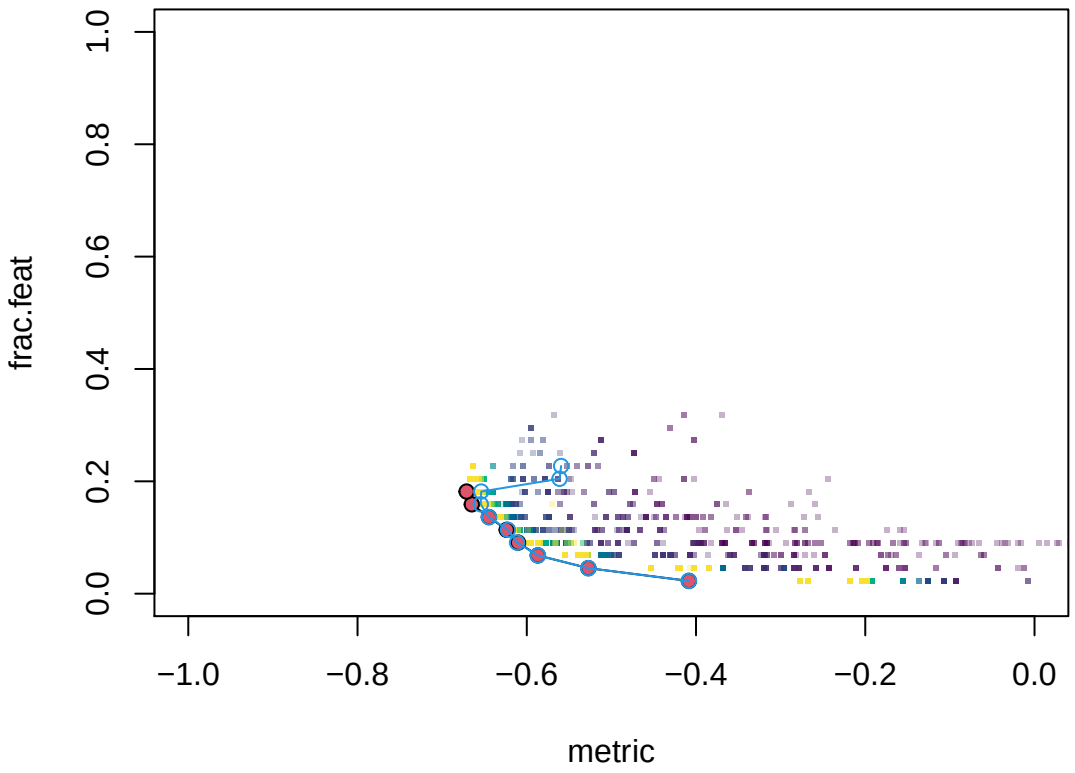
1.1 EnvBio example

```
data("varechem")
data("varespec")

# biological community variables that best correlate with environmental data
fix.mat = varechem
var.mat = wisconsin(varespec)
fix.dist.method = "euclidean"
var.dist.method = "bray"
scale.fix = TRUE
scale.var = FALSE
p.feat <- 0.1

# mantelNSGA2
set.seed(1111)
fitGA <- mantelNSGA2(
  fix.mat = fix.mat, var.mat = var.mat,
  fix.dist.method = fix.dist.method, var.dist.method = var.dist.method,
  scale.fix = scale.fix, scale.var = scale.var, p.feat = p.feat,
  pop.size = 50, max.iter = 100, stop.criterion = 50,
```

```
  mutation.rate = 0.2, crossover.rate = 0.8, verbose = FALSE
)
# plot(fitGA$generation.fitness)
# fitGA$pareto.solution
plot(fitGA, parFrontT = "o")
```



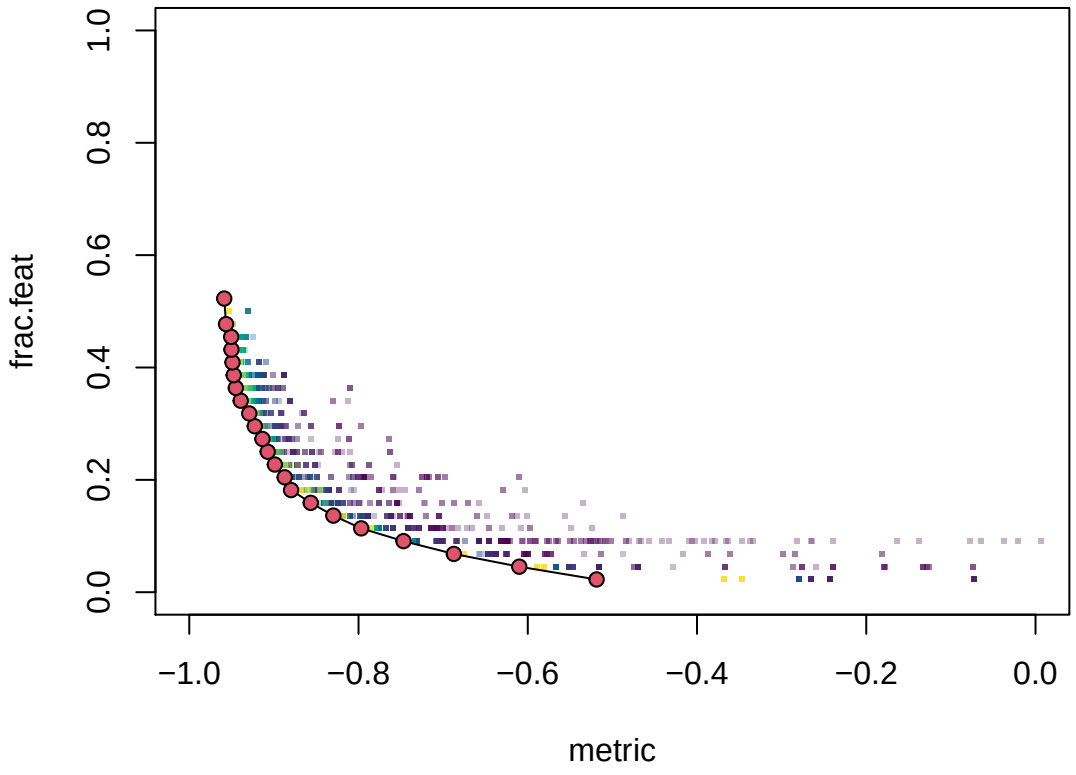

```
# Compare to bvStep
set.seed(1111)
fitBV <- bvStep(
  fix.mat = fix.mat, var.mat = var.mat,
  fix.dist.method = fix.dist.method, var.dist.method = var.dist.method,
  scale.fix = scale.fix, scale.var = scale.var,
  num.restarts = 50,
  random.selection = TRUE,
  prop.selected.var = 0.4,
  verbose = FALSE
)
# fitBV$order.by.best # mantelGA not looking in simpler solutions enough
# fitBV$order.by.i.comb
plot(fitGA, parFrontT = "o")
points(-fitBV$order.by.i.comb$rho, fitBV$order.by.i.comb$n.var/ncol(var.mat), pch = 1, col = 4, t = "o")
```



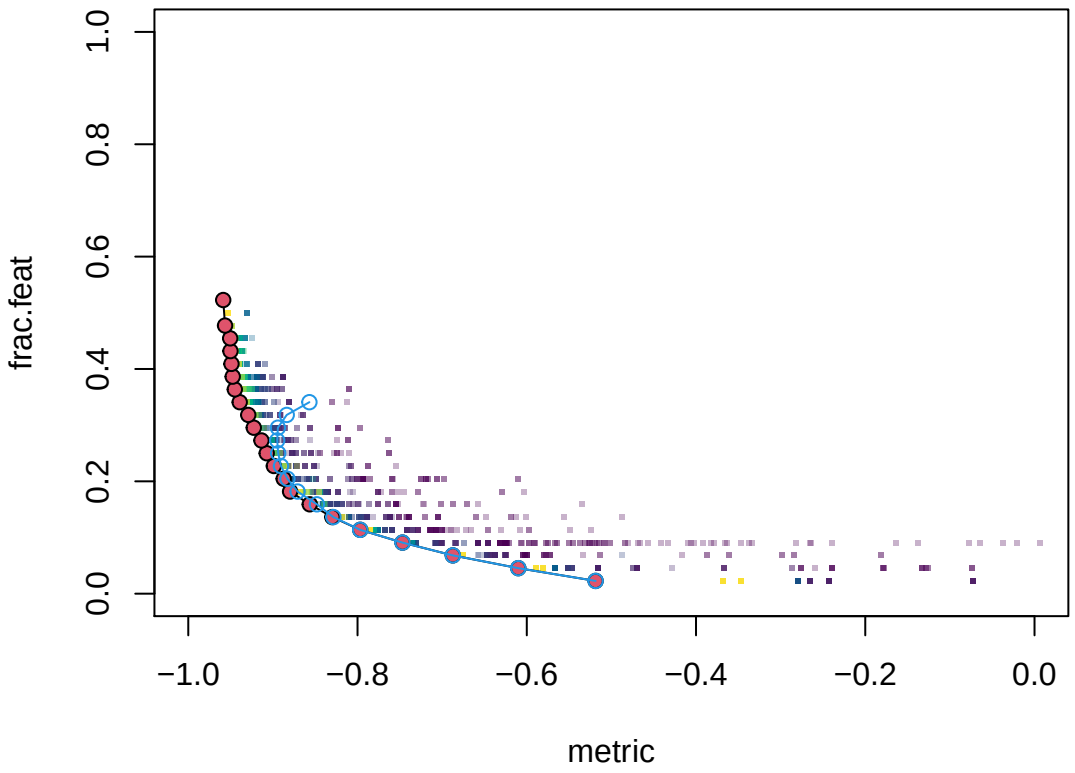
1.2 BioBio example

```
# biological community variables that best correlate with the
# overall biological community
fix.mat = wisconsin(varespec)
var.mat = wisconsin(varespec)
fix.dist.method = "bray"
var.dist.method = "bray"
scale.fix = FALSE
scale.var = FALSE
p.feat <- 0.1

# mantelNSGA2
set.seed(1111)
fitGA2 <- mantelNSGA2(
  fix.mat = fix.mat, var.mat = var.mat,
  fix.dist.method = fix.dist.method, var.dist.method = var.dist.method,
  scale.fix = scale.fix, scale.var = scale.var, p.feat = p.feat,
  pop.size = 50, max.iter = 100, stop.criterion = 50,
  mutation.rate = 0.2, crossover.rate = 0.8, verbose = FALSE
)
```

```
# plot(fitGA2$generation.fitness)
# fitGA2$pareto.solution
plot(fitGA2, parFrontT = "o")
```




```
# Compare to bvStep
set.seed(1111)
fitBV2 <- bvStep(
  fix.mat = fix.mat, var.mat = var.mat,
  fix.dist.method = fix.dist.method, var.dist.method = var.dist.method,
  scale.fix = scale.fix, scale.var = scale.var,
  num.restarts = 50,
  random.selection = TRUE,
  prop.selected.var = 0.3,
  var.always.include = c(15,23,26),
  verbose = FALSE
)
# fitBV2$order.by.best # mantelGA not looking in simpler solutions enough
# fitBV2$order.by.i.comb
plot(fitGA2, parFrontT = "o")
points(-fitBV2$order.by.i.comb$rho, fitBV2$order.by.i.comb$n.var/ncol(var.mat), pch = 1, col = 4, t = "
```



References

Clarke, KR, and RM Warwick. 1994. "An Approach to Statistical Analysis and Interpretation." *Change in Marine Communities* 2: 117–43.