User Manual for Reduced Piecewise EXponential Estimate (RPEXE) Software

Gang Han and Michael J. Schell
March 2014

GH email: ghan@sph.tamhsc.edu; gang.han@yale.edu

MJS email: Michael.Schell@moffitt.org

1 General Information

RPEXE (Reduced Piecewise EXponential Estimate) is a set of MATLAB programs or R programs for estimating the survival probability with a reduced piecewise exponential approach proposed in Han et al. (2014). This set of programs:

1. Imposes a pre-specified order restriction on the failure rate, which can be decreasing, increasing, monotone, decreasing then increasing, or increasing then decreasing;

2. Computes the p-values (Han et al. 2012) at all event times to compare adjacent total time on tests;

3. Reports the largest (the most non-significant) p-value at each iteration.

After imposing the order restriction the number of candidate changepoints can be reduced significantly. Using the list of p-values and the critical p-value ($\alpha^s tar$ in Han et al. 2014), one can decide the number and location of changepoints.

This user manual introduces the RPEXE program with an example from Han et al. (2014). Section 2 sketches the model and implementation. Section 3 describes inputs/outputs and files used by the program. Section 4 illustrates RPEXE program with the Non-small-cell lung cancer example in Han, et al. (2014).

2 Model and implementation

2.1 Survival Estimates Given the Changepoints in the Failure Rate

We let $[Z]$ denote the distribution of a generic random variable Z and z denote a realization from $[Z]$. We let "log" denote the natural logarithm. Suppose $X_e \in (0, \infty)$ is a random variable with the exponential distribution having parameter λ. Suppose $X_{pe} \in (0, \infty)$ is a random variable with a piecewise exponential distribution divided by change-points $t_{(1)} < t_{(2)} < \cdots < t_{(p)}$ at which the failure rate varies.

The p.d.f. $f(X_e|\lambda)$, survival function $S(X_e|\lambda)$, and hazard function $h(X_e|\lambda)$ are $f(X_e|\lambda) = \exp\{-X_e/\lambda\}/\lambda$, $S(X_e|\lambda) = \exp\{-X_e/\lambda\}$, and $h(X_e|\lambda) = f(X_e|\lambda)/S(X_e|\lambda) = 1/\lambda$. Further, the mean and median of X_e are $E(X_e|\lambda) = \lambda$ and $M(X_e|\lambda) = \lambda\log(2)$. Define $t_{(0)} = 0$ and $t_{(p+1)} = \infty$. Let $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{p+1})$ denote the vector of the unknown model parameters, where $1/\lambda_j$ is the instantaneous failure rate in $(t_{(j-1)}, t_j]$ for $j = 1, \ldots, p+1$. Define $E_0 = 1$ and

$$E_j = \prod_{k=1}^{j} \exp\left\{-\frac{t_{(k)} - t_{(k-1)}}{\lambda_k}\right\} = E_{j-1} \exp\left\{-\frac{t_{(j)} - t_{(j-1)}}{\lambda_j}\right\}.$$

Note that $(t_{(j)}, E_j)$ is the point where the j^{th} piece of the $S(X_{pe}|\boldsymbol{\lambda})$ starts in the plot of $S(X_{pe}|\boldsymbol{\lambda})$ against X_{pe}. So $\left[(X_{pe} - t_{(j-1)})/E_{j-1}\right] \sim \text{Gamma}(1, \lambda_j)$ for $X_{pe} \in \left[t_{(j-1)}, t_{(j)}\right)$, where Gamma$(1, \lambda_j)$ denotes the gamma distribution with location parameter 1 and scale parameter λ_j. Thus, $S(X_{pe}|\boldsymbol{\lambda})$, $f(X_{pe}|\boldsymbol{\lambda})$, $h(X_{pe}|\boldsymbol{\lambda})$, and the cumulative hazard function $H(X_{pe}|\boldsymbol{\lambda})$ are

$$\begin{aligned} S(X_{pe}|\boldsymbol{\lambda}) &= E_{j-1} \exp\left\{-\frac{X_{pe} - t_{(j-1)}}{\lambda_j}\right\}, \\ f(X_{pe}|\boldsymbol{\lambda}) &= E_{j-1} \frac{1}{\lambda_j} \exp\left\{-\frac{X_{pe} - t_{(j-1)}}{\lambda_j}\right\}, \\ h(X_{pe}|\boldsymbol{\lambda}) &= \frac{1}{\lambda_j}, \\ H(X_{pe}|\boldsymbol{\lambda}) &= -\log(E_{j-1}) + \frac{X_{pe} - t_{(j-1)}}{\lambda_j}, \end{aligned} \tag{1}$$

for $X_{pe} \in (t_{(j-1)}, t_{(j)}]$ and $j = 1, \ldots, p+1$. The mean and median survival times are $E(X_{pe}|\boldsymbol{\lambda}) = \lambda_1 + \sum_{i=1}^{p} (\lambda_{i+1} - \lambda_i) E_i$ and $\mu_{1/2}(X_{pe}|\boldsymbol{\lambda}) = t_{(m)} + \lambda_{m+1} \log(2E_m)$, respectively, where m is in $\{1, \ldots, p\}$ such that $E_m \geq 0.5$ and $E_{m+1} < 0.5$.

Suppose a study involves N patients, where D of them failed at time points $t_1 \leq t_2 \leq \cdots \leq t_D$, and the other $(N - D)$ subjects were censored at $t_{D+1}, \ldots, t_N$. Let $t_{1^\star} < t_{2^\star} < \ldots < t_{D^\star}$ denote all the distinct values in $\{t_1, \ldots, t_D\}$ where $D^\star \leq D$. We assume that the censoring mechanism is independent with the distribution of the failure time. Define $t_0 = 0$. The total-time-on-test (TTOT) between t_A and t_B for $t_A < t_B$ is defined to be the sum of all subjects' times in $(t_A, t_B]$ for all $t_A, t_B > 0$; i.e.,

$$\text{TTOT}\,(t_A, t_B) = \sum_{j=1}^{N} \max\left[0,\, \min(t_j, t_B) - t_A\right].$$

The normalized spacing between t_A and t_B is defined as $\text{TTOT}(t_A, t_B)\,/d_{AB}$, where $d_{AB} > 0$ is the number of events occurred in $(t_A, t_B]$.

For exponentially distributed data, the maximum likelihood estimate (MLE) of λ is $\widehat{\lambda} = \sum_{i=1}^{N} t_i / D$ and the $100(1-\alpha)\%$ equal-tailed confidence interval of λ is

$$\lambda \in \left[\frac{2D\widehat{\lambda}}{\chi^2_{2D,1-\alpha/2}},\ \frac{2D\widehat{\lambda}}{\chi^2_{2D,\alpha/2}}\right],$$

where $\chi^2_{2D,\alpha/2}$ is the $100 \times \alpha/2\%$ lower quantile of the central chi-square distribution with $2D$ degree of freedom. Similarly, the first and second order derivatives of the likelihood function of the piecewise exponential distribution $\log L(\boldsymbol{\lambda}|t_1, \ldots, t_N)$ lead to the MLE of λ_j, i.e.,

$$\widehat{\lambda}_j = \frac{\text{TTOT}(t_{(j-1)}, t_{(j)})}{d_{(j)}}$$

and confidence interval of λ_j is

$$\lambda_j \in \left[\frac{2d_{(j)}\widehat{\lambda}_j}{\chi^2_{2d_{(j)},1-\alpha/2}},\ \frac{2d_{(j)}\widehat{\lambda}_j}{\chi^2_{2d_{(j)},\alpha/2}}\right],$$

for all $j = 1, \ldots, p+1$. The MLEs of $S(X_{pe}|\boldsymbol{\lambda})$, $h(X_{pe}|\boldsymbol{\lambda})$, and $M(X_{pe}|\boldsymbol{\lambda})$ are $S(X_{pe}|\widehat{\boldsymbol{\lambda}})$, $h(X_{pe}|\widehat{\boldsymbol{\lambda}})$, and $M(X_{pe}|\widehat{\boldsymbol{\lambda}})$, respectively. The confidence intervals of $\{S(X_{pe}|\boldsymbol{\lambda}), h(X_{pe}|\boldsymbol{\lambda}), M(X_{pe}|\boldsymbol{\lambda})\}$ can be computed by plugging in the lower and upper bounds of $\boldsymbol{\lambda}$. Multiple testing adjustments, e.g., Bonferroni approach, can be implemented to control the level of a confidence interval when multiple model parameters are involved.

2.2 Three Components for Determining the Significant Change-points

The key to the proposed RPEXE model is to set change-points $t_{(1)}, \ldots, t_{(p)}$ given the data $\{t_1, \ldots, t_N\}$. Our approach builds on three components. The first component is a likelihood ratio test. Let x_1 and x_2 denote realizations from Gamma(n_1, λ_1) and Gamma(n_2, λ_2), respectively. We suppose that n_1 and n_2 are *known* positive integers since the numbers of events will be known positive integers. The scale parameters λ_1 and λ_2 are *unknown.* The null and alternative hypotheses are

$$\mathrm{H}_0 : \lambda_1 = \lambda_2 \text{ vs. } \mathrm{H}_1 : \lambda_1 \neq \lambda_2.$$

Under the null hypothesis, we let $\lambda_1 = \lambda_2 = \lambda$. The likelihood ratio test (LRT) statistic can be derived as

$$\phi\left(x_1,\, x_2\right) = \left(\frac{x_1}{x_1 + x_2}\right)^{n_1} \left(\frac{x_2}{x_1 + x_2}\right)^{n_2}.$$

The level α likelihood ratio test rejects H_0 if $\phi\left(x_1,\, x_2\right) \leq C_\alpha$, where C_α is a real value such that $P\left(\phi(X_1, X_2) \leq C_\alpha\right) = \alpha$ under H_0. In Han et al. (2012), we quantified the p-value and proved that this exact likelihood ratio test is the uniformly most powerful unbiased (UMPU) test of H_0 vs. H_1. This test is useful for detecting failure rate change in a single patient group and for testing the equivalence of the failure rates of two exponentially distributed groups.

The second component is a backward elimination procedure, which is used with the LRT to detect significant changes in the failure rate over time. Specifically, given $t_{1^\star} < t_{2^\star} < \cdots < t_{D^\star}$, it is possible to model the data with piecewise exponential distributions having 1 to $D^\star$ pieces. We let $\{\mathrm{T}_1, \mathrm{T}_2, \ldots, \mathrm{T}_{D^\star}\}$ denote $D^\star$ TTOTs in $(0, t_{1^\star}), (t_{1^\star}, t_{2^\star}), \ldots, (t_{D^\star-1}, t_{D^\star})$ and $\{d_1, d_2, \ldots, d_{D^\star}\}$ denote the corresponding numbers of events. Thus $D = \sum_{i=1}^{D^\star} d_i$. The backward elimination (BE) procedure has four steps:

Step 1. Let $\ell = D^\star$. Compute $\ell - 1$ pairs of TTOTs, $\left\{(\mathrm{T}_1, \mathrm{T}_2), \ldots, (\mathrm{T}_{\ell-1}, \mathrm{T}_\ell)\right\}$, and the corresponding pairs of events, $\left\{(d_1, d_2), \ldots, (d_{\ell-1}, d_\ell)\right\}$

Step 2. Compute the $\ell - 1$ p-values using the LRT for the $\ell - 1$ pairs of TTOTs.

Step 3. If the largest p-value exceeds a critical value $\alpha^\star$ and $\ell > 1$, add up the corresponding pair of TTOTs and pair of events (so that there become $\ell - 2$ pairs), and let $\ell = \ell - 1$.

Step 4. Repeat steps 2 and 3 until the largest p-value is smaller than a critical value $\alpha^\star$ or $\ell = 0$.

The third component is an optional order restriction. Expert knowledge may suggest that the change in failure rate follows a pattern. For example, the failure rate of NSCLC patients can decrease over time. In such situations, integrating appropriate order restrictions with a statistical model may be necessary. To implement simple order restriction, we use the pool-adjacent-violators-algorithm (PAVA). The RPEXE with non-increasing and non-decreasing failure rates, which we will call isotonic RPEXE and antitonic RPEXE, are computed by first implementing the PAVA to the $D^\star$ normalized spacings $\{\widehat{\lambda}_1, \ldots, \widehat{\lambda}_{D^\star}\}$, and then running the LRT test of H_0 vs. H_1. We let L denote the number of level sets after the PAVA operation. The resulting L estimates $\widehat{\lambda}_1, \ldots, \widehat{\lambda}_L$ will satisfy the assumed order restriction, and will be used in the BE procedure. The optional order restriction on the failure rate includes 1) isotonic, 2) antitonic, 3) monotonic, 4) increasing then decreasing (IDFR), and 5) decreasing then increasing (DIFR). One can also choose to impose no order restriction. When the partial order restriction is composed of multiple orderings (e.g., IDFR order where the shift time from an increasing to decreasing failure rate is not known), we use the ordering that maximizes the likelihood.

Integrating the three components, we choose the change-points by first implementing an order restriction on the failure rate estimates, and then eliminate all insignificant change-points detected by the likelihood ratio test using the BE procedure. The elimination procedure stops when the largest p-value of all possible change-points is less than a critical value $\alpha^\star$, which controls the type I error of the simultaneous inference of all changepoints.

2.3 Computation of the Critical Value $\alpha^\star$

Given a confidence level α, we estimate the critical value $\alpha^\star$ by $\widehat{\alpha}^\star$ using a 3-stage Monte Carlo (MC) procedure:

Stage 1. Generate an i.i.d. sample $\{x_1, \ldots, x_D\}$ from Exponential(1) (or Gamma$(1, 1)$). Let $D^\star$ denote the number of distinct values in $\{x_1, \ldots, x_D\}$.

Stage 2. If an order restriction is imposed, using PAVA to combine the values that violate the restriction to obtain L distinct level sets.

Stage 3. Initialize $\ell = D^\star$, or $\ell = L$ if there is an order restriction. Repeat the steps 2-3 in the BE procedure $(\ell - 1)$ times. Save the minimum of the p-values as $p_{\min}$.

Repeat the above 3-stage MC procedure N_s times. Let $\widehat{\alpha}^\star$ denote the αth lower quantile among the N_s values. The 95% asymptotic lower and upper bounds of $\alpha^\star$ are the N_s^lth and N_s^uth smallest p-values among all the N_s $p_{\min}$ values, where $N_s^l = N_s \times \left(\alpha - z^{0.025}\sqrt{\alpha(1-\alpha)/N_s}\right)$, $N_s^u = N_s \times \left(\alpha + z^{0.025}\sqrt{\alpha(1-\alpha)/N_s}\right)$, and $z^{0.025}$ is the upper 0.025 quantile of the standard normal distribution.

Following the aforementioned algorithm, we have numerically estimated some critical values for practical use (Han et al. 2014). The log of $\widehat{\alpha}^\star$ values were found to be linearly related with the log of the event number D with $R^2 > 0.9$. Thus we fit two regression lines of $\log(\widehat{\alpha}^\star)$ on $\log(D)$ for $\alpha = 0.05$ and $\alpha = 0.2$, respectively, which can be used as simple formulas to generate $\widehat{\alpha}^\star$ given α and $D \in [20, 800]$. Table 1 shows the estimated regression coefficients $(\widehat{\beta}_0, \widehat{\beta}_1)$ in regression lines of the form

$$\log(\widehat{\alpha}^\star) = \widehat{\beta}_0 + \widehat{\beta}_1 \times log(D)$$

for isotonic RPEXE, monotonic RPEXE, and BE with $\alpha = 0.05$ or $\alpha = 0.1$, and $D \in [20, 800]$. In practice, $\alpha^\star$ can simply be estimated as

$$\widehat{\alpha}^\star = \exp(\widehat{\beta}_0) \times D^{\widehat{\beta}_1},$$

given that the number of event D is between 20 and 800.

	Isotonic RPEXE	Monotonic RPEXE	Umbrella RPEXE	BE, no order restriction
$\alpha = 0.05$	$-3.483; -0.380$	$-4.233; -0.394$	$-1.223; -2.704$	$-2.356; -1.360$
$\alpha = 0.1$	$-2.670; -0.372$	$-3.448; -0.385$	$-1.195; -2.024$	$-1.511; -1.370$

Table 1: Estimated linear coefficients "$\widehat{\beta}_0; \widehat{\beta}_1$" of the isotonic RPEXE, monotonic RPEXE, umbrella alternative RPEXE, and BE for $\alpha = 0.05$ and $\alpha = 0.1$.

2.4 Testing Exponentiality

The RPEXE program can be used to test the exponentiality of an observed set of survival times because it can detect all significant changepoints in the failure rate. Without any significant changepoints the failure rate is a constant and the survival distribution is exponential.

To checking the exponential distribution, one needs to inspect whether any p-value in the list of (max) p-values from the backward elimination is lower than the critical value $\alpha^\star$. If yes, the exponential assumption is violated. We will incorporate this test in the next version of the RPEXE program.

3 Job Files and Inputs/Outputs

3.1 RPEXE in MATLAB

This section summarizes the files and the inputs/outputs of the RPEXE MATLAB program. Users of the RPEXE R program please go to 3.2.

3.1.1 MATLAB Files

1. Files for the likelihood ratio test

   - `totaltest.m` takes inputs (times and the indicator of event/censoring) and returns the total time on test and the number of event.
   - `exact_pvalue.m` computes the exact p-value of the test statistic in the likelihood ratio test. `exact_pvalue.m` is called by both `loopcuts.m`, `loopcuts_ttot.m`, and `loopcuts_umbrella.m`.
   - `bisec.m` runs the bisection algorithm to return a value satisfying a certain equality. `bisec.m` is called by `exact_pvalue.m`.

2. Files for the backward elimination

   - `loopcuts.m` uses a loop format to search changepoints in the backward elimination. This file is called by `RPEXEv1.m` if the order restriction is isotonic, antitonic, monotone, and no restriction.
   - `loopcuts_ttot.m` is the backward elimination job file used by `RPEXEv2.m`. It performs the same job but using the total time on test instead of the event times.
   - `loopcuts_umbrella.m` is the backward elimination job file used by `RPEXEv1.m` when the failure rate is increasing-decreasing or decreasing-increasing.

3. Files for implementing the order restriction

   - `pava.m` uses the event times and censoring status and returns time points (along with the total time on test and the number of events) where the hazard is decreasing (or increasing). `pava.m` is called if any order restriction is specified.
   - `umbrella.m` uses the umbrella alternative to merge certain entries to make sure the sequence of ttot/deaths increasing then decreasing or decreasing

then increasing. (Note that the pava function makes the sequence non decreasing. This function directly uses the pava function.)

- `gamllik.m` computes the log likelihood from the gamma distribution under an order restriction. This function is used to select the trend that maximizes the likelihood.

4. Driver files

    - `RPEXEv1.m` is the main job file if the input is a set of event times and the corresponding censoring status. It also requires the order restriction. The output includes the list of changepoints from the backward elimination procedure with the p-values, the final trend information, model information under different trends, and the failure rate peak information under the umbrella alternative.
    - `RPEXEv2.m` is the main job file if the input is a set of total-time-on-tests (instead of event times). The output structure is the same as `RPEXEv1.m`. The current version of the program (ver 1.0) does not allow umbrella alternative. We will add the function in a later version.

3.1.2 Inputs and Outputs

This package has two driver files `RPEXEv1.m` and `RPEXEv2.m`, which can take different types of inputs but return the same outputs.

Inputs to `RPEXEv1.m` include

- `‘EventTime’ = time`: a sequence of times at which the events occur. Note that 'EventTime' is a required input.
- `‘Censor’ = censor`: a sequence of dichotomous values indicating censored or not (0=censored and 1=not censored). We use `0` to denote censoring and `1` to denote event. The length of `time` and `censor` are identical. Note that 'Censor' is a required input.
- `‘CutTimes’ = cuttimes`, a vector of unique and sorted changepoint candidates. Default is sorted (from small to large) event times. Default == 'EventTime'.
- `‘Trend’`, indicator of the monotonicity assumption
    - `0`: no monotonic assumption;

  - `1`: failure rate is decreasing over time;
  - `2`: failure rate is increasing over time;
  - `3`: monotonic failure rate;
  - `4`: failure rate is increasing and then decreasing;
  - `5`: failure rate is decreasing and then increasing;
  - `6`: failure rate is increasing and then decreasing. The event time corresponding to the highest failure rate will not be considered as a changepoint in order to improve the program stability;
  - `7`: failure rate is decreasing and then increasing. The event time corresponding to the highest failure rate will not be considered as a changepoint in order to improve the program stability.

  Default value of 'Trend'== 0 corresponding to no order restriction.

Inputs to `RPEXEv2.m` are based on the total time on test and the number of events:

- `'Times' = times`, a sequence of times (in ascending order) at which the events occur. For instance, in SEER data sets `times` are the end of each month.
- `'ttot' = ttot`, a vector containing the total time on test during the time periods indicated by `times`.
- `'death' = deaths`, the number of events occurred during the time periods indicated by `times`.
- `'CutTimes' = cuttimes`, same as for `RPEXEv1.m`, `cuttimes` is a vector of unique and sorted changepoint candidates. Default is sorted (from small to large) event times. Default == 'times'.
- `'Monotone' = monotone`, an input having three levels indicating the monotonic assumption
  - `0`: no monotonic assumption;
  - `1`: failure rate is decreasing over time;
  - `2`: failure rate is increasing over time.

The outputs from both `RPEXEv1.m` and `RPEXEv2.m` have the same form. Suppose the output is a structure named "`pexeout`," then its elements are

- `pexeout.times`: changepoint times;
- `pexeout.pvalues`: p-values corresponding to the times `pexeout.times`;
- `pexeout.trend`: trend information in text. It is one of the following: 'No order restriction,' 'Decreasing failure rate,' 'Increasing failure rate,' 'Monotone failure rate,' 'Increasing-decreasing failure rate,' and 'Decreasing-increasing failure rate.'
- `pexeout.struct`: structure information for multiple order restrictions. Each element (i) in the `pexeout.struct` contains the model information under one order restriction.
  - `pexeout.struct(i).time`: sorted time points (from small to large) after the pava algorithm;
  - `pexeout.struct(i).ttot`: the total-time-on-test for each time point counted as from previous time point (0 for the first time point) to the current time point (the interval includes the current time but not the previous time);
  - `pexeout.struct(i).death`: the number of events corresponding to the total-time-on-test;
  - `pexeout.struct(i).loglik`: the log likelihood of based on gamma distribution of the total-time-on-test.

  `pexeout.struct` will be presented only if multiple order restrictions are available (monotonic and umbrella alternative order restrictions).
- `pexeout.changet`: changepoint time corresponding to lowest or highest failure rate for umbrella alternatives. `pexeout.changet` will be presented only under umbrella alternative order restriction.

Note that `pexeout.times` and `pexeout.times` are in the order of the backward selection. Each iteration will select a time point having the largest p-value.

3.2 RPEXE in R

This section summarizes the R functions and the inputs/outputs of the RPEXE R program. Users of the RPEXE MATLAB program please go to 3.1.

3.2.1 R Functions

1. Functions for the likelihood ratio test

    - `totaltest` takes inputs (times and the indicator of event/censoring) and returns the total time on test and the number of event.
    - `exact_pvalue` computes the exact p-value of the test statistic in the likelihood ratio test. `exact_pvalue` is called by both `loopcuts`, `loopcuts_ttot`, and `loopcuts_umbrella`.
    - `bisec` runs the bisection algorithm to return a value satisfying a certain equality. `bisec` is called by `exact_pvalue`.

2. Functions for the backward elimination

    - `loopcuts` uses a loop format to search changepoints in the backward elimination. This file is called by `RPEXEv1` if the order restriction is isotonic, antitonic, monotone, and no restriction.
    - `loopcuts_umbrella` is the backward elimination job file used by `RPEXEv1` when the failure rate is increasing-decreasing or decreasing-increasing.

3. Functions for implementing the order restriction

    - `pava` uses the event times and censoring status and returns time points (along with the total time on test and the number of events) where the hazard is decreasing (or increasing). `pava` is called if any order restriction is specified.
    - `umbrella` uses the umbrella alternative to merge certain entries to make sure the sequence of ttot/deaths increasing then decreasing or decreasing then increasing. (Note that the pava function makes the sequence non decreasing. This function directly uses the pava function.)
    - `gamllik` computes the log likelihood from the gamma distribution under an order restriction. This function is used to select the trend that maximizes the likelihood.

4. Driver files

    - `RPEXEv1` is the driver function for inputs being a set of event times and the corresponding censoring status. It also requires the order restriction. The output includes the list of changepoints from the backward elimination procedure with the p-values, the final trend information, model information under different trends, and the failure rate peak information under the umbrella alternative.

3.2.2 Inputs and Outputs

Inputs to the R function `RPEXEv1` has the format

```
(EventTime=NA,eventtime,Censor=NA,censor,CutTimes=NA,cuttime=1,Trend=NA,trend=0),
```

which is different from the MATLAB format but contains the same input information.

- `EventTime`: any text string to indicate whether there is event time. If it is left empty (or given value NA) then there is no event time. Default == NA.
- `eventtime`: a sequence of times at which the events occur. This is a required input.
- `Censor`: any text string to indicate whether there is censoring status. If it is left empty (or given value NA) then there is no censoring. Default == NA.
- `censor`: a sequence of dichotomous values indicating censored or not (0=censored and 1=not censored). We use `0` to denote censoring and `1` to denote event. The length of `time` and `censor` are identical. Note that 'Censor' is a required input.
- `Cuttimes`: any text string to indicate whether there are candidates of the changepoints. If it is left empty (or given value NA) then there is NO changepoint candidate. Default == NA.
- `cuttime`, a vector of unique and sorted changepoint candidates. Default is sorted (from small to large) event times. Default == event times.
- `Trend`, text string indicator about whether there is trend assumption. If left empty (or given value N) then there is no trend specified. Default = NA.
- `trend`, indicator of the monotonicity assumption
  - `0`: no monotonic assumption;
  - `1`: failure rate is decreasing over time;
  - `2`: failure rate is increasing over time;

  - `3`: monotonic failure rate;
  - `4`: failure rate is increasing and then decreasing;
  - `5`: failure rate is decreasing and then increasing;
  - `6`: failure rate is increasing and then decreasing. The event time corresponding to the highest failure rate will not be considered as a changepoint in order to improve the program stability;
  - `7`: failure rate is decreasing and then increasing. The event time corresponding to the highest failure rate will not be considered as a changepoint in order to improve the program stability.

  Default value of 'Trend'== 0 corresponding to no order restriction.

The outputs from the R function `RPEXEv1` have the same format as in MATLAB RPEXE. Suppose the output is a structure named "`pexeout`," then its elements are

- `pexeout$times`: changepoint times;
- `pexeout$pvalues`: p-values corresponding to the times `pexeout$times`;
- `pexeout$trend`: trend information in text. It is one of the following: 'No order restriction,' 'Decreasing failure rate,' 'Increasing failure rate,' 'Monotone failure rate,' 'Increasing-decreasing failure rate,' and 'Decreasing-increasing failure rate.'
- `pexeout$struct`: structure information for multiple order restrictions. Unlike in MATLAB, `pexeout$struct` contains the following elements.
  - `pexeout$struct[[1]][[1]]`: combined time points (as a single vector, sorted under each order restriction) after the pava algorithm for all order restrictions;
  - `pexeout$struct[[1]][[2]]`: combined total-time-on-tests (as a single vector, sorted under each order restriction) after the pava algorithm for all order restrictions. For each time point, the total-time-on-test was computed as from the previous time point (0 for the first time point) to the current time point (the interval includes the current time but not the previous time);
  - `pexeout.struct[[1]][[3]]`: the number of events corresponding to the total-time-on-test;

    - `pexeout.struct[[1]][[4]]`: the log likelihood values based on gamma distribution of the total-time-on-test under all order restrictions.

  `pexeout$struct` will be presented only if multiple order restrictions are available (monotonic and umbrella alternative order restrictions).

- `pexeout$changet`: changepoint time corresponding to lowest or highest failure rate for umbrella alternatives. `pexeout$changet` will be specified only under umbrella alternative order restrictions.

Note that `pexeout$times` and `pexeout$times` are in the order of the backward selection where each iteration selects a time point having the largest p-value.

3.3 Future Work Topics

We will add the following features in newer version of the program.

1. Include the umbrella order restriction in the `RPEXEv2.m` file.
2. Incorporating the formula for $\alpha^\star$ in the coding to translate the naive p-value to the real p-value.
3. Testing the exponentiality of the survival times. Return the p-value and the estimated power of the test. If one or more significant changepoints are identified, indicate up to what time the exponential assumption can hold.
4. Given the critical changepoints, report the descriptive statistics.

4 A Non-small-cell Lung Cancer Example

We demonstrate using `RPEXEv1.m` in MATLAB and `RPEXEv1` in R for a non-small-cell lung cancer example (Han et al. 2014). The hazard rate is believed to decrease over time and the critical value $\alpha^\star = 0.004$. We will demonstrate 1), how to load the data, 2), how to run RPEXE package with the decreasing hazard rate and monotonic hazard rate assumptions, and 3), how to interpret the results. Kaplan-Meier curve and fitted models have been visualized in Han et al. (2014).

4.1 RPEXE in MATLAB

4.1.1 Loading Data and Specifying Work Directory

The data is saved in file “data.ex1.txt” in the “Examples” directory with two columns having the censoring status and event times. MATLAB and R codes of this example are saved under the same directory with names “MATLAB.ex1.m” and “R.ex1.R” respectively.

```
1.0000 2.9123
1.0000 3.2740
...    ...
1.0000 3.6027
0      31.2192 ;
```

The MATLAB program the program is in folder “MATLAB version”. Code for reading the data and adding the program in the working directory is:

```
% Include the MATLAB files' directory
addpath('..\MATLAB version');

% Load the data
data_ex1 = textread('data.ex1.txt');
oscensor      = data_ex1(:,1);
ostime        = data_ex1(:,2);
```

4.1.2 Run RPEXE with Decreasing and Monotonic Failure Rate Assumptions

Code for running RPEXE with decreasing failure rate and monotonic failure rate is

```
% Fit RPEXE with decreasing failure rate assumption
pexeout1     = RPEXEv1('EventTime',ostime,'Censor',oscensor,...
'Trend',1);

% Fit RPEXE with monotonic failure rate assumption
pexeout2     = RPEXEv1('EventTime',ostime,'Censor',oscensor,...
'Trend',3);
```

4.1.3 Results

Results under the decreasing failure rate assumption:

```
pexeout1
[pexeout1.times pexeout1.pvalues]
% pexeout1 =
%       trend: 'Decreasing failure rate'
%       times: [9x1 double]
%     pvalues: [9x1 double]
% ans =
%     1.8932    0.9780
%    23.4603    0.9608
%    47.8548    0.7539
%    12.7096    0.7045
%    24.7096    0.6876
%     0.0849    0.6013
%     2.3863    0.3060
%    18.5945    0.0852
%    28.0301    0.0000
```

We can see that the time sequence of backward elimination and corresponding p-values. The last time 28.0301 (month) has the naive p-value ¡ 0.004 so that it is a significant changepoint.

Results under the monotonic failure rate assumption:

```
pexeout2
[pexeout2.times pexeout2.pvalues]
% pexeout2 =
%       struct: [1x2 struct]
%        trend: 'Monotone failure rate'
%        times: [9x1 double]
%      pvalues: [9x1 double]
% ans =
%      1.8932    0.9783
%     23.4603    0.9608
%     47.8548    0.7539
%     12.7096    0.7045
%     24.7096    0.6877
%      0.0849    0.6010
%      2.3863    0.3060
%     18.5945    0.0852
%     28.0301    0.0000
```

We can see that the time sequence in the backward elimination and the p-values are the same as the decreasing failure. Next we check the structure.

```
pexeout2.struct
pexeout2.struct(1)
[pexeout2.struct(1).time pexeout2.struct(1).ttot ...
pexeout2.struct(1).deaths]
pexeout2.struct(2)
[pexeout2.struct(2).time pexeout2.struct(2).ttot ...
pexeout2.struct(2).deaths]
% ans =
% 1x2 struct array with fields:
%     time
%     ttot
%     deaths
%     loglik
% ans =
%       time: [10x1 double]
%       ttot: [10x1 double]
%     deaths: [10x1 double]
%     loglik: -543.9703
% ans =
%    1.0e+03 *
%     0.0001    0.0151    0.0020
%     0.0019    0.3021    0.0250
%     0.0024    0.0733    0.0060
%     0.0127    1.0907    0.0760
%     0.0186    0.3450    0.0220
%     0.0235    0.1992    0.0100
%     0.0247    0.0412    0.0020
%     0.0280    0.0997    0.0040
%     0.0479    0.3699    0.0060
%     0.0511    0.0825    0.0010
% ans =
%       time: [2x1 double]
%       ttot: [2x1 double]
%     deaths: [2x1 double]
%     loglik: -559.4657
% ans =
%    1.0e+03 *
%     0.0006    0.1130    0.0050
%     0.0511    2.5057    0.1490
```

We can see that the structure has two elements corresponding to decreasing failure and increasing failure. In each element we can see the (sorted) time points after PAVA, the TTOT, the numbers of events, and the log likelihood. Under the decreasing failure assumption we can see 10 time points and 10 TTOTs, and the log likelihood is -543.9703. Under the increasing failure assumption there are only 2 time points left after the PAVA, and the log likelihood value is -559.4657, which is less than that under the decreasing failure.

4.2 RPEXE in R

4.2.1 Loading Data and Specifying Work Directory

The data is saved in file "data.ex1.txt" in the "Examples" directory with two columns having the censoring status and event times. The R program is saved in folder "R version" with the name "RcodeREPEXEv1.txt".

The following code can load the data and point to the source code

```
# Set your current work directory that contains this code
setwd("your current work directory")

# include the R functions from the command line
source("../R version/RcodeREPEXEv1.txt")

# load the data
data_ex1 <- read.table("data.ex1.txt", header=F)
oscensor <- data_ex1[,1]
ostime   <- data_ex1[,2]
```

4.2.2 Run RPEXE with Decreasing and Monotonic Failure Rate Assumptions

Code for running RPEXE with decreasing failure rate and monotonic failure rate is

```
# Run RPEXE with decreasing failure rate
pexeout1 <- RPEXEv1(EventTime="EventTime",ostime,
Censor="Censor", oscensor,Trend="Decreasing",trend=1)

# Run RPEXE with monotonic failure rate
pexeout2 <- RPEXEv1(EventTime="EventTime",ostime,
Censor="Censor",oscensor,Trend="monotonic",trend=3)
```

4.2.3 Results

Results under the decreasing failure rate assumption:

```
pexeout1
#$times
#[1]  1.8932 23.4603 47.8548 12.7096 24.7096  0.0849
 2.3863 18.5945 28.0301
#$pvalues
#[1] 9.782674e-01 9.608166e-01 7.538902e-01 7.045443e-01
 6.876717e-01 6.010137e-01 3.059760e-01
#[8] 8.523530e-02 9.099848e-07
#$trend
#[1] "Decreasing falilure rate"
#$struct
#NULL
#$changet
#NULL
```

We can see that the time sequence of backward elimination and corresponding p-values. The last time 28.0301 (month) has the naive p-value 9.099848e-07 ¡ 0.004 so that it is a significant changepoint.

Results under the monotonic failure rate assumption:

```
pexeout2
# $times
# [1]  1.8932 23.4603 47.8548 12.7096 24.7096
0.0849  2.3863 18.5945 28.0301
# $pvalues
# [1] 9.782674e-01 9.608166e-01 7.538902e-01
7.045443e-01 6.876717e-01 6.010137e-01 3.059760e-01
# [8] 8.523530e-02 9.099848e-07
# $trend
# [1] "Monotone failure rate"
```

We can see that the time sequence in the backward elimination and the p-values are the same as the decreasing failure. Next we check the four elements of the structure.

```
# $struct[[1]][[1]]
#  [1]  0.0849  1.8932  2.3863 12.7096 18.5945 23.4603
24.7096 28.0301 47.8548 51.0767  0.6438 51.0767
# $struct[[1]][[2]]
#  [1]   15.0794  302.0514   73.3403 1090.6864  345.0394
199.2019 41.2268   99.7137  369.8969 82.4547  112.9527 2505.7382
# $struct[[1]][[3]]
#  [1]   2  25   6  76  22  10   2   4   6   1   5 149
# $struct[[1]][[4]]
#           [,1]      [,2]
# [1,] -543.9703 -559.4657
# $changet
# NULL
```

Unlike in MATLAB, the time points after PAVA (`$struct[[1]][[1]]`), TTOTs (`$struct[[1]][[2]]`) and numbers of events (`$struct[[1]][[3]]`) are three vectors as the combination of results under decreasing failure rate and increasing failre rate. So their lengths are 12=10+2. The two log liklhood values are saved in `$struct[[1]][[4]]`) being -543.9703 for decreasing fauilure (trend = 1) -559.4657 and increaing failure (trend = 2). Results from the R program are identical to that in the MATLAB program.

Disclaimer

This set of programs is free software: you can redistribute it and/or modify it under

Reference

Han, G., Schell, M., and Kim, J. (2012) "Comparing Two Exponential Distributions Using the Exact Likelihood Ratio Test," *Statistics in Biopharmaceutical Research*, 4(4), 348-356.

Han, G., Schell, M., and Kim, J. (2014) "Improved Survival Modeling in Cancer Research Using a Reduced Piecewise Exponential Approach," *Statistics in Medicine*, 33(1), 59-73.