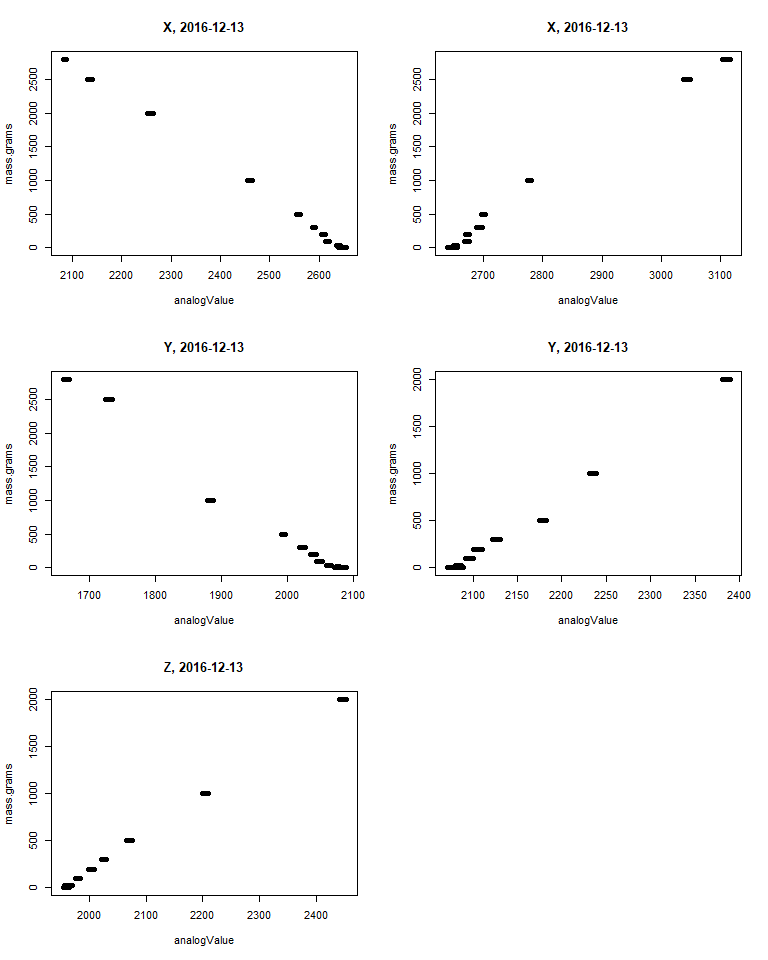
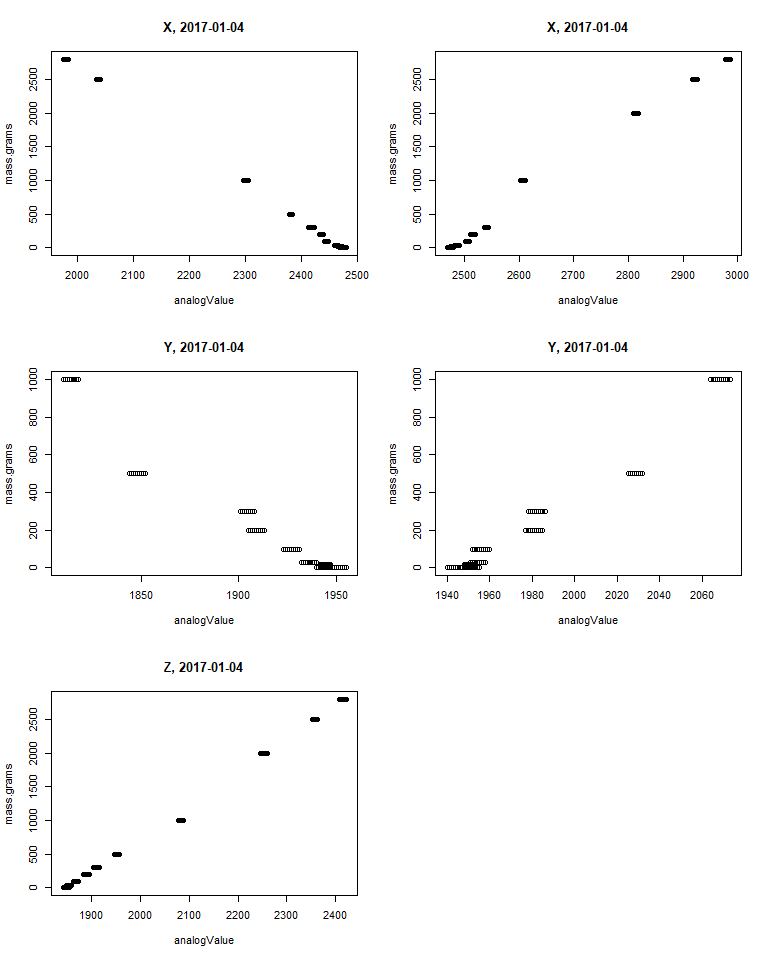
Limpet force meter analysis code

The calibration data for the positive Y-axis on 2016-12-14 appear to be no good in the summary calibration file. The connector for this transducer axis was not functional. In order to still use the data from the 2016-12-15 trial, we can instead use the same calibration from 2016-12-13, which is the closest previous calibration. For some of the other calibration events, data for certain masses appear to be erroneous (possibly due to the wrong mass value being entered during calibration). The code removes potentially spurious calibration data points, and the calibration data that end up being used in the analysis are plotted below.

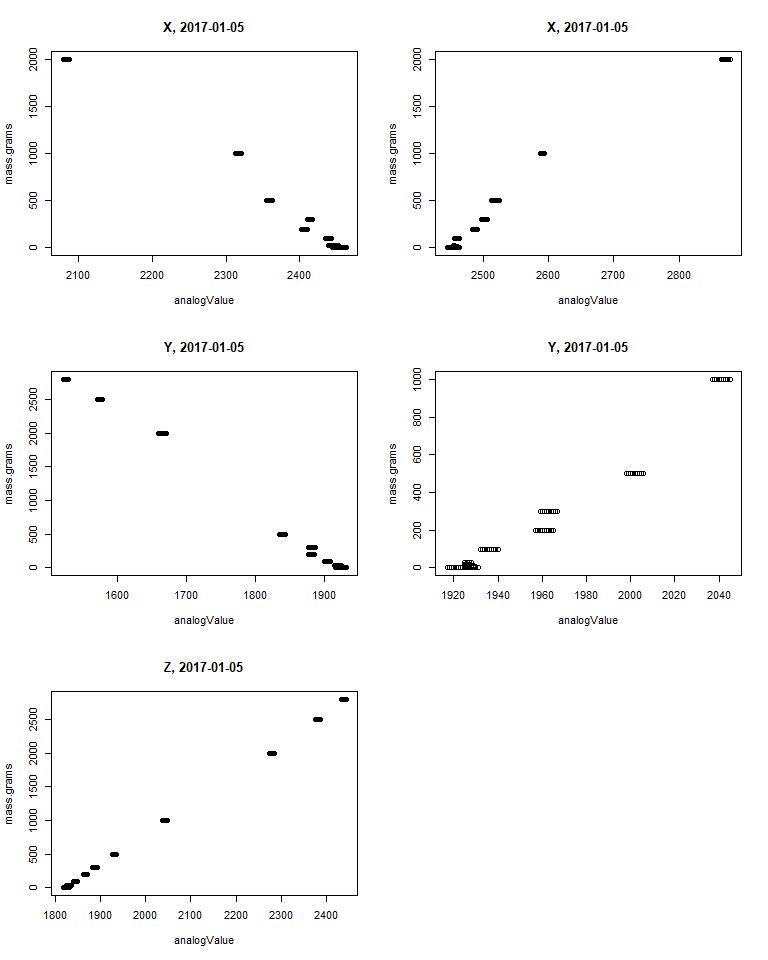
Note that the April 2016 trial data should not be analyzed, as the z-axis was not working. For the May 2016 trials, the closest available calibration data were from Apr 20 2016, so that calibration is used. The raw data from that calibration aren’t available currently, so we have to use the summary calibration data (means of the analog output value at each mass).



Calibration data from 2016-12-13



Calibration data from 2016-01-04



Calibration data from 2017-01-05

## [1] "CAL files raw data"   
## [2] "CalibrationFiles\_Apr202016.csv"   
## [3] "CalibrationFiles\_Dec132016.csv"   
## [4] "CalibrationFiles\_Dec142016.csv"   
## [5] "CalibrationFiles\_Jan42017.csv"   
## [6] "CalibrationFiles\_Jan52017.csv"   
## [7] "Events\_classified\_20180122.csv"   
## [8] "Events\_classified\_20180213.csv"   
## [9] "Events\_picker\_output\_20180213.csv"   
## [10] "ForceMeterData\_Apr262016.csv"   
## [11] "ForceMeterData\_Dec142016.csv"   
## [12] "ForceMeterData\_Dec152016.csv"   
## [13] "ForceMeterData\_Jan52017.csv"   
## [14] "ForceMeterData\_Jan62017.csv"   
## [15] "ForceMeterData\_May182016.csv"   
## [16] "ForceMeterData\_May192016.csv"   
## [17] "Pound\_ForceData\_Final.csv"   
## [18] "Pound\_ForceData\_withXYZ\_forces.csv"

The relevant calibration file for the May 2016 trials is the Apr202016 file.

During the import process, the calibration files and force trial files are concatenated together into two data frames, calibs and forces, each with a Date column that can be used to separate different days. The data frame events contains the identified peck and push events, with dates and millisecond timestamps.

The goal is to go through each identified event and extract the X, Y, and Z-axis forces (which need to be estimated based on the associated calibration values). The events data frame currently only has the net euclidean norm (total for all 3 axes).

## DEFINTIONS

* A given force can only be assigned to one category (cannot be a push and peck)
* Forces discussed below refer to Euclidean forces (norm)

### Peck:

* Magnitude: ≥ 2N and MUST BE greater than or equal to two times the intensity of the force data points immediately before AND after a given point.
* Duration: Force MUST BE sustained for ≤ 20 consecutive milliseconds
* Both magnitude and duration conditions must be met in order for a force to be categorized as a peck

### Push:

* Magnitude: ≥ 2N and may or may not be greater than or equal to two times the intensity of the force before and after a given force data point
* Duration: May be sustained for any amount of time
* Forces categorized as pushes do not meet conditions of Peck and Touch

### Touch:

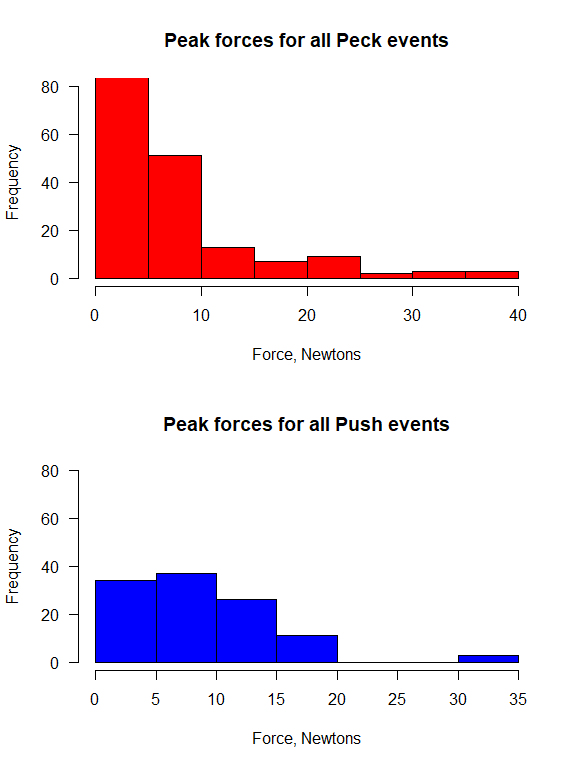
* Magnitude: < 2N (e.g. the bird seems to be playing with the ‘limpet’ mimic rather than actively attempting to remove it)
* We are not considering ‘Touches’ in our force meter analyses

I wrote an interactive function (PushPeckPickFunction) to let the user pick a trial date and go through the raw time series data and attempt to identify potential pecks and pushes. The script has the user first identify a period of at least 8 samples (80 milliseconds) where the baseline signal appears stable, and then the user clicks on each peak above the 2N threshold outlined above. Because the sensors tended to drift and take new baseline values, particularly after big hits by the bird, the user is identifying a new baseline for each peak or group of peaks that happen in quick succession and appear to share a common baseline value. The output of the function peakChooser() is a data frame that contains a row from the original forces data frame for each identified peak. As a result, the time in milliseconds (Time.msec), the raw X, Y, Z values (X.N,Y.N,Z.N, units of Newtons), calibrated forces in the X, Y, and Z axes (X.N.off, Y.N.off,Z.N.off, units of Newtons), the euclidean norm of all 3 axes (Norm, units of Newtons), and the timestamp of the chosen baseline value (BaselineTime.msec) are available for each peak event. These data could be used to relocate a chunk of the raw timeseries around each peak event for further analysis.

# Classifying peaks vs. pushes

The output from the peakChooser function was saved to a csv file called Events\_picker\_output\_20180213.csv. The data in that file next need to be processed to determine if each event was a peck or a push according to the criteria outlined above.

The code defines a function peakPushAlgo to classify each peak event as either a push or peck, when an sample force greater than 2 Newtons was identified. The function peckpushClassify then cycles through all identified events and classifies them. The code in peakpushClassify also determines the duration of each event, based on the time the force spent above the 2 Newton threshold during that event. Because the raw data were sampled at 10 millisecond intervals, a single peak force above 2 Newtons (peck) will have a duration of 20 ms. That is the time it takes to go from ~0 N to >2 N and back to ~0 N. For push events, the duration should end up longer than 20 ms. The output of peakpushClassify is a data frame similar to the input events data frame, but with four new columns added on, PeckPush, Duration.msec, StartPeak.msec, and EndPeak.msec. The latter two columns list the time stamp in milliseconds where the algorithm declared the start and end of a given event (the force exceeding the 2 Newton threshold). If an event in the events data frame does not meet the critera for a peck or a push, these new columns will contain NA.



## Summary data for all identified events. Values are force in Newtons.

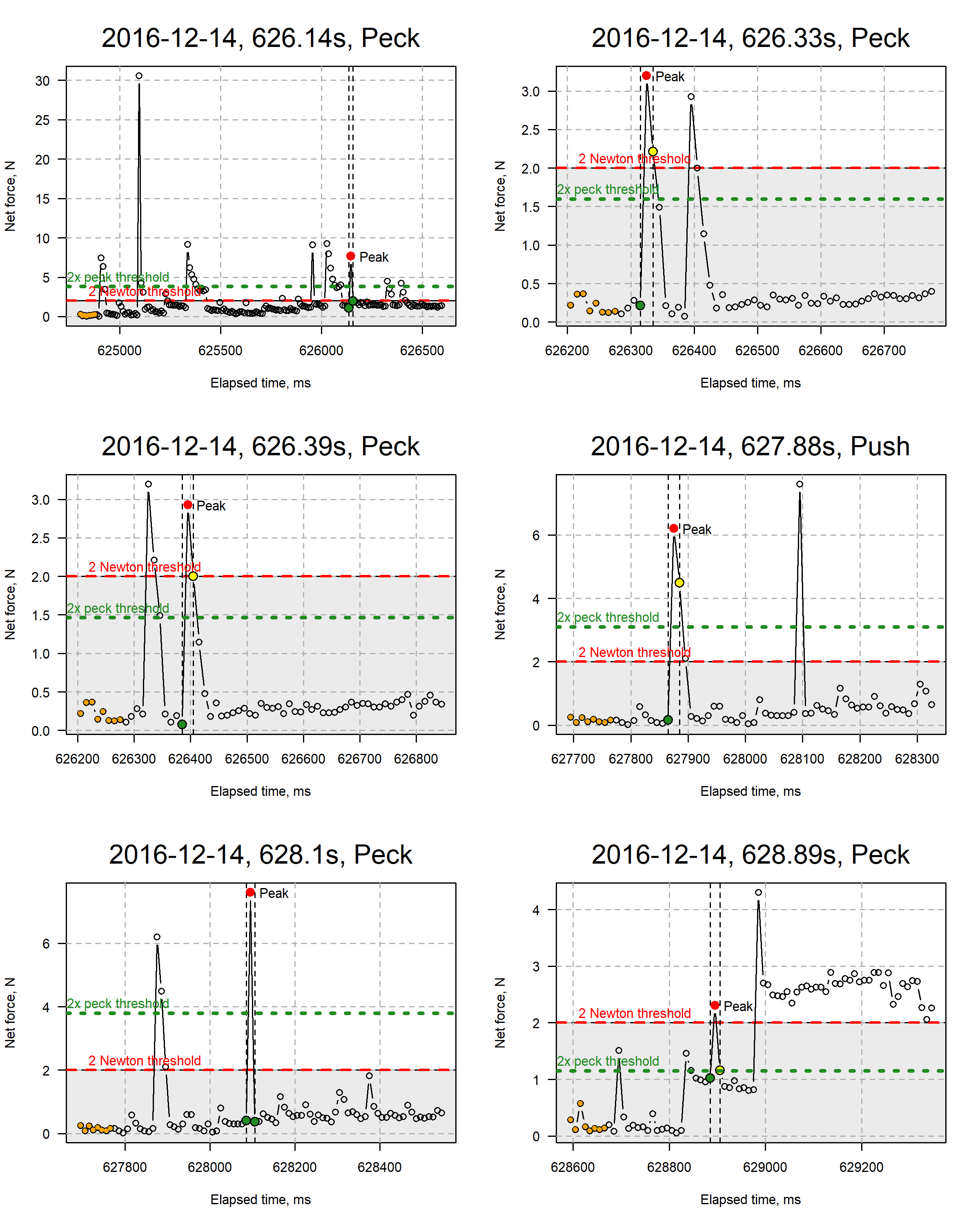
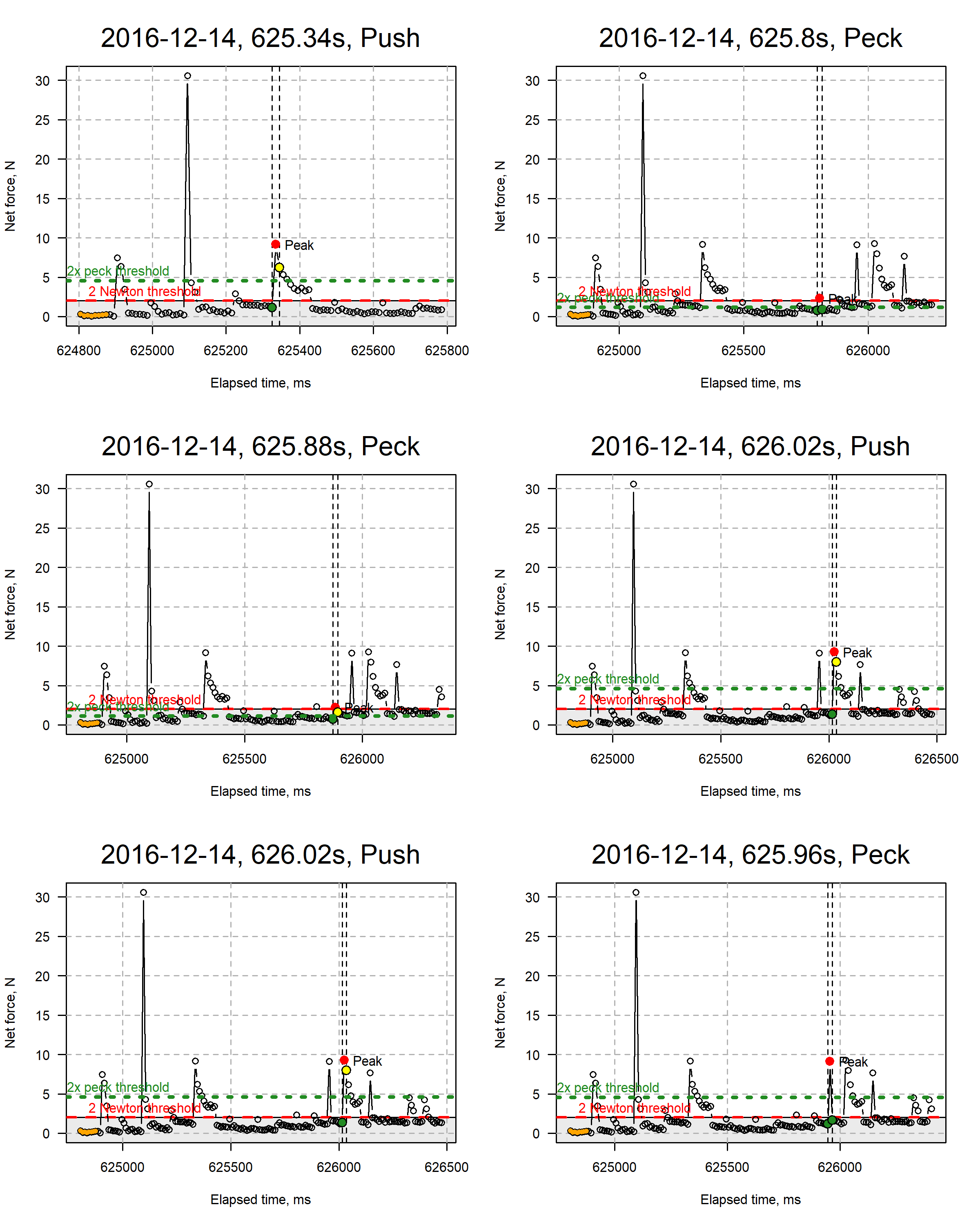
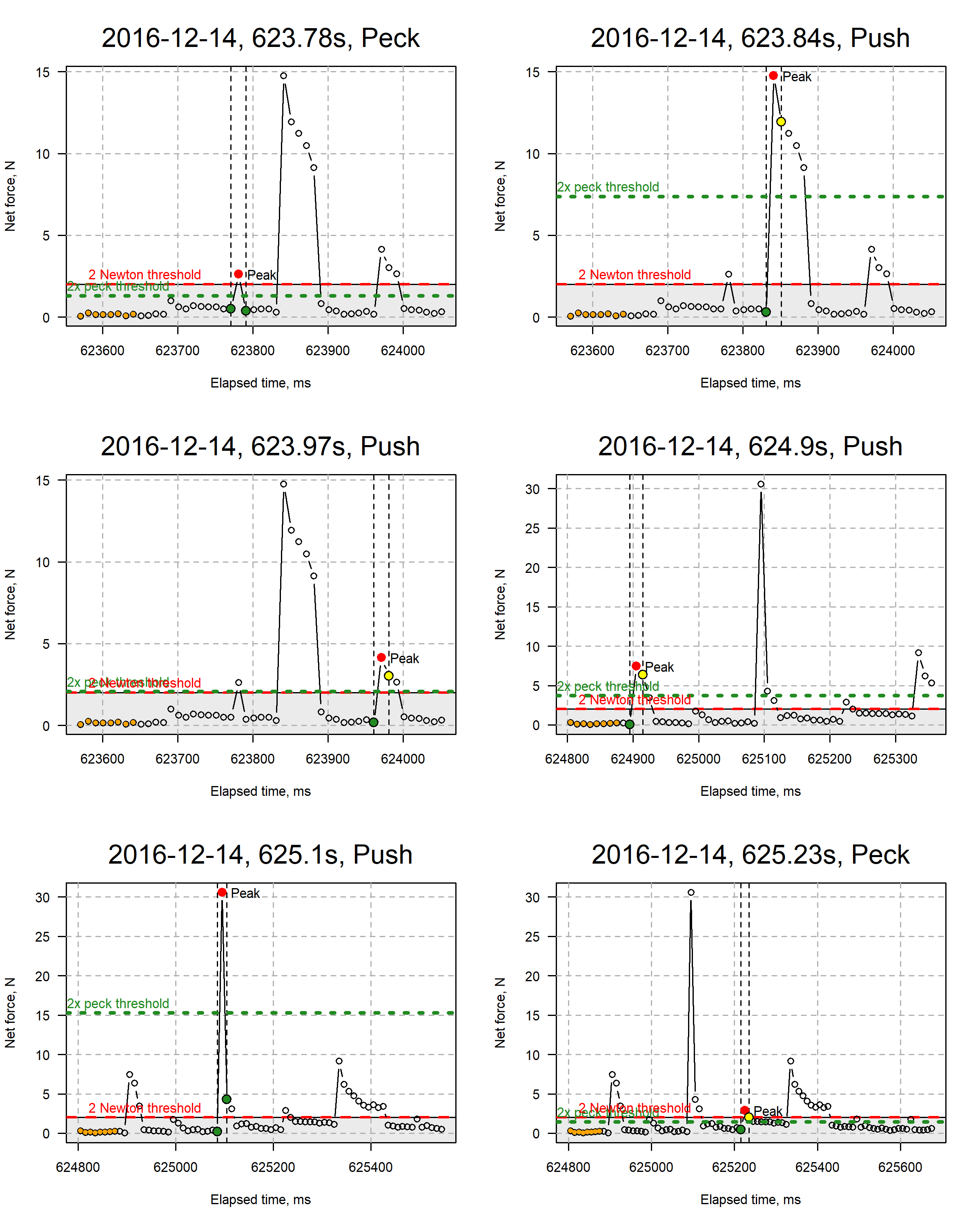
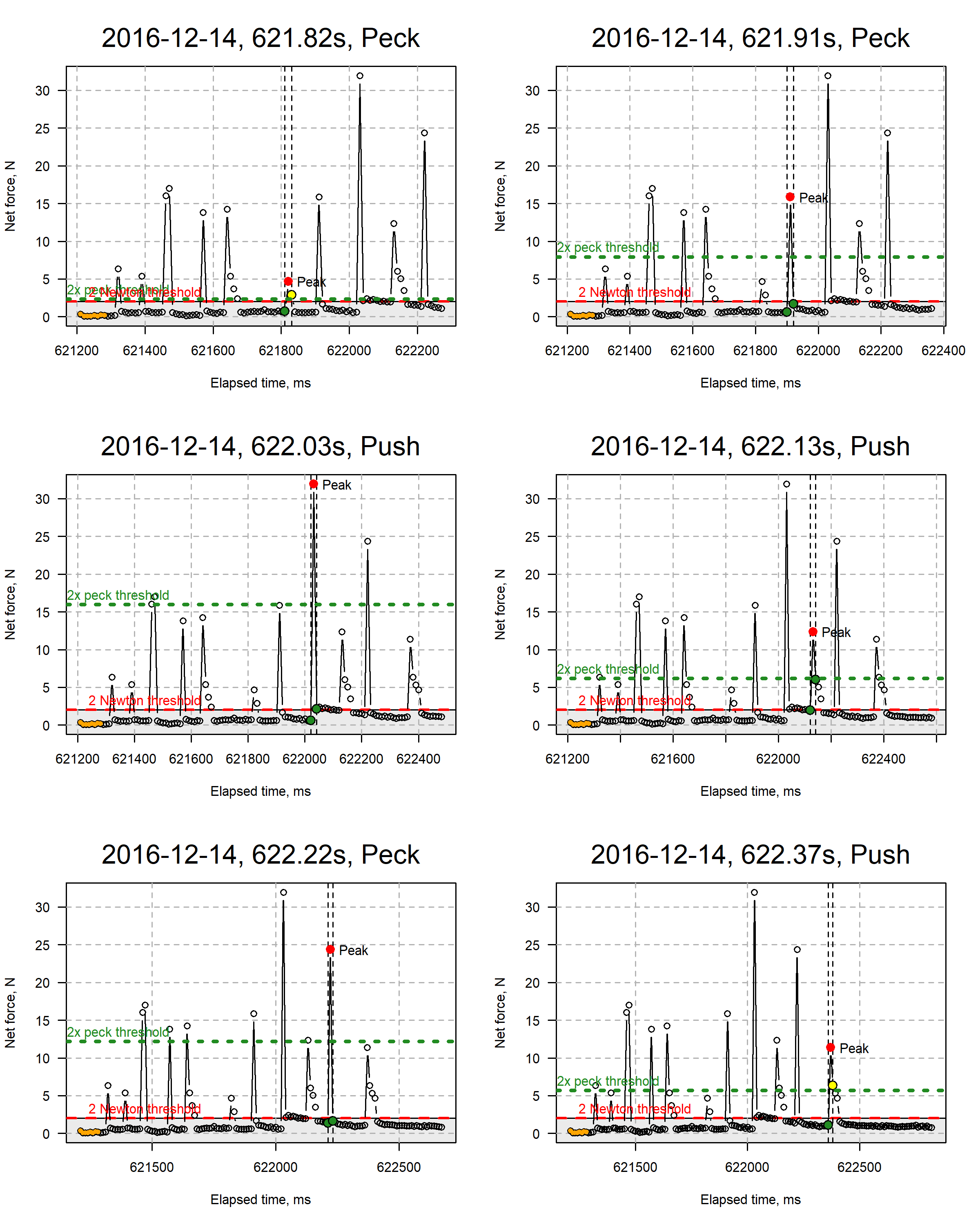
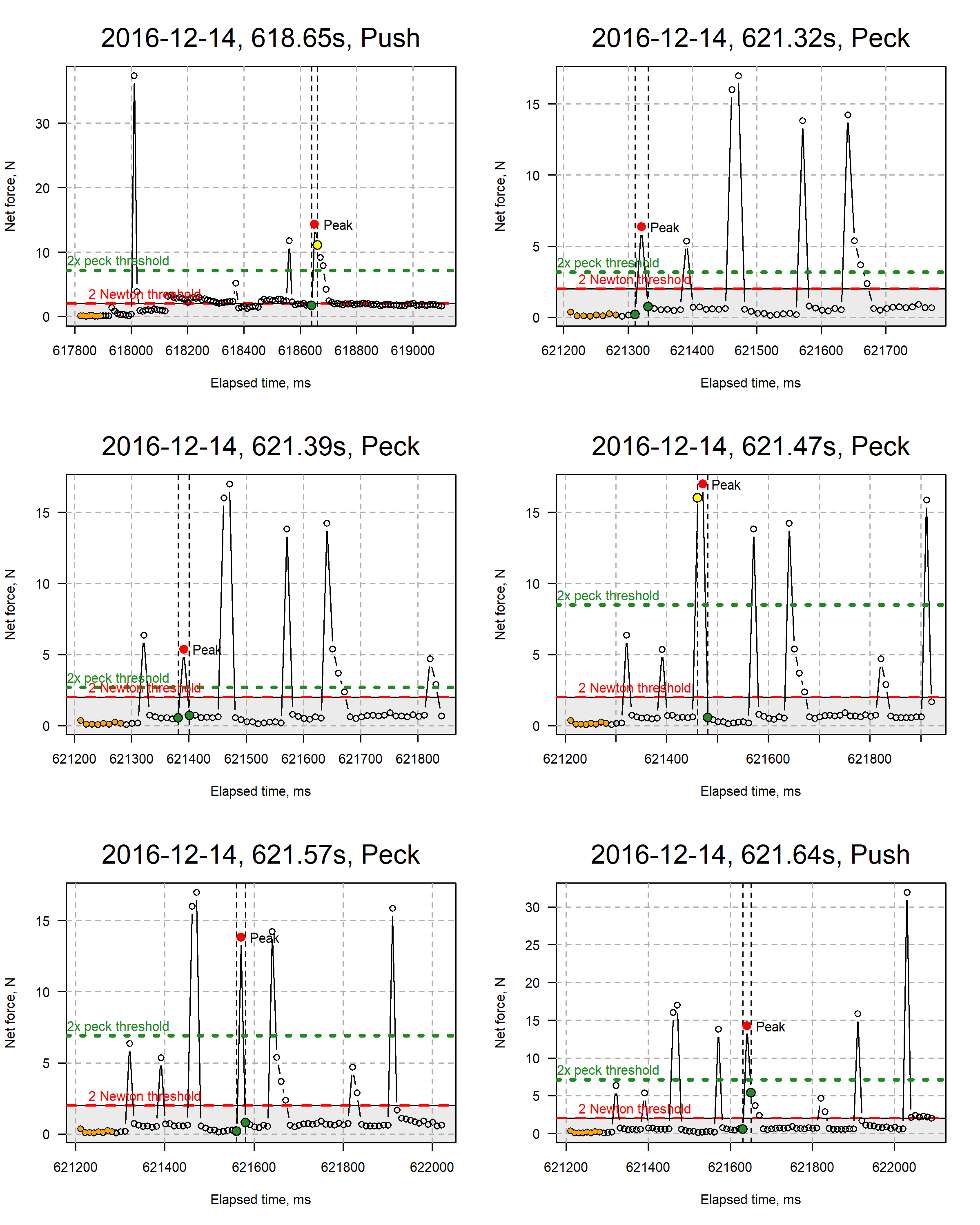
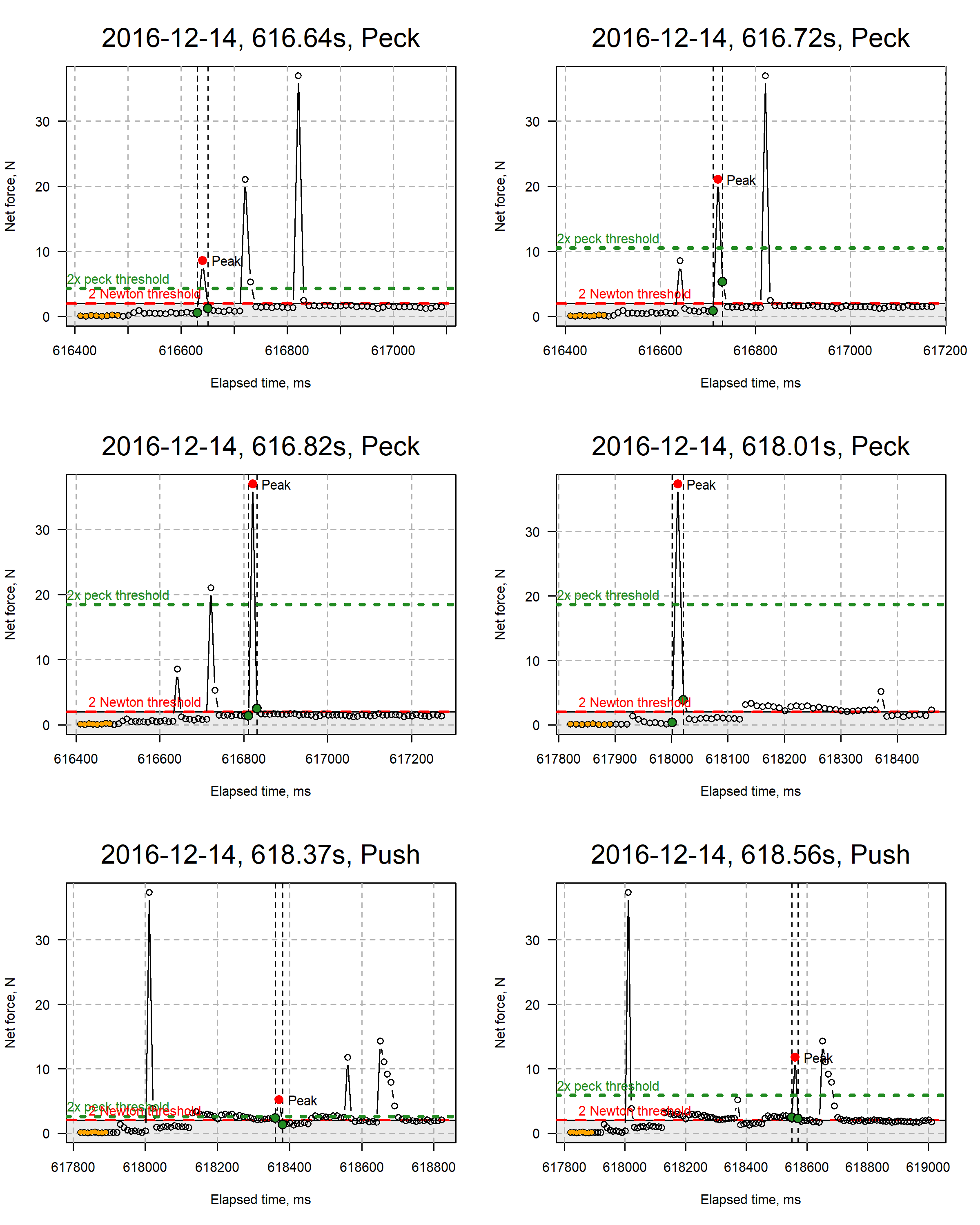
## Max Min Mean SD Median N  
## Peck 37.36 2.01 6.68 6.92 4.20 244  
## Push 31.89 2.15 9.07 5.93 7.95 129

## -----------------------------------

## Summary data for the highest 10% of events. Values are force in Newtons.

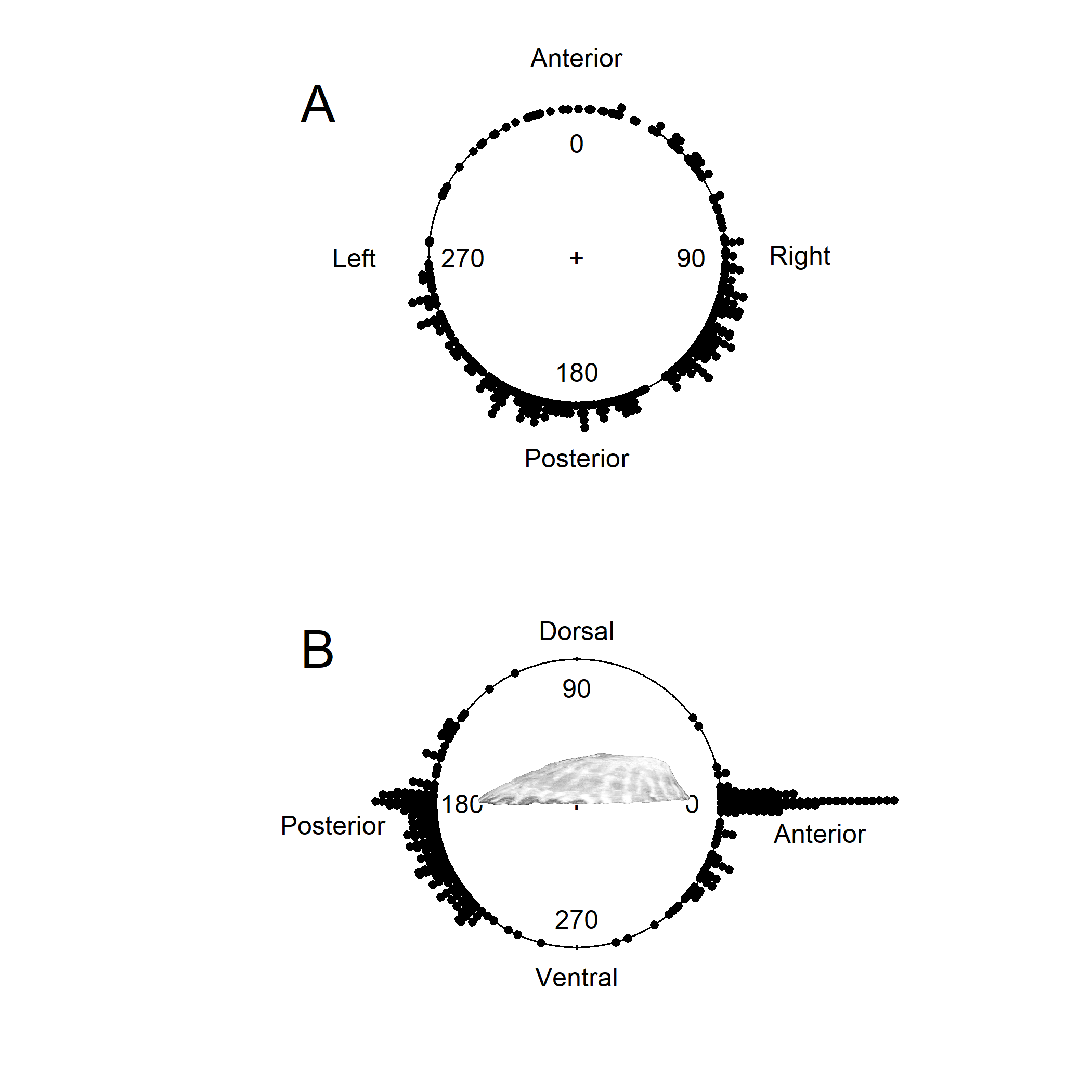
## Mean SD N  
## Peck 24.48 7.27 23  
## Push 21.76 5.86 12

Below are some example classified Pecks and Pushes. The figures have guidelines showing where the thresholds are for classification as a peck or push. These are not all of the events picked out.



# Peck/Push direction

We can use the 3-dimensional force data to estimate the direction/angle that the bird attacked the mimic limpet at.



Angular distribution of force vectors along the anterior-posterior axis and dorsal-ventral axis of the limpet mimic. Each point represents a single strike event by the black oystercatcher during trials on 6 dates. Forces directed anteriorly are plotted on the right side of the plot. Points lying below the horizontal axis indicate forces with a downwards (ventral) vertical component.