CFAssay: Statistics of the Colony Formation Assay

Herbert Braselmann

October 30, 2017

Research Unit Radiation Cytogenetics, Group Integrative Biology
Helmholtz Zentrum München
braselm@helmholtz-muenchen.de

Contents

1 Overview 1
  1.1 The models ................................. 2
  1.2 Remark on intercepts c and plating efficiencies .......................... 2

2 Example: Linear-quadratic model for cell survival curves 3
  2.1 Data input and double-check .......................... 3
  2.2 Calculation of cell survival curves ......................... 4
  2.3 Comparison of the two cell survival curves ................... 6

3 Example: ANOVA for experimental two-way design 9
  3.1 Data input and double-check ................................ 9
  3.2 ANOVA model ..................................... 10

1 Overview

The functions in this package provide tools for statistical analysis along with the colony formation assay (CFA) [Franken et al., 2006]. These allow fitting of the linear-quadratic (LQ) model for ionizing radiation dependent cell survival curves and ANOVA (analysis of variance) for experimental two-way designs with one dose level of a treatment factor. Maximum-Likelihood (ML) based methods are preferred because, theoretically, parameter estimations of ML for Poisson distributed data come with smaller variances compared to other methods. However, for the sake of comparability also simple least squares (LS) based methods can be optionally used. The ML based methods employ the R-function glm for generalised linear modelling, while LS based methods use the R-function lm. The functions provided by CFAssay intend to simplify and specialize the general use of that R-functions. The underlying distribution for the ML based methods is Poisson [Frome and Beauchamp, 1968] and modelling is performed using the link function "log", i.e. cell survival curves are logarithmically linear with "linear" parameters $\alpha$ (per dose unit) and $\beta$ (per squared-dose unit). In the two-way ANOVA model...
the dependency of treatment factors is considered as logarithmically additive. Output summaries are adapted from the \texttt{glm} or the \texttt{lm} functions and use the terminology of quantities in the CFA. An accompanying paper is published in Radiation Oncology (Braselmann et al., 2015).

1.1 The models

Cell survival $S$ as a function of radiation dose using the so called LQ-model is given by

$$S = S(D) = e^{c + \alpha D + \beta D^2}$$

(D is the radiation dose and named dose in the code. $D^2$ is named dose2 but has not to be set by the user. Coefficients $\alpha$ and $\beta$ appear as "alpha" and "beta" in the print function of CFAssay. The intercepts $c = \log(S0)$ represent the logarithmic plating efficiencies, i.e. surviving fractions of untreated cells in replicated experiments. They correspond to variable Exp which is treated as a factor. Due to the positive formulation (1), parameters $c$, $\alpha$ and $\beta$ take negative values in the fit results. The logarithmically additive 2-way ANOVA with two levels for each of the two factors can mathematically be formulated as

$$S = e^{c + Ax_1 + Bx_2 + D x_1 x_2}$$

or as nested parametrization

$$S = e^{c + Ax_1 + B_0 x_2 + (B_1 - B_0) x_1 x_2}$$

There $x_1$, $x_2$ take level values 0 or 1 for each of the two factors $A$ or $B$, where for e.g. 0 means untreated and 1 means treated. $D$ is the factor for potential interaction and is coded as $A:B$ in R. In the second, nested parametrization $B_0$ is the effect of treatment in control cells and $B_1$ the treatment effect after after applying $A$. The interaction $D$ is then the difference between $B_1$ and $B_0$. In the function \texttt{cfa.2way} of CFAssay we use per default the nested version, coded as $A/B$ (see "An Introduction to R", Chapter 11 Statistical models in R). $c$ represents again the logarithmic plating efficiencies for each experiment.

1.2 Remark on intercepts $c$ and plating efficiencies

By default CFAssay processes plating efficiencies (PE) from replicate experiments as model parameters, i.e. as intercepts $c$, controlled by setting the option parameter \texttt{PEmethod} to "fit". From a statistical point of view, this appears to be preferable, because likewise the colony counts from treated cells the data from untreated cells are random observations. The shape parameters $\alpha$ and $\beta$ can be viewed as averaged over the experiments. The conventional normalization method (\texttt{PEmethod} = "fix") that is applied on experimental replicate PE measurements is mathematically equivalent to forcing the mean curve of the data to go through the intercept of each particular replicate curve. In the case of somewhat increased variation between the shape parameters of different experiments, conventional normalization results in a larger dispersion parameter in combination with the ML method. In this case, fitted intercepts $c$ (named PE in the printed output) may deviate from the measured PEs, however, they result in better overall statistics. On the shape parameter values itself, it has little influence. Thus, it is more a matter of scale, what can be visualized in the diagnostic plots using the function \texttt{plotExp}. 

2
2 Example: Linear-quadratic model for cell survival curves

The data file contains data sets on irradiation experiments of the two cell lines CAL 33 [Bauer et al., 2010] and OKF6T/TERT1. The data set on CAL 33 comprises 4 repeated experiments, and that of OKF6T/TERT1 8 experiments. The workflow shown here is divided in the following three steps: 1. data input and double-check, 2. calculation of cell survival curves for each of the two cell lines separately, 3. comparison test of the curves for the two experiments

2.1 Data input and double-check

First we load the library and read the data into the memory. The data file, expl1_cellsurvcurves.txt, is an unformatted tab-delimited text file and contains the data from the two irradiation experiments.

```r
> library(CFAssay)
> datatab <- read.table(system.file("doc", "expl1_cellsurvcurves.txt", package="CFAssay"), header=TRUE, sep="\t")
```

The data file contains columns with header names `cline`, `Exp`, `dose`, `ncells` and `ncolonies`. `cline` distinguishes the curves in the data frame. `Exp` discriminates replicates within each curve. The `dose` column relates to the applied radiation dose, `ncells` to the number of cells seeded and `ncolonies` to the number of counted colonies. The last four names are required by the CFAssay function. The name of the first column (here "cline") is arbitrary and additional columns for the distinction of curves may be contained in the data frame, e.g. pre-treatment.

```r
> names(datatab)
[1] "cline"   "Exp"     "dose"     "ncells"   "ncolonies"
```

```r
> head(datatab, 3)  # First 3 lines

  cline Exp dose ncells ncolonies
1 cal33 e1 0 900 182
2 cal33 e1 1 1800 284
3 cal33 e1 2 3000 323
```

It is advisable to double-check the number of rows, columns and frequencies or cross frequencies of the data with the R functions `dim` and `table`. The output is not shown here.

```r
> dim(datatab)
> table(datatab$cline)
> table(datatab$cline, datatab$Exp)
> table(datatab$cline, datatab$dose)
```
2.2 Calculation of cell survival curves

With the CFAssay function `cellsurvLQfit` we calculate the parameters of a linear-quadratic cell survival curve along with quality or goodness-of-fit statistics. For that purpose the data frame `datatab` has to be filtered for data relating to one curve only, because the variable `cline` is ignored by the fit function. With the function `print` the result is shown in three tables, the coefficient table, the observed and fitted plating efficiencies table and a table for analysis of the residual sum of weighted squares in the replicate experiments. In the coefficient table the "t value" column represents values of the t-test against zero of the estimated coefficients ("Estimate") and column "Pr(>|t|)" contains the corresponding p-values. By default the maximum-likelihood method is used and plating efficiencies are fitted as intercepts. Other options can be chosen in the argument list of `cellsurvLQfit` as explained in the help document.

```r
> X <- subset(datatab, cline=="okf6TERT1")
> dim(X)
[1] 48  5
> fit <- cellsurvLQfit(X)
method = ml
PEmethod = fit
dose   dose2
-0.51937898 -0.02102614
Use 'print' to see detailed results
> print(fit)
*** Coefficients of LQ-model for cell survival ***
method = ml
PEmethod = fit
Logarithmic plating efficiencies PE fitted as intercepts
see remark in the manual, 1.2

|          | Estimate | Std. Error | t value | Pr(>|t|)   |
|----------|----------|------------|---------|------------|
| PEe1     | -1.606688| 0.1061229  | -15.1399| 1.112338e-17|
| PEe2     | -1.693346| 0.1090082  | -15.5341| 4.781145e-18|
| PEe3     | -2.010377| 0.1228551  | -16.3638| 8.506670e-19|
| PEe4     | -1.869228| 0.1165115  | -16.0432| 1.644185e-18|
| PEe5     | -2.052405| 0.1096987  | -19.1836| 3.828727e-21|
| PEe6     | -2.219654| 0.1333357  | -16.6471| 4.789596e-19|
| PEe7     | -2.434634| 0.1455642  | -16.7255| 4.091214e-19|
| PEe8     | -2.109080| 0.1258530  | -16.7582| 3.830883e-19|

Shape parameters alpha and beta

|          | Estimate | Std. Error | t value | Pr(>|t|)   |
|----------|----------|------------|---------|------------|
| alpha    | -0.51937898| 0.05889260  | -8.819088| 9.943331e-11|
| beta     | -0.02102614| 0.00978985  | -2.147749| 3.817362e-02|
```
Observed and fitted plating efficiencies (%):

<table>
<thead>
<tr>
<th>Experiment</th>
<th>PE</th>
<th>PEfitted</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEE1</td>
<td>e1</td>
<td>19.0</td>
</tr>
<tr>
<td>PEE2</td>
<td>e2</td>
<td>17.3</td>
</tr>
<tr>
<td>PEE3</td>
<td>e3</td>
<td>10.0</td>
</tr>
<tr>
<td>PEE4</td>
<td>e4</td>
<td>15.0</td>
</tr>
<tr>
<td>PEE5</td>
<td>e5</td>
<td>9.2</td>
</tr>
<tr>
<td>PEE6</td>
<td>e6</td>
<td>17.0</td>
</tr>
<tr>
<td>PEE7</td>
<td>e7</td>
<td>14.3</td>
</tr>
<tr>
<td>PEE8</td>
<td>e8</td>
<td>11.5</td>
</tr>
</tbody>
</table>

Residual Deviance: 167.8964
Total residual sum of weighted squares rsswTot: 164.8109
Residual Degrees of Freedom: 38
Dispersion parameter: 4.337128

Fraction rssw of rsswTot per Experiment

<table>
<thead>
<tr>
<th>Experiment</th>
<th>rssw perCent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 e1</td>
<td>4.84 2.9</td>
</tr>
<tr>
<td>2 e2</td>
<td>14.91 9.0</td>
</tr>
<tr>
<td>3 e3</td>
<td>9.01 5.5</td>
</tr>
<tr>
<td>4 e4</td>
<td>4.51 2.7</td>
</tr>
<tr>
<td>5 e5</td>
<td>34.77 21.1</td>
</tr>
<tr>
<td>6 e6</td>
<td>52.22 31.7</td>
</tr>
<tr>
<td>7 e7</td>
<td>41.09 24.9</td>
</tr>
<tr>
<td>8 e8</td>
<td>3.46 2.1</td>
</tr>
</tbody>
</table>

If the dispersion parameter is high, experimental data may have to be removed or replaced. An appropriate cut-off depends on experience and may vary between different labs. For the example data, where plating efficiencies were fitted, we recommend a cut-off of 9.0, which corresponds to 3 Poisson standard deviations. With fixed plating efficiencies a cut-off of 12.0 may be appropriate. For the pure Poisson distribution the expected value of the dispersion parameter is 1.0.

A plot of the mean curve is generated with `plot`. Values of plotted mean survival fractions and error bars are shown with functions `sfpmean` and `pes`.

```r
> plot(fit)
> S0 <- pes(X)$S0
> names(S0) <- pes(X)$Exp
> sfpmean(X, S0)
```

<table>
<thead>
<tr>
<th></th>
<th>dose_0</th>
<th>dose_1</th>
<th>dose_2</th>
<th>dose_3</th>
<th>dose_4</th>
<th>dose_6</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF</td>
<td>0.99275669 0.57563672 0.33274302 0.17935702 0.08517535 0.020704208</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stdev</td>
<td>0.03731008 0.04618487 0.03341709 0.02160074 0.01340983 0.004103207</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
With `plotExp` diagnostic plots for each experiment are generated. Here we plot them into a pdf-file.

```r
> pdf("okf6TERT1_experimental_plots.pdf")
> plotExp(fit)
> dev.off()
```

The procedure is repeated for the other cell line, "cal33". The result is not shown here.

```r
> X <- subset(datatab, cline=="cal33")
> dim(X)
> fit <- cellsurvLQfit(X)
> print(fit)
> plot(fit)
> plotExp(fit)
```

### 2.3 Comparison of the two cell survival curves

The two linear-quadratic cell survival curves are compared with the CFAssay function `cellsurvLQdiff`. The required argument `curvevar` is set to "cline", which is the name of the column in `datatab` which distinguishes the two curves to be compared. The function uses an ANOVA test for comparison of two model fits. In "Model 1", which corresponds to the
Null-hypothesis, the dose coefficient (alpha) and the dose-squared coefficient (beta) are independent of the two curves. In "Model 2" the coefficients are different. Detailed results are printed with function print.

```r
> fitcomp <- cellsurvLQdiff(datatab, curvevar="cline")

*** Overall comparison for two linear-quadratic cell survival curves ***
Compared curves: cal33 okf6TERT1
method: ml
PEmethod: fit

Test used: F-test

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Pr(&gt;F)</td>
<td></td>
</tr>
<tr>
<td>7.3509</td>
<td>0.001463</td>
<td>**</td>
</tr>
</tbody>
</table>

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Use 'print' to see detailed results

> print(fitcomp)

Overall comparison test for coefficients alpha and beta of LQ-models
====================================================================
method = ml
PEmethod = fit

12 PEs fitted as intercepts. To look at, use simple R print function.
Null hypothesis (Model 1): one set of shape parameters alpha and beta for all data

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|---------|
| alpha    | -0.39893900 | 0.047865098 | -8.334653 | 1.687495e-11 |
| beta     | -0.03434671 | 0.007828292 | -4.387510 | 4.906364e-05 |

Goodness-of-fit values
Residual Deviance: 317.2604
Total sum of squared weighted residuals rsswTot: 319.7611
Residual Degrees of Freedom: 58
Dispersion parameter: 5.513123

Alternative hypothesis (Model 2): two sets of shape parameters alpha and beta

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|---------|
| alpha:curvescal33 | -0.26831918 | 0.062588637 | -4.287027 | 7.206618e-05 |
| alpha:curvesokf6TERT1 | -0.51937898 | 0.059669207 | -8.704305 | 5.439622e-12 |
| beta:curvescal33 | -0.04892355 | 0.010058934 | -4.863691 | 9.739225e-06 |
| beta:curvesokf6TERT1 | -0.02102614 | 0.009918948 | -2.119795 | 3.846873e-02 |

Goodness-of-fit values
Residual Deviance: 251.804
Total sum of squared weighted residuals rsswTot: 249.3271
Residual Degrees of Freedom: 56
Dispersion parameter: 4.452269

Analysis of Variance Table and F-test
Model 2 versus Model 1

<table>
<thead>
<tr>
<th>Resid. Df</th>
<th>Resid. Dev</th>
<th>Df</th>
<th>Deviance</th>
<th>F</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>58</td>
<td></td>
<td>317.26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>56</td>
<td>2</td>
<td>251.80</td>
<td>65.456</td>
<td>0.001463 **</td>
</tr>
</tbody>
</table>

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

The two curves are plotted with different colors in one plot, using the option add=TRUE. Further annotations can be added by the user to the plot with the R functions legend and text as needed.

```r
> plot(cellsurvLQfit(subset(datatab, cline="okf6TERT1")), col=1)
```

method = ml
PEmethod = fit
dose  dose2
-0.51937898 -0.02102614
Use 'print' to see detailed results

```r
> plot(cellsurvLQfit(subset(datatab, cline="cal33")), col=2, add=TRUE)
```

method = ml
PEmethod = fit
dose  dose2
-0.26831918 -0.04892355
Use 'print' to see detailed results

```r
> legend(0, 0.02, c("OKF6/TERT1", "CAL 33"), text.col=1:2)
```
3 Example: ANOVA for experimental two-way design

In this section a two-way ANOVA is demonstrated for the human oesophageal adenocarcinoma cell line OE19 which was treated with the chemotherapeutic drug cisplatin/5-FU before and after siRNA transfection. The results were previously published in (Aichler et al., 2013). Of special interest was a potential interaction, i.e. chemosensitisation between the siRNA transfection and the drug effect.

3.1 Data input and double-check

First the data are read into memory.

```r
> datatab <- read.table(system.file("doc", "exp2_2waycfa.txt", package="CFAssay"),
  + header=TRUE, sep="\t")
```

The data file contains columns with header names Exp, x5fuCis, siRNA, ncells and ncolonies. x5fuCis and siRNA stand for the drug and biological treatment, respectively. They take values 0 for control or 1 for treated. The names of the other columns are as in the cell survival curve example.

```r
> names(datatab)
```
> head(datatab, 3) # First 3 lines

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp</td>
<td>x5fuCis</td>
<td>siRNA</td>
<td>ncells</td>
<td>ncolonies</td>
</tr>
<tr>
<td>1 e1a</td>
<td>0</td>
<td>0</td>
<td>1000</td>
<td>750</td>
</tr>
<tr>
<td>2 e1a</td>
<td>0</td>
<td>1</td>
<td>1000</td>
<td>546</td>
</tr>
<tr>
<td>3 e1a</td>
<td>1</td>
<td>0</td>
<td>1000</td>
<td>316</td>
</tr>
</tbody>
</table>

Again, number of rows and columns and frequencies or cross frequencies of the data may be checked with R functions `dim` and `table` (output not shown).

> dim(datatab)
> table(datatab$x5fuCis)
> table(datatab$siRNA)
> table(datatab$Exp, datatab$x5fuCis)
> table(datatab$Exp, datatab$siRNA)

### 3.2 ANOVA model

Statistical analysis is performed with CFAssay function `cfa2way`, using parametrisation option "A/B" corresponding to formula (3). In the argument list A and B have to be set as shown. Maximum-likelihood method is default, but least-squares can be chosen optionally. The output shows the result of a test for interaction.

> fitcomp <- cfa2way(datatab, A="siRNA", B="x5fuCis", param="A/B")

*** Two-way ANOVA for factors A and B with interaction ***

A= siRNA, B= x5fuCis
Test for interaction: F-test

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>values</td>
<td>9.831</td>
<td>0.01202</td>
</tr>
</tbody>
</table>

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Use 'print' to see detailed results

Detailed results are shown with function `print.cfa2wayfit`. In the output A0:B1 and A1:B1 correspond to B0 and B1 in formula (3).

> print(fitcomp, labels=c(A="siRNA", B="x5fuCis"))

*** Logarithmic linear two-way ANOVA for factors A and B with interaction ***

A= siRNA, B= x5fuCis
Postscript digits for A or B: 0 inactive, 1 active

surv_percent = exp(Estimate)*100

Null hypothesis (Model 1): no interaction
| Estimate  | Std. Error | t value | Pr(>|t|) | surv_percent |
|-----------|------------|---------|----------|--------------|
| A1        | -0.4237844 | 0.06366051 | -6.656944 | 5.660071e-05 | 65.5 |
| B1        | -1.1187559 | 0.07738539 | -14.456939 | 4.980588e-08 | 32.7 |

Goodness-of-fit values
Residual Deviance: 77.99569  
Total sum of squared weighted residuals ssqwresTot: 75.87275  
Residual Degrees of Freedom: 10 
Dispersion parameter: 7.587275

Alternative hypothesis (Model 2): interaction

-parametrization: A/B-

| Estimate  | Std. Error | t value | Pr(>|t|) | surv_percent |
|-----------|------------|---------|----------|--------------|
| A1        | -0.3484218 | 0.05266374 | -6.615972 | 9.746888e-05 | 70.6 |
| A0:B1     | -0.9757699 | 0.07170380 | -13.608343 | 2.619891e-07 | 37.7 |
| A1:B1     | -1.3432352 | 0.09470322 | -14.183627 | 1.832336e-07 | 26.1 |

Goodness-of-fit values
Residual Deviance: 37.15596  
Total sum of squared weighted residuals ssqwresTot: 37.38767  
Residual Degrees of Freedom: 9 
Dispersion parameter: 4.154185

Analysis of Variance Table and F-test
Model 2 versus Model 1

<table>
<thead>
<tr>
<th>Resid. Df</th>
<th>Resid. Dev</th>
<th>Df</th>
<th>Deviance</th>
<th>F</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>9</td>
<td>77.996</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>1</td>
<td>37.156</td>
<td>40.84</td>
<td>9.831</td>
</tr>
</tbody>
</table>

---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Diagnostic plots for repeated experiments are printed to pdf.

> pdf("TwoWay_experimental_plots.pdf")
> plotExp(fitcomp, labels=c(A="siRNA", B="x5fuCis"))
> dev.off()

References


