Performing Detections Using *detect_peaks*

In this document we will perform event detections using `detect_peaks()` from the tagtools package (https://github.com/stacyderuiter/TagTools). The data that will be used in this document was obtained from a DTAG3 deployment on a Blainville's beaked whale in the Canary Islands in 2017. This data is freely available at the tagtools project website (https://github.com/stacyderuiter/TagTools).

First, I will load in the dataset after downloading the 'nc' file from the tagtools project website. In R, this would look like:

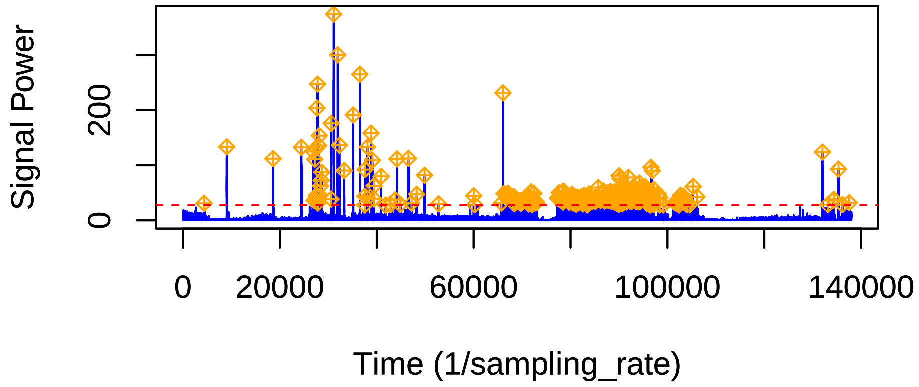
```
md <- load_nc('testset1.nc')
```

In Matlab or Octave, this would look like:

```
md = load_nc('testset1.nc')
```

The first event detection that I will perform will use the norm-jerk signal to detect times of large changes in the animal's acceleration.

```
detections <- detect_peaks(data = md$A$data, sr = md$A$sampling_rate, FUN = njerk, thresh = NULL,
                           bktime = NULL, plot_peaks = TRUE, sampling_rate = md$A$sampling_rate)
```



This same event detection can performed as follows.

```
jerk <- njerk(A = md$A)
detections <- detect_peaks(data = jerk, sr = md$A$sampling_rate, FUN = NULL, thresh = NULL,
                           bktime = NULL, plot_peaks = TRUE)
```

If using Matlab or Octave, these two forms of the event detection would be written as:

```
detections = detect_peaks(md.A.data, md.A.samping_rate, 'njerk', [], [], true, fs);
```

… or…

```
jerk = njerk(md.A);
detections = detect_peaks(jerk, md.A.sampling_rate, [], [], [], true);
```

A summary output of the detections made is shown below.

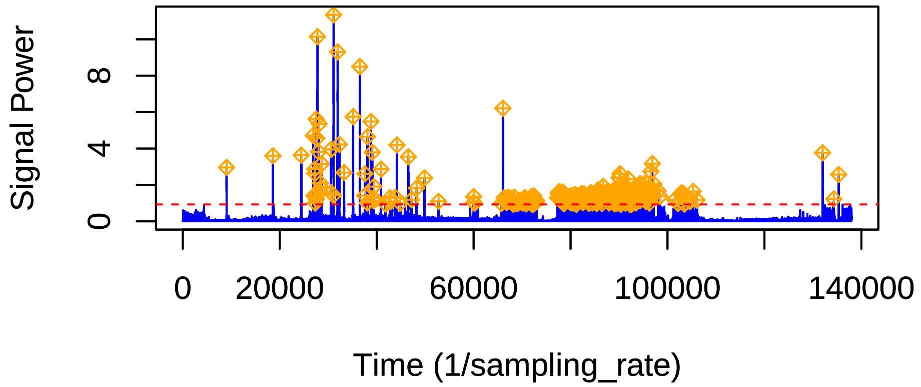
```
tibble::glimpse(detections)
```

```
## List of 6
##  $ start_time: num [1:279] 4376 9033 18598 24437 26911 ...
##  $ end_time  : num [1:279] 4379 9045 18620 24458 26929 ...
##  $ peak_time : num [1:279, 1] 4376 9038 18602 24439 26916 ...
##  $ peak_max  : num [1:279, 1] 30.9 133.6 112 132.6 125.2 ...
##  $ thresh    : Named num 27.4
##   ..- attr(*, "names")= chr "99%"
##  $ bktime    : Named num 0.44
##   ..- attr(*, "names")= chr "80%"
```

At this stage, it is possible to manually adjust the threshold and/or blanking time. If you choose to select a new threshold or blanking time, the detections will simply be rerun and the contents of `detections` will be changed appropriately.

The `detect_peaks()` function works with any uivariate input, not just norm-jerk. For example, suppose we want to find the times of high relative energy expenditure. To do this, we will use the tagtools `odba()` function, which includes several methods and options for calculating the overall dynamic body acceleration (ODBA), as a rough proxy for energy expenditure. For the sake of this document, I will use the 'fir' method for `odba()` and will set the high-pass filter cut-off frequency at 2 Hz.

```
detections <- detect_peaks(data = md$A$data, sr = md$A$sampling_rate, FUN = odba, thresh = NULL,
                           bktime = NULL, plot_peaks = TRUE,
                           sampling_rate = md$A$sampling_rate, fh = 2)
```



... or alternatively:

```
energy <- odba(A = md$A, fh = 2)
detections <- detect_peaks(data = energy, sr = md$A$sampling_rate, FUN = NULL, thresh = NULL,
                            bktime = NULL, plot_peaks = TRUE)
```

If using Matlab of Octave, these two forms of the event detection would be written as:

```
detections = detect_peaks(md.A.data, md.A.sampling_rate, 'odba', [], [], true, md.A.sampling_rate, 2);
```

... or...

```
energy = odba(md.A, 2);
detections = detect_peaks(energy, md.A.sampling_rate, [], [], [], true);
```

A summary output of the detections made is shown below.

```
tibble::glimpse(detections)
```

```
## List of 6
##  $ start_time: num [1:315] 9033 18597 24438 26912 27062 ...
##  $ end_time  : num [1:315] 9045 18620 24459 26929 27064 ...
##  $ peak_time : num [1:315, 1] 9039 18602 24440 26917 27062 ...
##  $ peak_max  : num [1:315, 1] 2.96 3.6 3.63 4.71 1.41 ...
##  $ thresh    : Named num 0.929
##   ..- attr(*, "names")= chr "99%"
##  $ bktime    : Named num 0.36
##   ..- attr(*, "names")= chr "80%"
```