Trabalho Final

Gustavo Vieira, Marcus Vinicius, Thais Matos, Rafael Castro

04 de maio de 2016

1. Introdução

Um pesquisador está interessado em investigar o efeito de diferentes operadores de recombinação no desempenho do algoritmo de evolução diferencial para uma dada classe de problemas. Dessa forma, o fator de impacto é o operador utilizado e o efeito a ser observado é o desempenho.

Considerando quatro operadores de recombinação para o experimento (problema de quatro tratamentos), torna-se necessário utilizar a análise de variância $ANOVA$ para comparação das médias em um único problema de teste.

Após a análise do experimento, responderemos perguntas como: *Há alguma diferença no desempenho médio do algoritmo quando equipado com estes diferentes operadores, para o problema de teste utilizado? Caso haja, qual o melhor operador em termos de desempenho médio, e qual a magnitude das diferenças encontradas? Há algum operador que deva ser recomendado em relação aos demais?*

2. Formulação das hipóteses de teste

Através da modelagem de efeitos fixos podemos estimar os efeitos dos operadores no algoritmo de evolução diferencial. Os efeitos dos operadores τ_i, que representam desvios da média global μ, são variáveis aleatórias. Como os conhecimentos acerca dos tratamentos particulares investigados não são relativamente importantes, testaremos as hipóteses sobre a variabilidade de τ_i e tentaremos estimar essa variabilidade.

Estamos interessados em testar a igualdade das médias dos quatro tratamentos, μ_1, μ_2, μ_3, μ_4. Logo, as hipóteses são:

- $H_0 : \tau_i = 0$
- $H_1 : \tau_i \neq 0$ (para pelo menos um i)

Se a hipótese nula for verdadeira, significa que a mudança dos operadores de recombinação não tem efeito no desempenho médio do algoritmo evolutivo.

3. Cálculo do tamanho amostral

O problema nos fornece os dados necessários para o cálculo do tamanho da amostra. É desejado um poder de 0.85 ($1 - \beta = 0.85; \beta = 0.15$) para uma mínima diferença de importância prática $d^* = \delta^*/\sigma = 0.25$ e um nível de significância $\alpha = 0.05$.

```r
alpha <- 0.05;
beta <- 0.15;
d <- 0.25;
a <- 4;

#Calculando o tamanho amostral (suficiente)
library(pwr);
n <- round(pwr.anova.test(k = a,f = d, sig.level = alpha, power = 1-beta)$n)
```

Dessa forma, encontramos um tamanho amostral suficiente para os parâmetros desejados de n = 50.

4. Coleta e tabulação dos dados

Realizaremos a comparação dos seguintes operadores de recombinação:

```
library(ExpDE);
selpars <- list(name = "selection_standard");
stopcrit <- list(names = "stop_maxeval", maxevals = 60000, maxiter = 1000);
probpars <- list(name = "sphere", xmin = -seq(1,20), xmax = 20 + 5 * seq(5, 24));

# Grupo C (Operadores para comparação)
recpars1 <- list(name = "recombination_blxAlphaBeta", alpha = 0, beta = 0)
mutpars1 <- list(name = "mutation_rand", f = 4)
popsize1 <- 200
recpars2 <- list(name = "recombination_linear")
mutpars2 <- list(name = "mutation_rand", f = 1.5)
popsize2 <- 250
recpars3 <- list(name = "recombination_mmax", lambda = 0.25)
mutpars3 <- list(name = "mutation_best", f = 4)
popsize3 <- 375
recpars4 <- list(name = "recombination_npoint", N = 17)
mutpars4 <- list(name = "mutation_rand", f = 2.2)
popsize4 <- 225
```

Coletamos n observações para cada operador de forma não sequencial, pois buscamos intercalar as observações entre os operadores com o objetivo de evitar o efeito de qualquer variável de ruído que possa influenciar o desempenho do algoritmo.

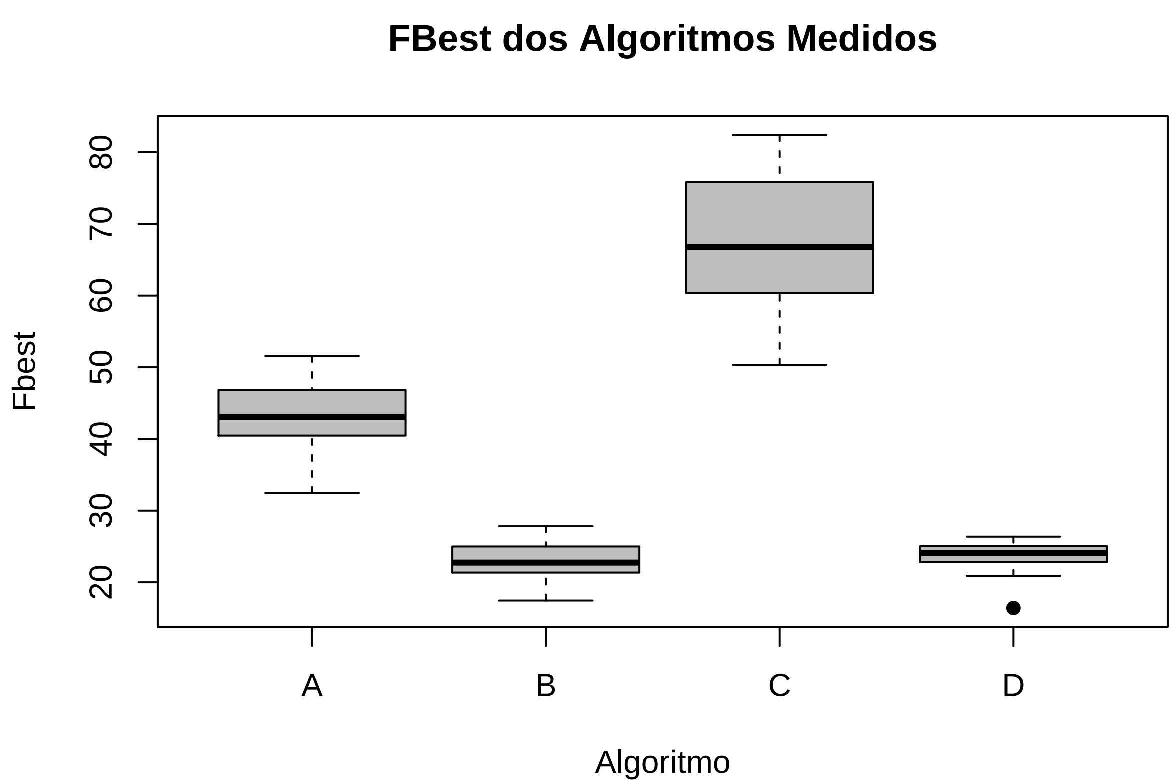
```
#Gerando n observações para cada operador
fbest1 <- c(0);
fbest2 <- c(0);
fbest3 <- c(0);
fbest4 <- c(0);
for (i in seq(1:n)){
  out1 <- ExpDE(popsize1, mutpars1, recpars1, selpars, stopcrit, probpars);
  fbest1[i] <- out1$Fbest;
  out2 <- ExpDE(popsize2, mutpars2, recpars2, selpars, stopcrit, probpars);
  fbest2[i] <- out2$Fbest;
  out3 <- ExpDE(popsize3, mutpars3, recpars3, selpars, stopcrit, probpars);
  fbest3[i] <- out3$Fbest;
  out4 <- ExpDE(popsize4, mutpars4, recpars4, selpars, stopcrit, probpars);
  fbest4[i] <- out4$Fbest;
}
algoritmo <- c(rep("A",n), rep("B",n), rep("C",n), rep("D",n));
fbest <- c(fbest1, fbest2, fbest3, fbest4);
dadosColetados <- data.frame(algoritmo, fbest);
summary(dadosColetados);
```

```
##  algoritmo     fbest
##  A:50      Min.   :16.42
##  B:50      1st Qu.:23.61
```

```
##  C:50      Median :30.14
##  D:50      Mean   :39.45
##            3rd Qu.:51.26
##            Max.   :82.40
```

A distribuição dos dados coletados está representada no gráfico abaixo, o qual mostra evidências de que há diferença no desempenho do algoritmo de acordo com o operador de recombinação utilizado.

```r
# Boxplot
boxplot(fbest~algoritmo,
        data = dadosColetados,
        xlab = "Algoritmo",
        ylab = "Fbest",
        main = "FBest dos Algoritmos Medidos",
        pch  = 16,
        col  = "gray");
```



5. Teste das hipóteses

Utilizando o teste ANOVA, obtemos o seguinte resultado:

```r
# Teste da hipótese
model <- aov(fbest~algoritmo, data = dadosColetados);
summary.aov(model);
```

```
##               Df Sum Sq Mean Sq F value Pr(>F)    
## algoritmo     3  64542   21514   781.7 <2e-16 ***
## Residuals   196   5394      28                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Analisando o teste de hipótese podemos ver que o valor p é bastante inferior ao nível de significância α. Dessa forma, podemos rejeitar a hipótese nula, em favor da alternativa, com 95% de confiança. Esse resultado indica uma diferença entre o desempenho dos operadores de recombinação.

O ANOVA não identifica quais médias são diferentes, portanto precisamos utilizar métodos de comparações múltiplas para este fim. Basicamente, faremos uma série de t-testes para saber como cada um dor operadores difere dos demais.

All vs. All - Tukey's Honest Significant Difference

```r
# All vs. all
library(multcomp);
```

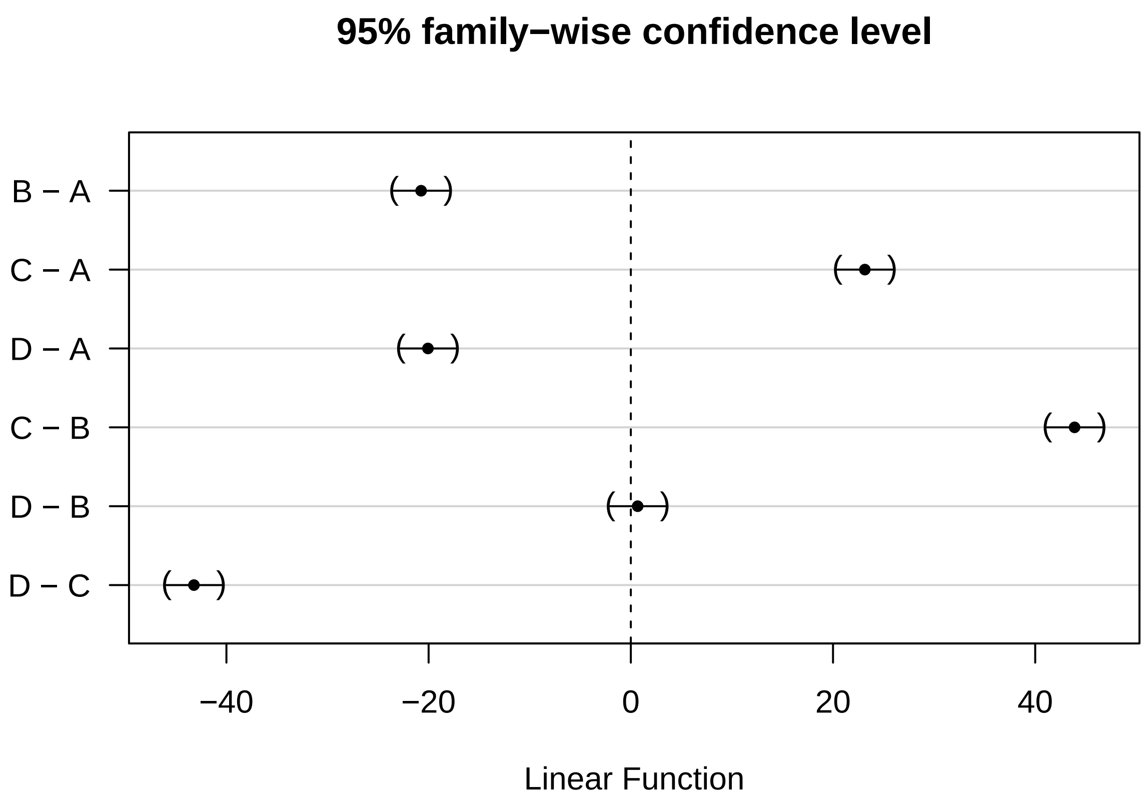
```
## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data

## Loading required package: MASS

## 
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':
## 
##     geyser
```

```r
tukey <- glht(model, linfct = mcp(algoritmo = "Tukey"));
tukey_CI <- confint(tukey, level = 0.95);
summary(tukey_CI);
```

```
## 
##   Simultaneous Tests for General Linear Hypotheses
## 
## Multiple Comparisons of Means: Tukey Contrasts
## 
## 
## Fit: aov(formula = fbest ~ algoritmo, data = dadosColetados)
## 
## Linear Hypotheses:
##            Estimate Std. Error t value Pr(>|t|)    
## B - A == 0 -20.7437     1.0492 -19.771   <1e-05 ***
## C - A == 0  23.1485     1.0492  22.063   <1e-05 ***
## D - A == 0 -20.0669     1.0492 -19.126   <1e-05 ***
## C - B == 0  43.8922     1.0492  41.833   <1e-05 ***
```

```
## D - B == 0   0.6768     1.0492   0.645    0.917
## D - C == 0 -43.2154     1.0492 -41.188   <1e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```r
plot(tukey_CI);
```



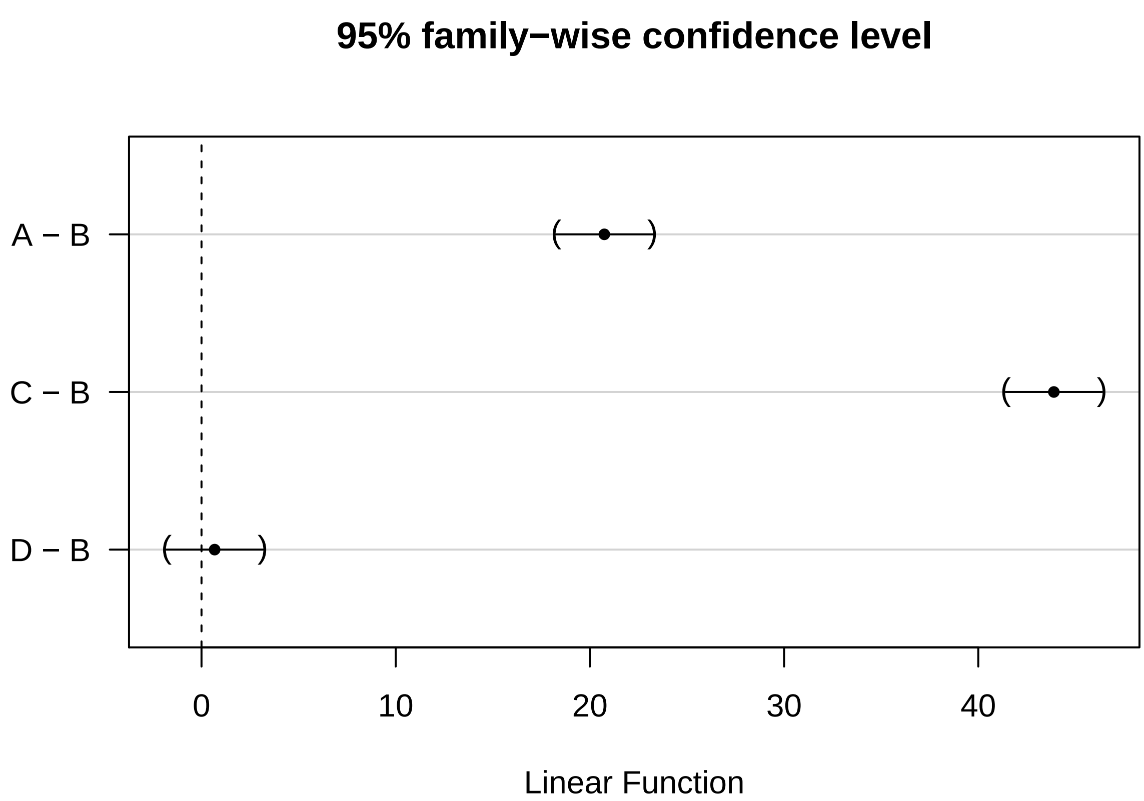
All vs. One - Dunnett's test

```r
# All vs. one
dadosColetados$algoritmo <- relevel(dadosColetados$algoritmo, ref = "B");
model2 <- aov(fbest~algoritmo, data = dadosColetados);
dunnet <- glht(model2, linfct = mcp(algoritmo = "Dunnet"));
dunnet_CI <- confint(dunnet, level = 0.95);
summary(dunnet_CI);
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Dunnett Contrasts
##
##
## Fit: aov(formula = fbest ~ algoritmo, data = dadosColetados)
##
```

```
## Linear Hypotheses:
##            Estimate Std. Error t value Pr(>|t|)
## A - B == 0  20.7437     1.0492  19.771   <1e-05 ***
## C - B == 0  43.8922     1.0492  41.833   <1e-05 ***
## D - B == 0   0.6768     1.0492   0.645    0.856
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

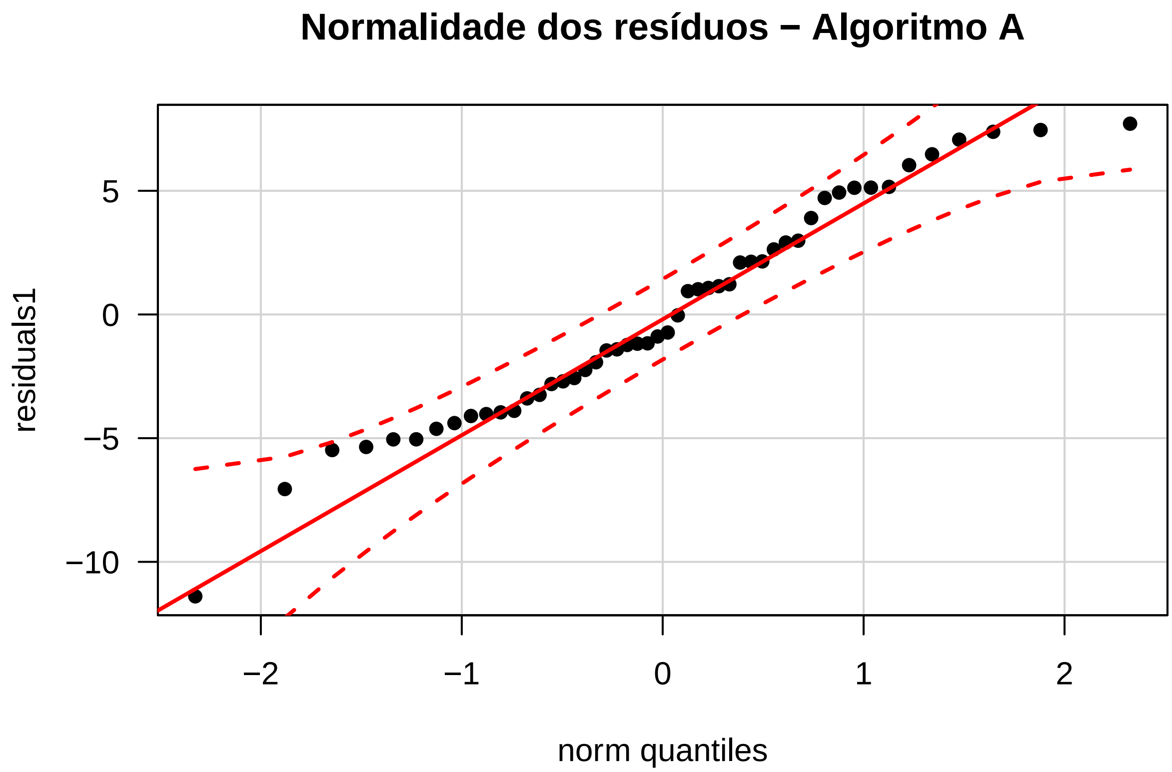
```r
plot(dunnet_CI);
```



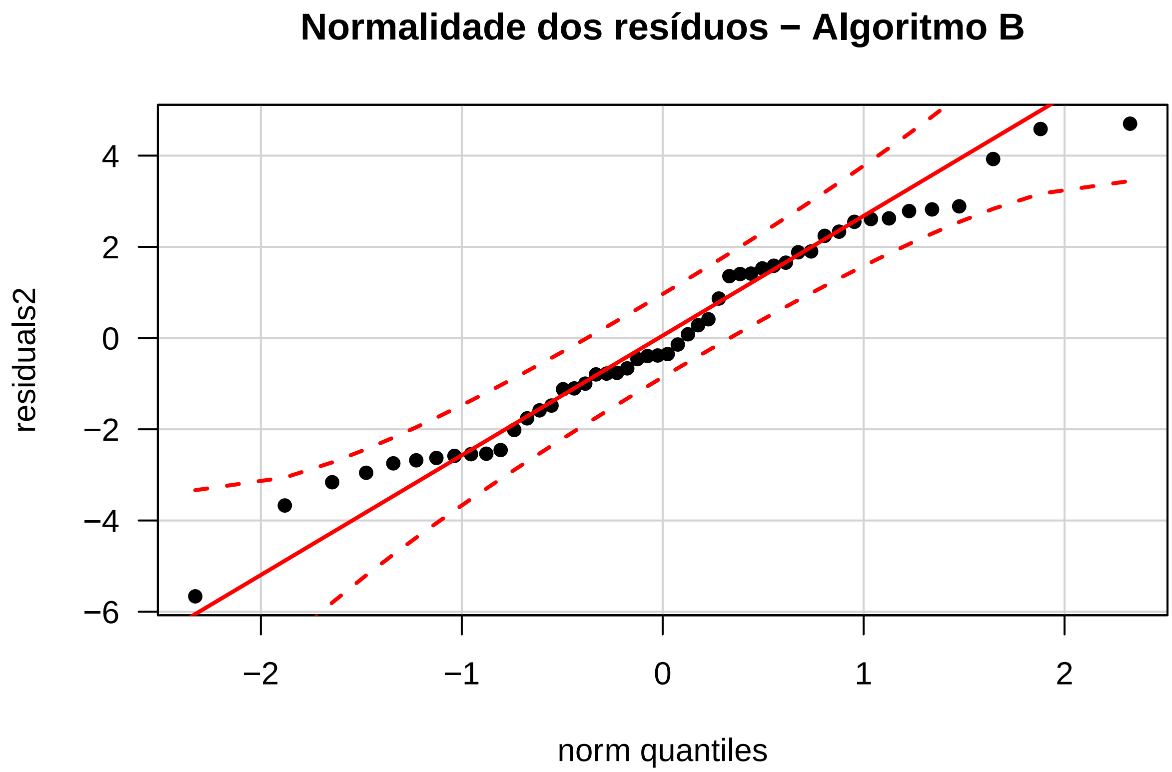
6. Verificação das premissas dos testes

As premissas dos teste: normalidade, homogeneidade e independência podem ser validadas nos seguintes gráficos:

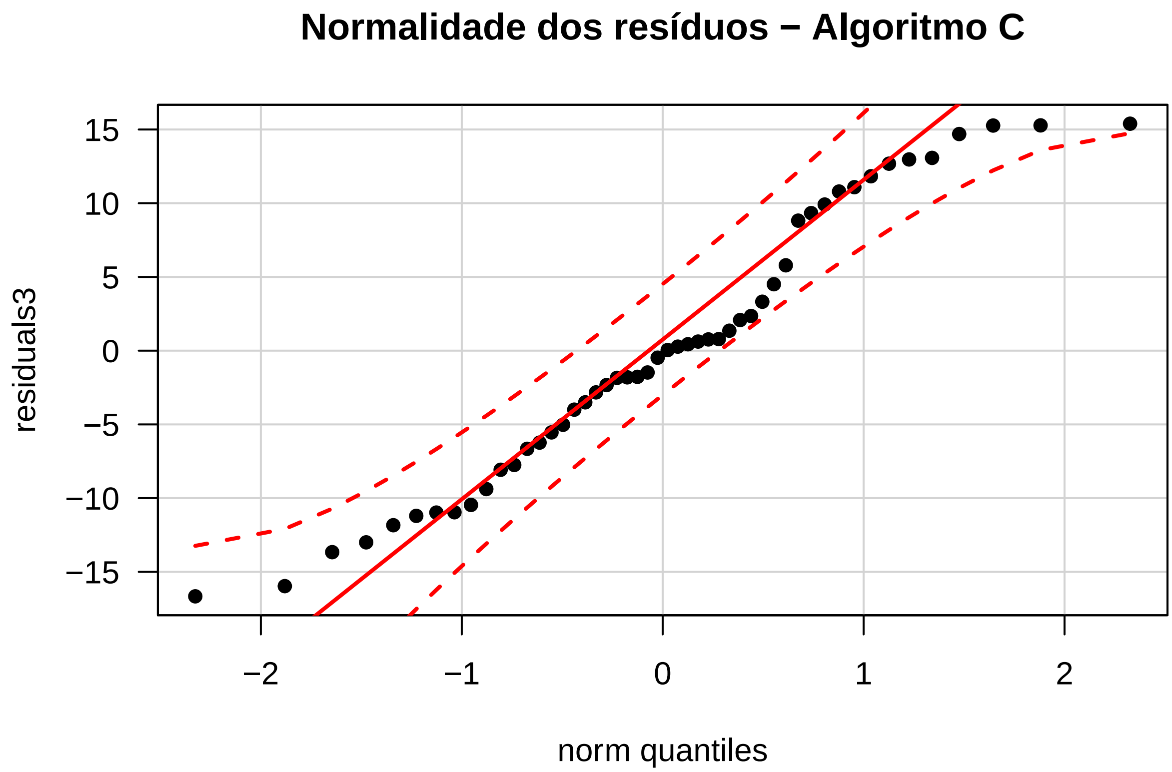
```r
# Verificando normalidade
library(car);
residuals1 <- fbest1 - mean(fbest1);
residuals2 <- fbest2 - mean(fbest2);
residuals3 <- fbest3 - mean(fbest3);
residuals4 <- fbest4 - mean(fbest4);
qqPlot(residuals1, pch=16, cex=1.0, las=1, main="Normalidade dos resíduos - Algoritmo A");
```



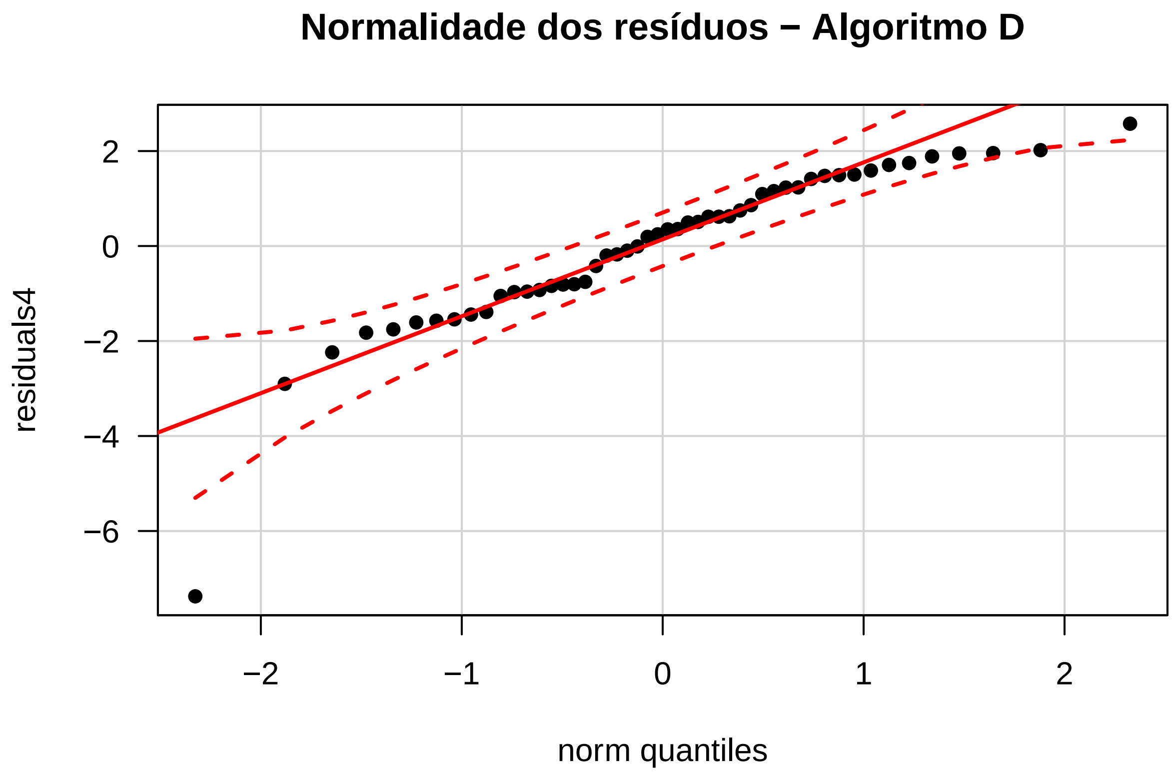

```
qqPlot(residuals2, pch=16, cex=1.0, las=1, main="Normalidade dos resíduos - Algoritmo B");
```




```
qqPlot(residuals3, pch=16, cex=1.0, las=1, main="Normalidade dos resíduos - Algoritmo C");
```




```r
qqPlot(residuals4, pch=16, cex=1.0, las=1, main="Normalidade dos resíduos - Algoritmo D");
```




```r
# Verificando homogeneidade
fligner.test(fbest~algoritmo, data = dadosColetados)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  fbest by algoritmo
## Fligner-Killeen:med chi-squared = 70.412, df = 3, p-value =
## 3.483e-15
```

```r
# Verificando independência
durbinWatsonTest(model)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      -0.1374133      2.274323   0.088
##  Alternative hypothesis: rho != 0
```

- A normalidade dos dados pode ser assumida, visto que ocorreram poucase pequenas violações do limite, de acordo com o gráfico de simulação.
- A homogeneidade pode ser comprovada dado que o valor p encontrado pelo teste é inferior ao nível de significância desejado.
- O teste de independência não é muito conclusivo, porém a independência dos dados é garantida na aleatoriedade da coleta de dados inicial.

7. Conclusão

Conforme os resultados do experimento, podemos concluir que existe uma diferença entre o desempenho médio dos quatro operadores de recombinação. Em termos de desempenho médio, o operador B seria o mais recomendado dentre os operadores testados.

A partir dos resultados do teste de Tukey (all vs. all), pode-se ver as diferenças de magnitude existentes entre os operadores de recombinação, exceto se B (recombination_linear) e D (recombination_npoint) forem comparados entre si, quando a diferença é menos perceptível. Pelo teste de Dunnet(all vs. one), colocando o operador B como referência (controle), podemos concluir que este é mais eficiente que A (recombination_blxAlphaBeta) e C (recombination_mmax), mas possui eficiência bem semelhante ao algoritmo D. Assim, pelos nossos testes, não podemos inferir uma diferença significativa entre os algoritmos B e D, mas podemos concluir que ambos são mais eficientes que A e C. Para melhorar o experimento, poderia ser considerado um tamanho de amostras n limite, ao invés de suficiente.

8. Referências

[1] https://github.com/fcampelo/Design-and-Analysis-of-Experiments

[2] Estatística Aplicada e Probabilidade para Engenheiros (4a edição) - Montgomery

[3] A Estatística Básica e Sua Prática (6a edição) - David S. Moore, William I. Nortz, Michael A. Fligner

[4] http://support.minitab.com/pt-br/minitab/17/topic-library/basic-statistics-and-graphs/hypothesis-tests/tests-of-means/why-use-paired-t/

[5] http://www.statmethods.net/stats/power.html

[6] http://docslide.com.br/documents/anova-com-r.html