PULSE: A Suite of R Functions for Detecting Pulsatile Hormone Secretions

Yu-Chieh Yang
Department of Statistics, National Taichung Institute of Technology, Taiwan.
email: yuchieh@ntit.edu.tw
Anna Liu
Department of Mathematics and Statistics, University of Massachusetts, Amherst, MA 01003.
email: anna@math.umass.edu
Yuedong Wang
Department of Statistics and Applied Probability, University of California, Santa Barbara, CA 93106.
email: yuedong@pstat.ucsb.edu
September 7, 2004

1 Introduction

PULSE is a R package for hormone pulse detection and estimation based on the nonlinear mixed effects partial spline models proposed in Yang, Liu and Wang (2004). Yang et al. (2004) proposed the following algorithm (simplified)

Step 1 *Initialize*: identify potential pulse locations and provide initial values. Denote the total number of potential pulses as K_{max}. Specify a low bound for the number of pulses K_{min}.

Step 2 *Pulse detection*:

1. For $K = K_{max}, K_{max} - 1, \cdots, K_{min}$, fit a nonlinear mixed effects partial spline model with K pulses and delete the location with the least significant amplitude.
2. Select the final model using one the AIC, BIC, RIC and GCV criteria.

Step 3 *Parameter estimation*. Fit the final model.

The PULSE package consists of three main functions, *pulini*, *puldet* and *pulest*, for three steps in the above algorithm, and other pulse detection and utility functions. In this manual, we describe these functions and illustrate how to use them with examples.

2 R Functions and Data

`pulini` *Identify Initial Pulse Locations*

Description

This function accomplishes the first step of the algorithm in Yang, Liu and Wang (2004). It finds potential pulse locations. The function itself can also be used for pulse detection.

Usage

```
pulini(x, y, data, method = c("pcp", "CLUSTER"), alpha,
     control=list(pcp=list(spline=list(nb=~x, rk=cubic(x)), spar="v",
     limnla=c(-10, 3)), cluster=list(sd=mean(y)*0.07, nnadir=2, npeak=3)))
```

Arguments

`x`	a vector of observation time points.
`y`	a vector of hormone concentrations.
`data`	a data frame containing the variables occurring in the *x* and *y* arguments. If this option is not specified, the variables should be on the search list. Missing values are not allowed.
`method`	the method to be used for identifying initial pulse locations. If "pcp", the change point method based on partial smoothing spline models is used to detect pulse locations as change points to the first derivative of the mean function. If "CLUSTER", the CLUSTER method proposed by Veldhuis and Johnson (1986) is used.
`alpha`	for '`method="pcp"`', *alpha* controls the significance level of a potential change point; for '`method="CLUSTER"`', *alpha* controls the significance level of the t test.
`control`	A list of two components, *pcp* and *cluster*, to replace the default values in the `pcp` and `CLUSTER` functions.

Details

`pulini` is a wrapper of two other functions, `pcp` and `CLUSTER`. See these two functions for details about control options. Larger *alpha* leads to more identified pulses, thus increases false positive rate and decreases false negative rate. `CLUSTER` is faster than `pcp`, however, its false negative rate is usually a bit larger.

Value

a vector of pulse locations.

Author(s)

Yu-Chieh Yang, Anna Liu, Yuedong Wang

References

Veldhuis, J. D. and Johnson, M. L. (1986), Cluster analysis: a simple versatile and robust algorithm for endocrine pulse detection, *American Journal of Physiology*, **250**, E486-E493.

Yang, Y. (2002), *Detecting Change Points and Hormone Pulses Using Partial Spline Models*, Ph.D. Thesis, University of California-Santa Barbara, Dept. of Statistics and Applied Probability.

Yang, Y. and Liu, A. and Wang, Y., (2004), *Detecting Pulsatile Hormone Secretions Using Nonlinear Mixed Effects Partial Spline Models*. Available at `www.pstat.ucsb.edu/faculty/yuedong/research`.

See Also

`CLUSTER`, `pcp`

Examples

```
pl1 <- pulini(time, conc, data=acth, method="pcp", alpha=0.6)
pl2 <- pulini(time, conc, data=acth, method="CLUSTER", alpha=.2,
              control=list(cluster=list(sd=.05*mean(acth$conc))))
```

`pcp` *Potential Change Points*

Description

Detect change points to the dth derivative of the mean function using partial smoothing spline models.

Usage

```
pcp(x, y, data, d=1, spline = list(nb=~x, rk=cubic(x)),
    spar="v", limnla=c(-10, 3), alpha)
```

Details

The mean function is assumed to be smooth, except for some potential change points to the derivatives. The smooth function plus representations of those potential change points is called a partial spline model (Wahba, 1990). We use the methods proposed in Yang (2002) to detect potential change points. The `ssr` function in the **assist** package is used to fit partial spline models. See `cran.us.r-project.org/doc/packages/assist.pdf` for more information about *nb*, *rk*, *spar* and *limnla*.

Value

a vector of pulse locations.

Author(s)

Yu-Chieh Yang, Anna Liu, Yuedong Wang

References

Wahba, G. (1990), *Spline Models for Observational Data*, SIAM, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 59, Philadelphia.

Wang, Y. and Ke, C. (2002), *ASSIST: A Suite of S-plus functions Implementing Spline smoothing Techniques*. Available at `cran.us.r-project.org/src/contrib/PACKAGES.html`. Manual for the ASSIST package is available at `www.pstat.ucsb.edu/faculty/yuedong/software`.

Yang, Y. (2002), *Detecting Change Points and Hormone Pulses Using Partial Spline Models*, Ph.D. Thesis, University of California-Santa Barbara, Dept. of Statistics and Applied Probability.

See Also

`pulini`, `ssr`

Examples

```
pl3 <- pcp(time, conc, data=acth, alpha=0.6)
```

`CLUSTER` *CLUSTER Method for Detecting Pulse Locations.*

Description

Detect pulse locations using the CLUSTER method

Usage

```
CLUSTER(x, y, data, sd=.07*mean(y), nnadir=2, npeak=3, alpha)
```

Arguments

`x`	a vector of observation time points.
`y`	a vector of hormone concentrations.
`data`	a data frame containing the variables occurring in the x and y arguments. If this option is not specified, the variables should be on the search list. Missing values are not allowed.
`sd`	standard deviation for the pooled t test. Default is 7% of the mean response.
`nnadir`	number of nadir points used in the t statistics. Default is 2.
`nnpeak`	number of peak points used in the t statistics. Default is 3.
`alpha`	Significance level of the t statistics.

Details

CLUSTER uses t test between two overlapping consecutive sets of points to identify significant increases and decreases in observations. The default for the standard deviation *sd* is by no means standard. Users are recommended to provide an accurate estimate instead of using the default.

Value

A vector of pulse locations.

Author(s)

Yu-Chieh Yang, Anna Liu, Yuedong Wang

References

Veldhuis, J. D. and Johnson, M. L. (1986), Cluster analysis: a simple versatile and robust algorithm for endocrine pulse detection, *American Journal of Physiology*, **250**, E486-E493.

See Also

`pulini`

Examples

```
pl4 <- CLUSTER(time, conc, data=acth, sd=.05*mean(acth$conc), alpha=.2)
```

`baseini` *Initial Values for the Baseline Function*

Description

This function can be used to derive initial values for the baseline function which are required by two other functions: `puldet` and `pulest`.

Usage

```
baseini(x, y, data, puloc, method=c("shift", "select"),
     smooth=c("spline", "loess"), control=list(shift=-1, spline=list(nb=~x,
     rk=cubic(x), spar="v", limnla=c(-1.5, 0)), loess=list(span=0.75),
     select=list(npl=2, npr=3)))
```

Arguments

`x`	a vector of observation time points.
`y`	a vector of hormone concentrations.
`data`	a data frame containing the variables occurring in the *x* and *y* arguments. If this option is not specified, the variables should be on the search list. Missing values are not allowed.
`puloc`	a vector of pulse locations.

`method`	a character string. If '`method="shift"`', a smooth curve will be fit to all observations using a smoothing method specified by the *smooth* argument, and then the whole curve will be shifted downward; if '`method="select"`', a smooth curve will be fit to the selected observations without those around pulse locations.
`smooth`	a character string specifying which smoothing method will be used: '`smooth="spline"`' for smoothing spline and '`smooth="loess"`' for `loess`.
`control`	a list of control parameters for different *method* and *smooth* options. Names of the list are *shift*, *spline*, *select*, and *loess*. When '`method="shift"`', *shift* specifies the amount of shifting downward with a nonnegative value; the default '`shift=-1`' asks the program to estimate the amount of shift. When '`method="select"`', the *select* list in the control argument decides the number of observation points to be eliminated on the left (*npl*) and on the right (*npr*) of each pulse location. When '`smooth="spline"`', as in `pcp`, the *spline* list in the control argument specifies the null space, reproducing kernel, and method and range for estimating the smoothing parameter. When '`smooth="loess"`', the *loess* list in the control argument specifies the span for the `loess` function.

Details

When '`smooth="spline"`', the `ssr` function in the **assist** package is used to fit the spline function. See `cran.us.r-project.org/doc/packages/assist.pdf` for more information about *nb*, *rk*, *spar* and *limnla*. The appropriate choices of *npl* and *npr* depend on the sampling rate as well as prior knowledge about the infusion and decay rates.

Value

Estimated baseline function evaluated at x.

Author(s)

Yu-Chieh Yang, Anna Liu, Yuedong Wang

References

Wang, Y. and Ke, C. (2002), *ASSIST: A Suite of S-plus functions Implementing Spline smoothing Techniques*. Available at `cran.us.r-project.org/src/contrib/PACKAGES.html`. Manual for the ASSIST package is available at `www.pstat.ucsb.edu/faculty/yuedong/software`.

Yang, Y. (2002), *Detecting Change Points and Hormone Pulses Using Partial Spline Models*, Ph.D. Thesis, University of California-Santa Barbara, Dept. of Statistics and Applied Probability.

See Also

`loess`, `ssr`

Examples

```
bl1 <- baseini(time, conc, data=acth, puloc=pl1, method="shift", smooth="spline")
bl2 <- baseini(time, conc, data=acth, puloc=pl1, method="select", smooth="spline")
bl3 <- baseini(time, conc, data=acth, puloc=pl1, method="shift", smooth="loess")
bl4 <- baseini(time, conc, data=acth, puloc=pl1, method="select", smooth="loess")
```

`puldet` *Pulse Detection*

Description

This function accomplishes the second step of the algorithm in Yang, Liu and Wang (2004). It detects pulse locations.

Usage

```
puldet(x, y, data, baseline=list(nb=~x, rk=cubic(x)),
       start, type = c("dblexp", "user"),
       params=list(infrate="common", decrate="mixed", amplitude="mixed"),
       pulse.usr=list(shape, parlist, devlist, drop.var),
       weights = NULL, correlation = NULL, control=pul.control())
```

Arguments

`x`	a vector of observation time points.
`y`	a vector of hormone concentrations.
`data`	a data frame containing the variables occurring in the x and y arguments. If this option is not specified, the variables should be on the search list. Missing values are not allowed.
`baseline`	a list specifies the model space for the baseline function. *nb* is a one-sided formula specifying the basis of the null space. *rk* is an expression specifying the reproducing kernel of the orthogonal complementary space to the null space. The default options of *nb* and *rk* correspond to the cubic spline. If '`baseline=0`', the baseline function is fixed at its initial input.
`start`	a list of initial values. It includes four components: *puloc*, a vector of initial pulse locations; *kmin*, the minimal number of pulses at which the program will stop the elimination process; *fixed*, a vector of starting values for the *common* and *fixed* parameters specified in the argument *params* or in the component *parlist* of the argument *pulse.usr* (note that the *order* is very important: the initial values for *common* parameters appear before those for *fixed* parameters); *Inif*, initial baseline estimates evaluated at x.
`type`	a character string specifies a predefined type of the pulse shape functions. If '`type="dblexp"`', a double exponential function will be used as the pulse shape

function. If '`type="user"`', the user can input the shape function and parameters in the argument *pulse.usr*.

`params` when '`type="dblexp"`', *params* is a list of three components, *infrate*, *decrate* and *amplitude*, which correspondingly specifies models to be assumed for the pulse infussion rate, pulse decay rate and pulse amplitude parameters. Current possible options are "common" for *infrate*; "common", "fixed" and "mixed" for pulse *decrate*; "fixed", "mixed" and "random" for *amplitude*. Here "common" means that a common parameter is assumed in all pulses; "fixed" means that different deterministic parameters are assumed for all pulses; "random" means that the parameters are assumed to be from a normal distribution with mean 0; "mixed" means that the paramters are assumed to be from a normal distribution with a unknown constant mean. Current program assumes that all random parameters are independent.

`pulse.usr` when '`type="user"`', *pulse.usr* is a list of four components, *shape*, *parlist*, *devlist*, and *drop.var*, which correspondingly specify the pulse shape, pulse parameters, first derivatives of the pulse shape function with respect to the parameters, and the variable used to drop pulses. *shape* is a formula. *parlist* is a list of character vectors with four components: *common*, *random*, *fixed*, and *p*. The component *common* specifies fixed parameters used in the pulse formula which are common for all pulses; the component *random* specifies random parameters in the pulse formula; and the component *fixed* specifies fixed parameters which are different for each pulse. Current program assumes that all random parameters are independent. The fourth component *p* is used to specify the parameter name of the pulse locations. *devlist* is a list of the derivatives in the order of the common, random and fixed parameters. For each type of parameters, the order should be consistent with the order of the parameters supplied in *parlist*. *drop.var* is a character variable that specifies which parameter to use for dropping a pulse. Note that this parameter should not be specified as common.

`weights` an optional varFunc object or one-sided formula describing the heteroscedasticity structure. See the documentation on varClasses in the **nlme** package for a description of the available varFunc classes. Defaults to NULL, corresponding to homoscesdatic errors.

`correlation` an optional corStruct object describing the correlation structure. See the documentation of corClasses in the **nlme** package for a description of the available corStruct classes. Defaults to NULL, corresponding to no correlations.

`control` a list of control values for the detection algorithm to replace the default values returned by the function `pul.control`.

Details

`puldet` creates a sequence of nested sets of pulse locations by eliminating the least significant locations one-by-one, and selects final pulse locations by a model selection criterion such as AIC, BIC, RIC and GCV.

Value

a list which contains

- `tbl` — a summary matrix of the detection process with columns correspond to the number of pulses (NP), *BIC* (BIC), *RIC* (RIC), *AIC* (AIC), *GCV* (GCV), total degrees of freedom (DF), degrees of freedom of the baseline function (df(base)), smoothing parameters in $\log_{10}$ scale (nlaht), residual sum of squares (RSS), residual variance (Sigma2), the index of the location dropped at the current iteration (DROP).
- `est` — a list, with each component again a list recording estimation at each set of pulse locations. Each sublist contains the following components: *coef* for estimated parameters; *Var* for variance of the estimated parameters; *baseFitted* for baseline fit; *nlmeFitted* for fit of pulsatile hormone secretions; *nlmeObj* for nlme object of the `nlme` function if it is used; *gnlsObj* for gnls object of the `gnls` function if it is used; *callxObj* for object returned from the inside function NLMEcallx.
- `stat` — a list of matrices recording the statistics used to dertimine the "DROP" variables.
- `detloc` — a list of four sublists AIC, GCV, BIC and RIC, with each sublist contains the estimated pulse number (*pulnum*) and pulse locations (*puloc*), as well as estimated coefficients (*coef*), baseline function (*baseline*) and pulsatile hormone secretions based on the selected model.

Author(s)

Yu-Chieh Yang, Anna Liu, Yuedong Wang

References

Pinheiro, J. and Bates, D. M. (2000), *Mixed-effects Models in S and S-plus*, Springer, New York.

Yang, Y. (2002), *Detecting Change Points and Hormone Pulses Using Partial Spline Models*, Ph.D. Thesis, University of California-Santa Barbara, Dept. of Statistics and Applied Probability.

Yang, Y. and Liu, A. and Wang, Y., (2004), *Detecting Pulsatile Hormone Secretions Using Nonlinear Mixed Effects Partial Spline Models*. Available at `www.pstat.ucsb.edu/faculty/yuedong/research`.

Wang, Y. and Ke, C. (2002), *ASSIST: A Suite of S-plus functions Implementing Spline smoothing Techniques*. Available at `cran.us.r-project.org/src/contrib/PACKAGES.html`. Manual for the ASSIST package is available at `www.pstat.ucsb.edu/faculty/yuedong/software`.

See Also

`baseini`, `nlme`, `pul.control`, `pulini`, `ssr`,`summary.puldet`

Examples

```
det.cmr <- puldet(time, conc, data=acth, type=c("dblexp"),
    baseline=list(nb=~time, rk=cubic(time)),
```

```
    start=list(puloc=pl1, kmin=5, fixed=fix.ini[1:2], Inif=bl1),
    params=list(infrate="common", decrate="mixed", amplitude="random"))

det.ccm <- puldet(time, conc, data=acth, type=c("dblexp"),
    start=list(puloc=pl1, kmin=5, fixed=fix.ini, Inif = bl1),
    params=list(infrate="common", decrate="common", amplitude="mixed"))
```

`pulest` *Pulse Estimation*

Description

This function accomplishes the third step of the algorithm in Yang, Liu and Wang (2004). It estimates all parameters in the model.

Usage

```
pulest(x, y, data, baseline=list(nb=~x, rk=cubic(x)),
       puloc, start, type = c("dblexp", "user"),
       params=list(infrate="common", decrate="mixed", amplitude="mixed"),
       pulse.usr=list(shape, parlist, devlist),
       weights = NULL, correlation = NULL, control=pul.control(IDF=1))
```

Arguments

`x`	a vector of observation time points.
`y`	a vector of hormone concentrations.
`data`	a data frame containing the variables occurring in the *x* and *y* arguments. If this option is not specified, the variables should be on the search list. Missing values are not allowed.
`baseline`	a list specifies the model space for the baseline function. *nb* is a one-sided formula specifying the basis of the null space. *rk* is an expression specifying the reproducing kernel of the orthogonal complementary space to the null space. The default options of *nb* and *rk* correspond to the cubic spline. If '`baseline=0`', the baseline function is fixed at its initial input.
`start`	a list of initial values. It includes two components: *fixed*, a vector of starting values for the *common* and *fixed* parameters specified in the argument *params* or in the component *parlist* of the argument *pulse.usr* (note that the *order* is very important: the initial values for *common* parameters appear before those for *fixed* parameters); *Inif*, initial baseline estimates evaluated at *x*.
`type`	a character string specifies a predefined type of the pulse shape functions. If '`type="dblexp"`', a double exponential function will be used as the pulse shape function. If '`type="user"`', the user can input the shape function and parameters in the argument *pulse.usr*.

`params` when '`type="dblexp"`', *params* is a list of three components, *infrate*, *decrate* and *amplitude*, which correspondingly specifies models to be assumed for the pulse infussion rate, pulse decay rate and pulse amplitude parameters. Current possible options are "common" for *infrate*; "common", "fixed" and "mixed" for pulse *decrate*; "common", "fixed", "mixed" and "random" for *amplitude*. Here "common" means that a common parameter is assumed in all pulses; "fixed" means that different deterministic parameters are assumed for all pulses; "random" means that the parameters are assumed to be from a normal distribution with mean 0; "mixed" means that the paramters are assumed to be from a normal distribution with a unknown constant mean. Current program assumes that all random parameters are independent.

`pulse.usr` when '`type="user"`', *pulse.usr* is a list of three components, *shape*, *parlist*, and *devlist*, which correspondingly specify the pulse shape, pulse parameters, and first derivatives of the pulse shape function with respect to the parameters. *shape* is a formula. *parlist* is a list of character vectors with four components: *common*, *random*, *fixed*, and *p*. The component *common* specifies fixed parameters used in the pulse formula which are common for all pulses; the component *random* specifies random parameters in the pulse formula; and the component *fixed* specifies fixed parameters which are different for each pulse. Current program assumes that all random parameters are independent. The fourth component *p* is used to specify the parameter name of the pulse locations. *devlist* is a list of the derivatives in the order of the common, random parameters and fixed parameters. For each type of parameters, the order should be consistent with the order of the parameters supplied in palist.

`weights` an optional varFunc object or one-sided formula describing the heteroscedasticity structure. See the documentation on varClasses in the **nlme** package for a description of the available varFunc classes. Defaults to NULL, corresponding to homoscesdatic errors.

`correlation` an optional corStruct object describing the correlation structure. See the documentation of corClasses in the **nlme** package for a description of the available corStruct classes. Defaults to NULL, corresponding to no correlations.

`control` a list of control values for the estimation algorithm to replace the default values returned by the function `pulest.control`.

Details

`pulest` estimates all parameters using the double-penalized log-likelihood and the approximate profile-likelihood (Yang, Liu and Wang, 2004).

Value

a list which contains

`tbl` a summary matrix of the fit with columns corresponding to the number of pulses

(NP), *BIC* (BIC), *RIC* (RIC), *AIC* (AIC), *GCV* (GCV), total degrees of freedom (DF), degrees of freedom of the baseline function (df(base)), smoothing parameters in $\log_{10}$ scale (nlaht), and residual sum of squares (RSS).

`coef` estimated parameters.

`Var` Covariance matrix of the estimated parameters.

`baseFitted` estimates of the baseline function.

`nlmeFitted` estimates of the hormone secretion with baseline subtracted.

`nlmeObj` nlme object of the `nlme` function if it is used.

`gnlsObj` gnls object of the `gnls` function if it is used.

`callxObj` object returned from the inside function NLMEcallx or GNLScallx.

`nlme` whether nlme is used in the fitting.

Author(s)

Yu-Chieh Yang, Anna Liu, Yuedong Wang

References

Pinheiro, J. and Bates, D. M. (2000), *Mixed-effects Models in S and S-plus*, Springer, New York.

Yang, Y. (2002), *Detecting Change Points and Hormone Pulses Using Partial Spline Models*, Ph.D. Thesis, University of California-Santa Barbara, Dept. of Statistics and Applied Probability.

Yang, Y. and Liu, A. and Wang, Y., (2004), *Detecting Pulsatile Hormone Secretions Using Nonlinear Mixed Effects Partial Spline Models*. Available at `www.pstat.ucsb.edu/faculty/yuedong/research`.

Wang, Y. and Ke, C. (2002), *ASSIST: A Suite of S-plus functions Implementing Spline smoothing Techniques*. Available at `cran.us.r-project.org/src/contrib/PACKAGES.html`. Manual for the ASSIST package is available at `www.pstat.ucsb.edu/faculty/yuedong/software`.

See Also

`baseini`, `nlme`, `pul.control`, `puldec`, `ssr`, `summary.pulest`

Examples

```
fit.cmm <- pulest(time, conc, data=acth, type=c("dblexp"),
    puloc=pl5, start=list(fixed=fix.ini, Inif=bl1),
    params=list(infrate="common", decrate="mixed", amplitude="mixed"),
    control =list(pul=list(IDF=1)))
```

`pul.control` *Control Values for puldet and pulest*

Description

The values supplied in the function call replace the defaults and a list with all possible arguments is returned. The returned list is used as the control argument to the puldet or pulestthe function.

Usage

```
pul.control(maxIter= 100, pnlsMaxIter=5, msMaxIter=10,
         tolerance=10e-3, pnlsTol=10e-3, msTol=10e-3, gradHess=TRUE,
         apVar=TRUE, returnObject=TRUE, msVerbose=FALSE, spar="v",
         TOLr=10e-3, nlsMaxIter=5, minScale=0.01, nlsTol=0.01,
         limnla=c(-3,1), IDF=1.2)
```

Arguments

`spar`	method for selecting the smoothing parameter in the estimation of the baseline function. '`spar="v"`' for GCV (default) and '`spar="m"`' for GML.
`limnla`	a vector of length one or two, specifying a search range for log10(n*lambda), where lambda is the smoothing parameter and n is the sample size. If it is a single value, the smoothing parameter will be fixed at this value. Default is c(-10, 3).
`TOLr`	tolerance level of the GCV or GML scores for the iteration between the baseline estimation and estimation of the pulsatile part.
`IDF`	Inflated degree of freedom due to the selection process. Default is 1.2.
`trace`	a logic value which indicates whether the selection procedure should be printed out.
`others`	Other arguments are used for either `nlme` or `gnls` control. Refer to the manual of `nlme` and `gnls` in the **NLME** package for their definitions.

Value

a list with names *ssr*, *pul*, *gnls*, and *nlme*. *ssr* is a list with names *method* and *limnla*. *pul* is a list with names *TOLr*, *IDF* and *trace*. *gnls* and *nlme* contains control parameters for the `gnls` and `nlme` functions.

Author(s)

Yu-Chieh Yang, Anna Liu, Yuedong Wang

References

Pinheiro, J. and Bates, D. M. (2000), *Mixed-effects Models in S and S-plus*, Springer, New York.

Yang, Y. (2002), *Detecting Change Points and Hormone Pulses Using Partial Spline Models*, Ph.D. Thesis, University of California-Santa Barbara, Dept. of Statistics and Applied Probability.

Yang, Y. and Liu, A. and Wang, Y., (2004), *Detecting Pulsatile Hormone Secretions Using Nonlinear Mixed Effects Partial Spline Models*. Available at `www.pstat.ucsb.edu/faculty/yuedong/research`.

Wang, Y. and Ke, C. (2002), *ASSIST: A Suite of S-plus functions Implementing Spline smoothing Techniques*. Available at `cran.us.r-project.org/src/contrib/PACKAGES.html`. Manual for the ASSIST package is available at `www.pstat.ucsb.edu/faculty/yuedong/software`.

Examples

```
# change IDF=1 when there is no selection and
# decrease the maximum number iterations
pul.control(maxIter= 30, IDF=1.2)
```

`summary.puldet` *Summarizing Pulse Detection*

Description

Produce summary for class “puldet”.

Usage

```
summary(object)
```

Arguments

`object`	an object of class “puldet”, usually a result of a call to `puldet`.

Details

This function summarizes the pulse detection results. It prints the call to `puldet`, a summary table of the detection process, and the pulses identified by different criteria. It is recommended to use the output of `puldet` to extract detected pulses and estimates.

Author(s)

Yu-Chieh Yang, Anna Liu, Yuedong Wang

Examples

```
summary(det.cmr)
```

`summary.pulest` *Summarizing Pulse Estimation*

Description

Produce summary for class "pulest".

Usage

```
summary(object)
```

Arguments

`object`	an object of class "pulest", usually a result of a call to`pulest`.

Details

This function summarizes the pulse estimation results. It prints the call to `pulest`, a summary table of the model, a table of estimated common and fixed parameters, a table of estimated variance of the random effects, if any, and a summary of the baseline estimate. It is recommended to use the output of `pulest` to extract estimates.

Author(s)

Yu-Chieh Yang, Anna Liu, Yuedong Wang

Examples

```
summary(fit.cmm)
```

3 Load into R

After downloading and decompressing the file, the R codes, data and help files are in the directories *R*, *data* and *man* respectively. One can build a package called *pulse* using *R CMD build*. Then the library can be loaded using *library(pulse, lib.loc="path")*. The help files and data will also be available through this library. Another way to load is to directly source these R functions and data as follows:

```
> source("path/ssrfuns.R")
> source("path/puldet.R")
> source("path/pulest.R")
> source("path/pulini.R")
> source("path/baseini.R")
> acth <- source("path/acth.R")$value
> ewe <- source("path/ewe.R")$value
```

The R packages *ASSIST* and *nlme* are required which can be downloaded from *http://cran.r-project.org* or *http://www.pstat.ucsb.edu/faculty/yuedong/software.html*, and loaded into R using

```
> library(assist, lib.loc=``path'')
```

4 Examples

We use real data sets to illustrate possible applications of the functions in the PULSE package. We use the default double exponential pulse shape function. Other shape function will be added later.

4.1 Adrenocorticotropic hormone

The data frame, *acth*, consists two variables, *time* and *conc*, which represent hormones adrenocorticotropic (ACTH) concentrations over a period of 24 hours from one subject. See Yang et al. (2004) for more information about the experiment. Our first step is to detect potential pulse locations using *pulini*:

```
> pl1 <- pulini(time, conc, data=acth, method=''pcp'', alpha=0.6)
> pl1
 [1]  2.833333  3.500000  6.333333  8.500000 13.500000 16.000000 17.000000
 [8] 19.666667 21.000000 22.666667
> pl2 <- pulini(time, conc, data=acth, method="CLUSTER", alpha=.2,
              control=list(cluster=list(sd=.05*mean(acth$conc))))
> pl2
 [1]  3.000000  3.500000  7.166667  8.333333 13.500000 14.666667 17.000000
 [8] 18.833333 19.500000 21.000000 21.666667 22.666667
```

Those locations are shown in Figure 1.

There is a general agreement between locations detected by those two methods. As indicated in our help files that *pulini* is a wrapper of two other functions, *pcp* and *CLUSTER*, therefore, we can use those functions directly and get the same locations.

```
> pl2 <- pcp(time, conc, data=acth, alpha=0.6)
> pl4 <- CLUSTER(time, conc, data=acth, sd=.05, nnadir=2, npeak=3, alpha=.2)
```

Note that the default for the *spline* option of the *pcp* method is cubic spline. We used the default here. It can be change to other forms of spline functions. Also, the design points, *time*, are automatically rescaled into the interval [0,1]. These remarks are also true for other functions such as *baseini*, *puldet* and *pulest*, which will not be repeated.

Other methods such as PULSAR (Merriam and Wachter 1982), ULTRA (Van Cauter, L'Hermite, Copinschi, Refetoff, Desir and Robyn 1981) and wavelets may also be used to detect potential pulse locations. These methods will be added later.

In the first step, we also need to derive initial values for the baseline function and parameters. Our experiences indicate that initial values are very important to the performance of our algorithm. Non-convergence are often caused by bad initial values. We can use the function *baseini* to derive initial values for the baseline function:

```
> bl1 <- baseini(time, conc, data=acth, puloc=pl1, method=''shift'', smooth=''spline'')
> bl2 <- baseini(time, conc, data=acth, puloc=pl1, method=''select'', smooth=''spline'')
> bl3 <- baseini(time, conc, data=acth, puloc=pl1, method=''shift'', smooth=''loess'')
> bl4 <- baseini(time, conc, data=acth, puloc=pl1, method=''select'', smooth=''loess'')
```

Above four fits are shown in Figure 1. Four different combinations of options *method* and *smooth* in *baseini* usually provide estimates with similar shapes except for a horizontal shift. Of course, other

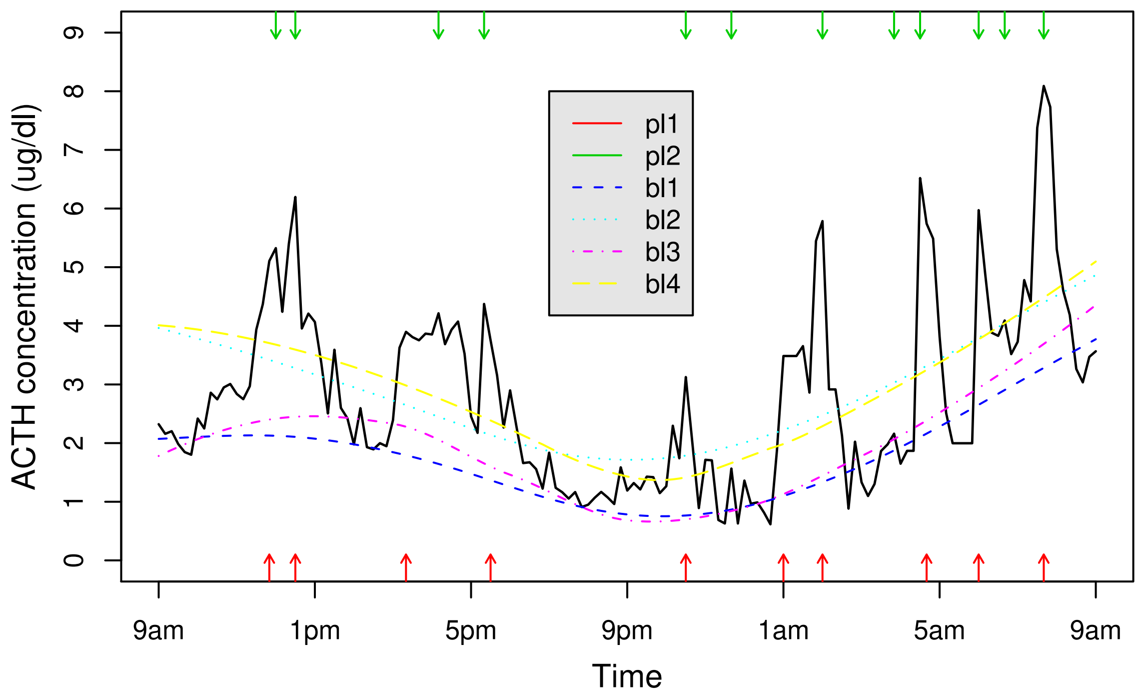

Figure 1: Points linked by the solid line are observations. Potential pulse locations identified by the change point method (pl1) and the CLUSTER method (pl2) are marked below and above. Four lines are the initial estimates of the baseline function.

methods such as wavelets may also be used to generate initial values for the baseline function.

To generate initial values for fixed parameters, we first fit a simple fixed effects model with common parameters for the infusion rate, decay rate and amplitude. That is, the following fit assumes that these three parameters are the same for all pulses:

```
> fix.ini <- pulest(time, conc, data=acth, baseline = 0, puloc=pl1,
    start = list(fixed = c(1,1,1), Inif = bl1), type = c("dblexp"),
    params=list(infrate="common", decrate="common", amplitude="common"),
    control =list(pul=list(TOLr=0.005, IDF=1, trace=F)))$coef$fixed
> fix.ini
       l        r        a
1.216217 0.752323 1.131563
```

Note that we fix the baseline to its initial values by setting *baseline=0*. By definition, all three parameters are positive. we used log transformations to relax positive constraints inside the program.

Now we are ready for the second step: pulse detection. As indicated in the help file that we have various options for each of the three parameters, infusion rate, decay rates and amplitudes. The infusion rates are usually difficult to estimate since there are very few observations providing information about them due to fast infusion relative to the sampling rate. Furthermore, the parameters of interest are usually the decay rates (or equivalently half-life) and amplitudes. To improve numerical stability, in the following we assume a common parameter for all infusion rates within a subject. We can use common, fixed and mixed parameters for the decay rates, and common, fixed, random and mixed for the amplitudes. Since our goal at the second step is to detect those among the initial locations which have amplitudes significantly different from zero, we first try a random effects model for the amplitudes which is equivalent to shrinking towards zero. We use a mixed effects model for the decay rates. Note we set *puloc=pl1*, the initial locations selected by the change point method. It is a good practice to check these locations before calling *puldet* to find obvious omissions and/or false locations. Locations identified by other methods may also be used.

```
> det.cmr <- puldet(time, conc, data=acth, type=c(``dblexp''),
    baseline=list(nb=~time, rk=cubic(time)),
    start=list(puloc=pl1, kmin=5, fixed=fix.ini[1:2], Inif=bl1),
    params=list(infrate=''common'', decrate=''mixed'', amplitude=''random''))
> det.cmr
> det.cmr
Pulses Detection with Semiparametric Model
Call:
puldet(x = time, y = conc, data = acth, baseline = list(nb = ~time,
    rk = cubic(time)), start = list(puloc = pl1, kmin = 5, fixed = fix.ini[1:2],
    Inif = bl1), type = c(``dblexp''), params = list(infrate = ``common'',
    decrate = ``mixed'', amplitude = ``random''))


Fitting Table :
        BIC      RIC      AIC      GCV DROP       DF
10  89.00715 107.6563 60.30416 67.14925    4 31.67091
```

```
9   89.98678 108.7905 61.04577 68.41823    5 31.93354
8   91.73921 109.1297 64.97329 74.10178    2 29.53354
7   92.73977 108.7171 68.14894 78.13252    6 27.13354
6   97.11936 111.6835 74.70363 86.69749    9 24.73354
5  105.38503 118.5359 85.14439 99.96884    3 22.33354

Initial location(s):2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667

Location(s) selected:
with AIC,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
with GCV,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
with BIC,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
with RIC,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
```

All four criteria select all 10 original locations. To avoid missing pulse locations, one may try to add more locations to the initial set of locations and run the *puldet* function again. At the second step, it seems that the above assumptions about three parameters are the most reasonable. However, it may have numerical problems of non-convergence. One may try other models when they are more appropriate and/or numerical problems occur. For example, one may assume a common parameter for all decay rates which can improve numerical stability. This assumption may not be appropriate. However, it is likely that the detection process based on testing the amplitude will be similar as before.

```
> det.ccr <- puldet(time, conc, data=acth, type=c("dblexp"),
    start=list(puloc=pl1, kmin=5, fixed=fix.ini[1:2], Inif = bl1),
    params=list(infrate="common", decrate="common", amplitude="random"))
> det.ccr
Pulses Detection with Semiparametric Model
Call:
puldet(x = time, y = conc, data = acth, start = list(puloc = pl1,
    kmin = 5, fixed = fix.ini[1:2], Inif = bl1), type = c(``dblexp''),
    params = list(infrate = ``common'', decrate = ``common'', amplitude = ``random''))


Fitting Table :
        BIC      RIC      AIC      GCV DROP       DF
10 100.5692 111.9318 73.19190 76.56787    3 22.33354
9   99.7097 110.4617 73.80337 77.28378    4 21.13354
8   99.4106 109.5521 74.97528 78.71158    5 19.93354
7  102.7404 112.2713 79.77605 84.85692    6 18.73354
6  104.8671 113.7875 83.37379 89.20123    1 17.53354
5  107.1052 115.4152 87.08293 93.51095    9 16.33354
```

```
Initial location(s):2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667

Location(s) selected:
with AIC,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
with GCV,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
with BIC,
8 pulse(s) :2.833 3.5 13.5 16 17 19.667 21 22.667
with RIC,
8 pulse(s) :2.833 3.5 13.5 16 17 19.667 21 22.667
```

The selections based on AIC and GCV remain the same. Two initial locations are dropped by the BIC and RIC criteria. For illustration, the following shows other models for the decay rates and amplitudes.

```
> det.ccm <- puldet(time, conc, data=acth, type=c(``dblexp''),
    start=list(puloc=pl1, kmin=5, fixed=fix.ini, Inif = bl1),
    params=list(infrate=''common'', decrate=''common'', amplitude=''mixed''))
> det.ccm
Pulses Detection with Semiparametric Model
Call:
puldet(x = time, y = conc, data = acth, start = list(puloc = pl1,
    kmin = 5, fixed = fix.ini, Inif = bl1), type = c(``dblexp''),
    params = list(infrate = ``common'', decrate = ``common'', amplitude = ``mixed''))


Fitting Table :
        BIC      RIC      AIC      GCV DROP       DF
10 106.1305 119.5626 76.29433 80.15510    3 23.53354
9  104.6417 117.3889 76.32693 80.06834    4 22.33354
8  103.8287 115.8910 77.03529 80.89589    5 21.13354
7  107.2104 118.5878 81.93836 87.31551    6 19.93354
6  108.9340 119.6265 85.18340 91.29112    1 18.73354
5  111.0646 121.0721 88.83532 95.62875    9 17.53354

Initial location(s):2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667

Location(s) selected:
with AIC,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
with GCV,
9 pulse(s) :2.833 3.5 8.5 13.5 16 17 19.667 21 22.667
with BIC,
8 pulse(s) :2.833 3.5 13.5 16 17 19.667 21 22.667
```

```
with RIC,
8 pulse(s) :2.833 3.5 13.5 16 17 19.667 21 22.667

> det.cfm <- puldet(time, conc, data=acth, type = c(``dblexp''),
    start=list(puloc=pl1, kmin=5, fixed=c(fix.ini[c(1,3)],
               rep(fix.ini[2],length(pl1))), Inif=bl1),
    params=list(infrate=''common'', decrate=''fixed'', amplitude=''mixed''))
> det.cfm
Pulses Detection with Semiparametric Model
Call:
puldet(x = time, y = conc, data = acth, start = list(puloc = pl1,
    kmin = 5, fixed = c(fix.ini[c(1, 3)], rep(fix.ini[2], length(pl1))),
    Inif = bl1), type = c(``dblexp''), params = list(infrate = ``common'',
    decrate = ``fixed'', amplitude = ``mixed''))


Fitting Table :
         BIC       RIC      AIC       GCV DROP       DF
10  74.94987  82.38278 54.71231  56.04081    4 20.80137
9   77.98348  85.96388 56.25528  58.20606    5 22.33354
8   81.45872  89.01032 60.89800  64.52073    6 21.13354
7   87.60491  94.72772 68.21166  74.17370    9 19.93354
6   97.31614 104.01016 79.09037  88.15119    3 18.73354
5  110.58350 116.84873 93.52520 106.19339    2 17.53354

Initial location(s):2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667

Location(s) selected:
with AIC,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
with GCV,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
with BIC,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
with RIC,
10 pulse(s) :2.833 3.5 6.333 8.5 13.5 16 17 19.667 21 22.667
```

Note that the initial values for common parameters appear before those for fixed parameters. Thus, in *det.cfm*, we set *fixed=c(fix.ini[c(1,3)],rep(fix.ini[2],length(pl1)))* where initial values for the common mean of amplitude, *fix.ini[3]*, appears before the initial values for the fixed effects of decay rates, *rep(fix.ini[2],length(pl1))*. *puldet* returns a puldet object from which various information about the selection process and the final model can be extracted. For example, to extract the selected locations according to the BIC criterion from *det.cmr*, we can use

```
pl5 <- det.cmr$detloc$BIC$puloc
```

The final step is to fit the final model and derive estimates of the parameters. As in *puldet*, we have various options for each of the three parameters, infusion rate, decay rates and amplitudes. As discussed above, we assume a common parameter for the infusion rate. The most reasonable model for the acth data is perhaps to assume mixed effects for both decay rates and amplitudes. Note we set *puloc=pl5*, the selected locations according to the BIC criterion from *det.cmr*. It is a good practice to check out these locations before calling *pulest*. Locations selected by other criteria or other methods may also be used. Since the estimation does not involve a selection process, we set *IDF=1*.

```
> fit.cmm <- pulest(time, conc, data=acth, type=c("dblexp"),
    puloc=pl5, start=list(fixed=fix.ini, Inif=bl1),
    params=list(infrate="common", decrate="mixed", amplitude="mixed"),
    control=list(pul=list(IDF=1)))
> summary(fit.cmm)
Estimation for Hormone Pulses
Call:
pulest(x = time, y = conc, data = acth, puloc = pl5, start = list(fixed = fix.ini,
    Inif = bl1), type = c(``dblexp''), params = list(infrate = ``common'',
    decrate = ``mixed'', amplitude = ``mixed''), control=list(pul=list(IDF=1)))


Fitting Table :
 NP      BIC      RIC      AIC      GCV       DF df(base)     nlaht   N
 10 84.96242 99.74505 59.00027 63.93854 28.10171 5.101708 -2.087880 145
      RSS
 41.55689
Pulse locations :2.833333 3.5 6.333333 8.5 13.5 16 17 19.66667 21 22.66667

Estimation at the selected model:

Parametric component

 Fixed effects:
      Value  Std.Error   t-value      p-value
l 1.2568761 0.07469437 16.826919 7.712539e-35
r 0.6840169 0.10888465  6.282032 4.350668e-09
a 1.1535645 0.07332225 15.732803 3.111616e-32

Random effects:
       A1        A2        A3        A4        A5        A6        A7        A8
0.3628441 0.3628441 0.3628441 0.3628441 0.3628441 0.3628441 0.3628441 0.3628441
       A9       A10        R1        R2        R3        R4        R5        R6
0.3628441 0.3628441 0.7540031 0.7540031 0.7540031 0.7540031 0.7540031 0.7540031
       R7        R8        R9       R10
0.7540031 0.7540031 0.7540031 0.7540031
```

```
Non-parametric
estimate of smoothing parameter : 5.633163e-05
Degrees of Freedom of the baseline (df(base)):  5.101708

Residual standard deviation: 0.557 on  133.898  degrees of freedom
```

The *summary* function provides summary of the fit and estimates of the parameters. Estimates can also be extracted directly from the fitted object which contains

```
> names(fit.cmm)
 [1] "tbl"        "coef"       "Var"        "sigma"      "baseFitted"
 [6] "nlmeFitted" "nlmeObj"    "gnlsObj"    "ssrObj"     "callxObj"
[11] "nlme"       "call"
```

The estimate of the baseline function and overall fit are shown in Figure 2.

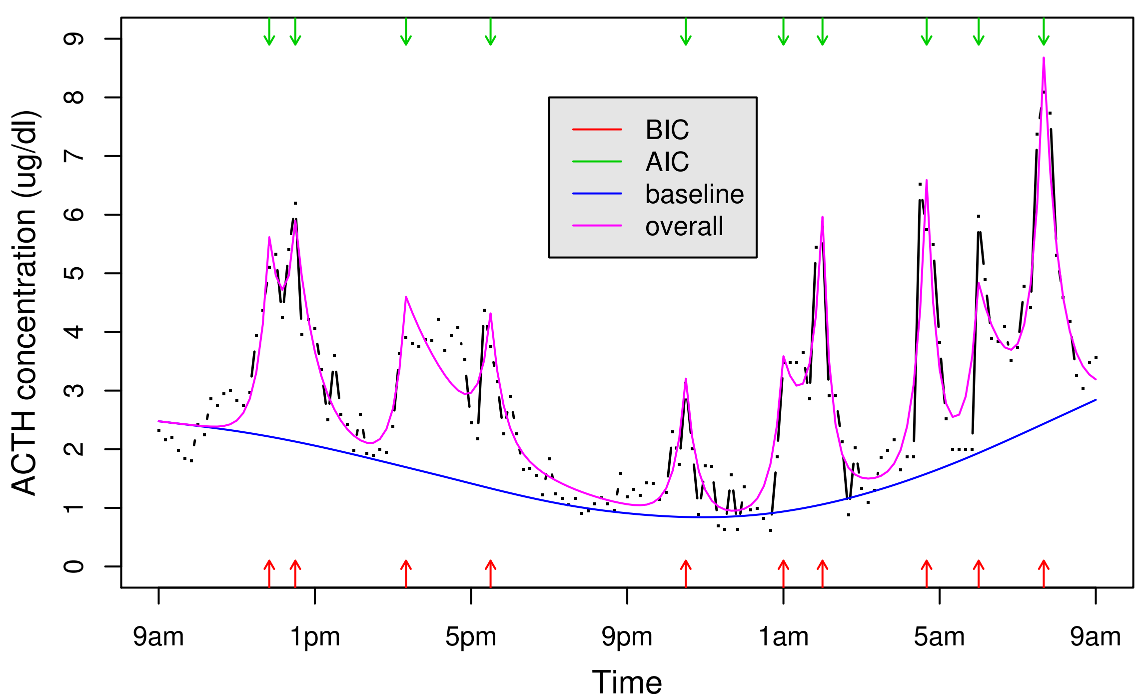

Figure 2: Points are observations. Final pulse locations identified by the BIC and AIC criteria are marked below and above. Two lines are the overall estimate and the estimate of the baseline function.

For illustration, the following shows fits with other options for the decay rates and amplitudes.

```
> fit.ccm <- pulest(time, conc, data=acth, type=c(``dblexp''),
    puloc=pl5, start=list(fixed=fix.ini, Inif=bl1),
    params=list(infrate=''common'', decrate=''common'', amplitude=''mixed''),
    control=list(pul=list(IDF=1)))
```

```
> summary(fit.ccm)
Estimation for Hormone Pulses
Call:
pulest(x = time, y = conc, data = acth, puloc = pl5, start = list(fixed = fix.ini,
    Inif = bl1), type = c(``dblexp''), params = list(infrate = ``common'',
    decrate = ``common'', amplitude = ``mixed''), control=list(pul=list(IDF=1)))

Fitting Table :
 NP      BIC      RIC      AIC      GCV       DF df(base) nlaht   N     RSS
 10 100.6194 110.4800 74.07961 76.83076 20.93354 7.933543    -3 145 56.2481
Pulse locations :2.833333 3.5 6.333333 8.5 13.5 16 17 19.66667 21 22.66667

Estimation at the selected model:

Parametric component

 Fixed effects:
      Value  Std.Error  t-value      p-value
l 1.0386123 0.08564869 12.12642 3.286025e-23
r 0.9350778 0.08793500 10.63374 1.848213e-19
a 1.1167663 0.08234202 13.56253 8.620011e-27

 Random effects:
       A1        A2        A3        A4        A5        A6        A7        A8
0.3822977 0.3822977 0.3822977 0.3822977 0.3822977 0.3822977 0.3822977 0.3822977
       A9       A10
0.3822977 0.3822977

Non-parametric
estimate of smoothing parameter : 6.896554e-06
Degrees of Freedom of the baseline (df(base)):  7.933543

Residual standard deviation: 0.653 on  132.066  degrees of freedom

> fit.cfm <- pulest(time, conc, data=acth, type=c(``dblexp''),
    puloc=pl5, start=list(fixed=c(fix.ini[c(1,3)],
    rep(fix.ini[2],length(pl5))), Inif=bl1),
    params=list(infrate=''common'', decrate=''fixed'', amplitude=''mixed''),
    control=list(pul=list(IDF=1)))
> summary(fit.cfm)
Estimation for Hormone Pulses
Call:
pulest(x = time, y = conc, data = acth, puloc = pl5, start = list(fixed = c(fix.ini[c(1,
```

```
    3)], rep(fix.ini[2], length(pl5))), Inif = bl1), type = c(``dblexp''),
    params = list(infrate = ``common'', decrate = ``fixed'', amplitude = ``mixed''),
    control=list(pul=list(IDF=1)))


Fitting Table :
 NP     BIC      RIC      AIC      GCV       DF df(base)     nlaht   N      RSS
 10 70.7208 75.63607 53.01278 53.76614 18.20137 5.201371 -2.129591 145 41.11516
Pulse locations :2.833333 3.5 6.333333 8.5 13.5 16 17 19.66667 21 22.66667

Estimation at the selected model:

Parametric component

 Fixed effects:
           Value  Std.Error     t-value      p-value
l     1.297105557 0.07851599 16.52027251 2.630730e-33
a     1.150700986 0.07941195 14.49027513 1.359278e-28
r1    0.651609595 0.54202436  1.20217770 2.315527e-01
r2    0.464464665 0.37256057  1.24668229 2.148303e-01
r3   -0.764964970 0.37386958 -2.04607438 4.283308e-02
r4    1.475199720 0.58009229  2.54304315 1.220107e-02
r5    1.502419363 0.49499332  3.03523160 2.921508e-03
r6   -0.004955199 0.44825206 -0.01105449 9.911975e-01
r7    1.856003727 0.38360441  4.83832743 3.757501e-06
r8    1.256037779 0.23346443  5.37999630 3.505155e-07
r9   -0.133089436 0.47199903 -0.28196972 7.784301e-01
r10   0.936508902 0.22183447  4.22165631 4.609339e-05

 Random effects:
       A1        A2        A3        A4        A5        A6        A7        A8
0.3375959 0.3375959 0.3375959 0.3375959 0.3375959 0.3375959 0.3375959 0.3375959
       A9       A10
0.3375959 0.3375959

Non-parametric
estimate of smoothing parameter : 5.117305e-05
Degrees of Freedom of the baseline (df(base)):  5.201371

Residual standard deviation: 0.572 on  125.799  degrees of freedom
```

4.2 Gonadotropin-releasing hormone

The data frame, *ewe*, consists two variables, *time* and *conc*, which represent gonadotropin-releasing hormones (GnRH) concentrations on log scale over a period of 24 hours from an ewe. Our first step is to detect potential pulse locations using *pulini*:

```
> pl1 <- pulini(time, conc, data=ewe, method=''pcp'', alpha=1)
> pl2 <- pulini(time, conc, data=ewe, method=''CLUSTER'', alpha=.4,
             control=list(cluster=list(sd=.01*mean(ewe$conc))))
> bl1 <- baseini(time, conc, data=ewe, puloc=pl2, method=''shift'', smooth=''spline'')
> bl2 <- baseini(time, conc, data=ewe, puloc=pl2, method=''select'', smooth=''spline'')
> bl3 <- baseini(time, conc, data=ewe, puloc=pl2, method=''shift'', smooth=''loess'')
> bl4 <- baseini(time, conc, data=ewe, puloc=pl2, method=''select'', smooth=''loess'')
> fix.ini <- pulest(time, conc, data=ewe, baseline=0, puloc=pl2,
    start = list(fixed = c(1,1,1), Inif = bl1), type = c(``dblexp''),
    params=list(infrate=''common'', decrate=''common'', amplitude=''common''),
    control =list(pul=list(TOLr=0.005, IDF=1, trace=F)))$coef$fixed
```

Figure 3 shows the identified locations and estimates of baseline function. Obviously the change point method missed some obvious locations even with setting *alpha=1*, possibly due to the fact that data are sparse. The CLUSTER method identifies all possible locations. The baseline function is almost a constant.

Next, we run *puldet* to detect pulse locations.

```
> ewe.det.cmr <- puldet(time, conc, data=ewe, type=c(``dblexp''),
    baseline=list(nb=~time, rk=cubic(time)),
    start=list(puloc=pl2, kmin=8, fixed=fix.ini[1:2], Inif=bl1),
    params=list(infrate=''common'', decrate=''mixed'', amplitude=''random''))
> summary(ewe.det.cmr)
Identification for Hormone Pulses
Call:
puldet(x = time, y = conc, data = ewe, baseline = list(nb = ~time,
    rk = cubic(time)), start = list(puloc = pl2, kmin = 8, fixed = fix.ini[1:2],
    Inif = bl1), type = c(``dblexp''), params = list(infrate = ``common'',
    decrate = ``mixed'', amplitude = ``random''))


Fitting Tables :
  NP      BIC       RIC      AIC      GCV       DF df(base)     nlaht  N
1 13 78.63712 116.05378 49.27110 93.87693 35.61741 2.017407 0.9999998 71
2 12 74.92880 109.82422 47.54154 82.39817 33.21741 2.017407 0.9999998 71
3 11 70.97206 103.34624 45.56357 72.13453 30.81741 2.017407 0.9999998 71
4 10 66.98645  96.83939 43.55673 63.51582 28.41741 2.017407 0.9999998 71
5  9 63.05515  90.38686 41.60419 56.41203 26.01741 2.017407 0.9999998 71
6  8 59.22121  84.03168 39.74902 50.60375 23.61741 2.017407 0.9999998 71
       RSS    Sigma2 DROP
```

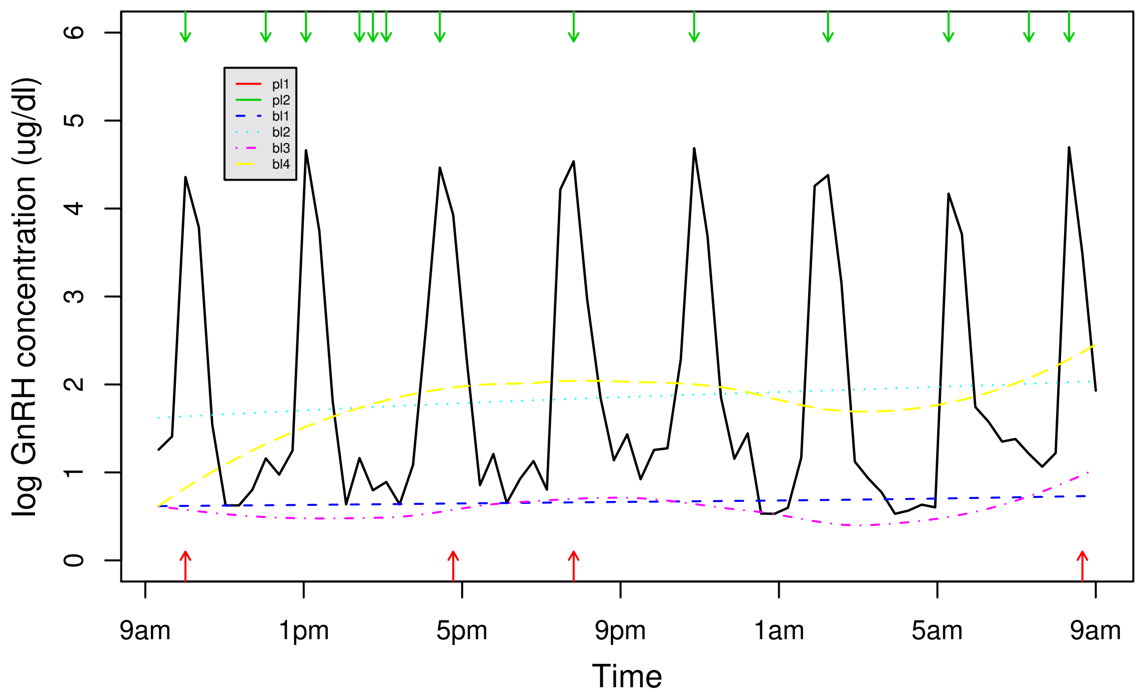

Figure 3: Points linked by the solid line are observations. Potential pulse locations identified by the change point method (pl1) and the CLUSTER method (pl2) are marked below and above. Four lines are the initial estimates of the baseline function.

```
1 23.31425 0.3643843    5
2 23.33374 0.3646889    4
3 23.10481 0.3611109    2
4 22.84701 0.3570817    6
5 22.64352 0.3539013   12
6 22.53739 0.3522425   11

Initial location(s):1.014 3.042 4.056 5.408 5.746 6.085 7.437 10.817 13.859 17.239 20.282 22.31
324

Location(s) selected:
with AIC,
8 pulse(s) :1.014 4.056 7.437 10.817 13.859 17.239 20.282 23.324
with GCV,
8 pulse(s) :1.014 4.056 7.437 10.817 13.859 17.239 20.282 23.324
with BIC,
8 pulse(s) :1.014 4.056 7.437 10.817 13.859 17.239 20.282 23.324
with RIC,
8 pulse(s) :1.014 4.056 7.437 10.817 13.859 17.239 20.282 23.324

> ewe.det.ccr <- puldet(time, conc, data=ewe, type=c(``dblexp''),
    start=list(puloc=pl2, kmin=8, fixed=fix.ini[1:2], Inif = bl1),
    params=list(infrate=''common'', decrate=''common'', amplitude=''random''))
> summary(ewe.det.ccr)
Identification for Hormone Pulses
Call:
puldet(x = time, y = conc, data = ewe, start = list(puloc = pl2,
    kmin = 8, fixed = fix.ini[1:2], Inif = bl1), type = c(``dblexp''),
    params = list(infrate = ``common'', decrate = ``common'', amplitude = ``random''))


Fitting Tables :
  NP      BIC      RIC      AIC      GCV       DF df(base)     nlaht  N
1 13 53.82201 66.22586 37.60384 45.12745 20.01741 2.017407 0.9999998 71
2 12 51.80621 63.46648 36.56029 42.73483 18.81741 2.017407 0.9999998 71
3 11 49.98383 60.90052 35.71015 40.85148 17.61741 2.017407 0.9999998 71
4 10 47.71261 57.88571 34.41117 38.33116 16.41741 2.017407 0.9999998 71
5  9 45.87203 55.30155 33.54284 36.68521 15.21741 2.017407 0.9999998 71
6  8 43.92981 52.61574 32.57286 34.98466 14.01741 2.017407 0.9999998 71
       RSS    Sigma2 DROP
1 23.26848 0.3580725    5
2 23.08430 0.3552381    4
3 23.09353 0.3553803    2
4 22.65393 0.3486154    6
```

```
5 22.64497 0.3484774   12
6 22.53437 0.3467754   11

Initial location(s):1.014 3.042 4.056 5.408 5.746 6.085 7.437 10.817 13.859 17.239 20.282 22.31
324

Location(s) selected:
with AIC,
8 pulse(s) :1.014 4.056 7.437 10.817 13.859 17.239 20.282 23.324
with GCV,
8 pulse(s) :1.014 4.056 7.437 10.817 13.859 17.239 20.282 23.324
with BIC,
8 pulse(s) :1.014 4.056 7.437 10.817 13.859 17.239 20.282 23.324
with RIC,
8 pulse(s) :1.014 4.056 7.437 10.817 13.859 17.239 20.282 23.324
```

All criteria with both models detected the same eight pulse locations which are obvious from Figure 3. We then fit the final model.

```
> pl5 <- ewe.det.cmr$detloc$BIC$puloc
> ewe.fit.cmm <- pulest(time, conc, data=ewe, type=c(``dblexp''),
    puloc=pl5, start=list(fixed=fix.ini, Inif=bl1),
    params=list(infrate=''common'', decrate=''mixed'', amplitude=''mixed''),
    control=list(pul=list(IDF=1)))
> summary(ewe.fit.cmm)
Estimation for Hormone Pulses
Call:
pulest(x = time, y = conc, data = ewe, puloc = pl5, start = list(fixed = fix.ini,
    Inif = bl1), type = c(``dblexp''), params = list(infrate = ``common'',
    decrate = ``mixed'', amplitude = ``mixed''), control=list(pul=list(IDF=1)))

Fitting Table :
 NP      BIC      RIC      AIC      GCV       DF df(base)      nlaht  N
  8 56.58334 71.64595 38.61334 47.23938 21.75055 2.750551 -0.9076687 71
      RSS
 22.72952
Pulse locations :1.014085 4.056338 7.43662 10.81690 13.85915 17.23944 20.28169 23.32394

Estimation at the selected model:

Parametric component

 Fixed effects:
      Value Std.Error   t-value      p-value
l 1.1213152 0.1998733  5.610129 4.970360e-07
```

```
r 0.5190614 0.1914955  2.710568 8.668375e-03
a 1.4084899 0.0681754 20.659797 1.470847e-29

 Random effects:
          A1           A2           A3           A4           A5           A6
3.067750e-05 3.067750e-05 3.067750e-05 3.067750e-05 3.067750e-05 3.067750e-05
          A7           A8           R1           R2           R3           R4
3.067750e-05 3.067750e-05 2.583084e-06 2.583084e-06 2.583084e-06 2.583084e-06
          R5           R6           R7           R8
2.583084e-06 2.583084e-06 2.583084e-06 2.583084e-06

Non-parametric
estimate of smoothing parameter : 0.001742099
Degrees of Freedom of the baseline (df(base)):  2.750551

Residual standard deviation: 0.604 on  62.249  degrees of freedom
```

The estimate of the baseline function and overall fit are shown in Figure 4.

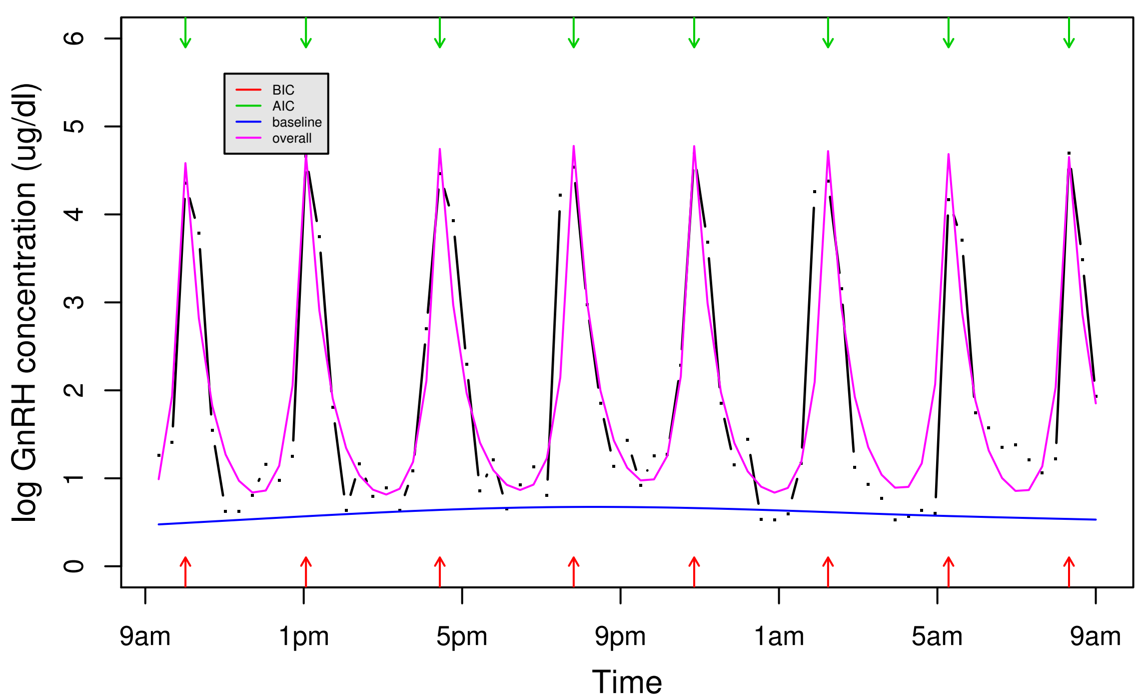

Figure 4: Points are observations. Final pulse locations identified by the BIC and AIC criteria are marked below and above. Two lines are the overall estimate and the estimate of the baseline function.

The baseline function is close to a constant. To fit a model with a constant baseline, we use linear spline in the *baseline* argument and fix the smoothing parameter to a large number by setting *ssr=list(limnla=10)* in the *control* argument. A large smoothing parameter forces the baseline function to be in the null space of the linear spline model which is a constant.

```
> ewe.fit.cmm.cb <- pulest(time, conc, data=ewe, type=c(``dblexp''),
    baseline=list(nb=~1, rk=linear(time)),
    puloc=pl5, start=list(fixed=fix.ini, Inif=bl1),
    params=list(infrate=''common'', decrate=''mixed'', amplitude=''mixed''),
    control=list(ssr=list(limnla=10), pul=list(IDF=1)))
> summary(ewe.fit.cmm.cb)
Estimation for Hormone Pulses
Call:
pulest(x = time, y = conc, data = ewe, baseline = list(nb = ~1,
    rk = linear(time)), puloc = pl5, start = list(fixed = fix.ini,
    Inif = bl1), type = c(``dblexp''), params = list(infrate = ``common'',
    decrate = ``mixed'', amplitude = ``mixed''), control = list(ssr = list(limnla = 10),
    pul = list(IDF = 1)))


Fitting Table :
 NP      BIC      RIC      AIC      GCV DF df(base) nlaht  N      RSS
  8 54.69803 67.36923 38.11362 45.45726 20        1    10 71 23.45454
Pulse locations :1.014085 4.056338 7.43662 10.81690 13.85915 17.23944 20.28169 23.32394

Estimation at the selected model:

Parametric component

 Fixed effects:
      Value  Std.Error   t-value      p-value
l 1.1236035 0.19999304  5.618213 4.503258e-07
r 0.5154661 0.19139737  2.693172 9.025276e-03
a 1.4058437 0.06803503 20.663528 5.130725e-30

 Random effects:
          A1           A2           A3           A4           A5           A6
4.176418e-05 4.176418e-05 4.176418e-05 4.176418e-05 4.176418e-05 4.176418e-05
          A7           A8           R1           R2           R3           R4
4.176418e-05 4.176418e-05 1.521971e-04 1.521971e-04 1.521971e-04 1.521971e-04
          R5           R6           R7           R8
1.521971e-04 1.521971e-04 1.521971e-04 1.521971e-04

Non-parametric
```

```
estimate of smoothing parameter : 140845070
Degrees of Freedom of the baseline (df(base)):  1

Residual standard deviation: 0.605 on  64  degrees of freedom
```

For illustration, one can fit a zero baseline function as follows.

```
> ewe.fit.cmm.zero <- pulest(time, conc, data=ewe, type=c(``dblexp''),
    baseline=0, puloc=pl5, start=list(fixed=fix.ini, Inif=0),
    params=list(infrate=''common'', decrate=''mixed'', amplitude=''mixed''),
    control=list(pul=list(IDF=1)))
> summary(ewe.fit.cmm.zero)
Estimation for Hormone Pulses
Call:
pulest(x = time, y = conc, data = ewe, baseline = 0, puloc = pl5,
    start = list(fixed = fix.ini, Inif = 0), type = c(``dblexp''),
    params = list(infrate = ``common'', decrate = ``mixed'', amplitude = ``mixed''),
    control = list(pul = list(IDF = 1)))


Fitting Table :
 NP     BIC      RIC      AIC      GCV DF df(base) nlaht  N     RSS
  8 49.6861 60.20175 35.05470 41.24131 19        0    NA 71 22.1219
Pulse locations :1.014085 4.056338 7.43662 10.81690 13.85915 17.23944 20.28169 23.32394

Estimation at the selected model:

Parametric component

 Fixed effects:
      Value  Std.Error   t-value      p-value
l 0.8217700 0.09294406  8.841555 9.422517e-13
r 0.2036410 0.10623990  1.916803 5.966083e-02
a 1.5165695 0.04250688 35.678212 2.064645e-44

 Random effects:
          A1           A2           A3           A4           A5           A6
0.0000710847 0.0000710847 0.0000710847 0.0000710847 0.0000710847 0.0000710847
          A7           A8           R1           R2           R3           R4
0.0000710847 0.0000710847 0.0686135181 0.0686135181 0.0686135181 0.0686135181
          R5           R6           R7           R8
0.0686135181 0.0686135181 0.0686135181 0.0686135181

Non-parametric
estimate of smoothing parameter : NA
```

```
Degrees of Freedom of the baseline (df(base)):  0

Residual standard deviation: 0.583 on  65  degrees of freedom
```

References

Merriam, G. R. and Wachter, K. W. (1982). Algorithms for the study of episodic hormone secretion, *American Journal of Physiology* **243**: E310–E318.

Van Cauter, E. L., L'Hermite, M., Copinschi, G., Refetoff, S., Desir, D. and Robyn, C. (1981). Quantitative analysis of spontaneous variations in plasa prolactin in normal man, *American Journal of Physiology* **241**: E355–E363.

Yang, Y., Liu, A. and Wang, Y. (2004). Detecting pulsatile hormone secretions using nonlinear mixed effects partial spline models, Technical Report No. ???. Department of Statistics and Applied Probability, UCSB. Available at `http://www.pstat.ucsb.edu/faculty/yuedong/research`.