```
//Global Constants
SET half_minute=0.5                     //Value for half a minute
SET threshold_multiplier=1.75           //Level of intensity considered a peak
                                                  //when no pre-stimulation.
SET extrapolation_size=10               //Number of timepoints to add when extrapolating
                                                  //data.
SET smooth_width=10                     //Width of the smoothing window
SET runin_modifier=0.1                  //How far below intensity at time 0 to set
                                                  //values for an artificial run-in.
SET max_detection_width=40              //Maximum peak width to detect.
SET fitpoly_width=7                     //Peak width beyond which a polynomial is
                                                  //fitted.

DEFINE AnalysePeaks(timepoints, ratios) RETURNS peaklist, oscilation period

     // Detect prestimulation period
     SET timepoint_gap = timpoint[1] – timepoint[0]

     FOR each timepoint
          IF current timepoint - previous timepoint > (timepoint_gap + half_minute) THEN
               SET prestimulation_end = current timepoint
          END IF
     ENDFOR

     // Calculate detection threshold
     IF prestimulation_end is not 0 THEN
          SET detection_threshold = mean(prestimulation ratios) + 2*stddev(prestim ratios)
     ELSE
          SET minimum_ratio = lowest ratio value
          SET detection_threshold = minimum_ratio * threshold_multiplier
     END IF


     // Extrapolate values
     IF prestimulation_end is 0 THEN
          SET all timepoints = timepoint + extrapolation_size*timepoint_gap
          FOR count = 0 to extrapolation_size-1
               create new timepoint at timepoint_gap*count
               SET ratio for new timepoint to ratio at timepoint[0] – runin_modifier
          ENDFOR
     ENDIF


     IF derivative of ratios at final two timepoints is negative THEN
          REPEAT
               create new timepoint
               extrapolate new ratio using gradient as measured between previous two timepoints
               append new timepoint and ratio to the timepoints
          UNTIL ratio at final timepoint < detection_threshold
     END IF


     // Optimise detection width
     FOR width = 1 to max_detection_width
          CALL detectPeaks(timepoints, ratios, threshold, width)
          STORE firstPeakPosition in FirstPeaks[width]
          IF varience(FirstPeaks) > 1 THEN
               SET detection_width = width
          ENDIF
     ENDFOR

     SET peaks = CALL detectPeaks(timepoints, ratios, threshold, detection_width)

     IF number of peaks > 1 THEN
          FOR each peak
               STORE current peak position - previous peak position in peakDistances
          ENDFOR
          SET oscillation period = mean (peakDistances)
     ENDIF
RETURN peaks, oscillation period

DEFINE detectPeaks(timepoints, ratios, threshold, detection_width) RETURNS peaklist

     SET ratios = CALL smooth(ratios, smoothwidth)

     FOR each ratio value
          IF slope direction between ratio values has changed from positive to flat or negative THEN
               IF ratio value > threshold THEN
                    CALL fitpoly(time[current to current+detection_width],
                                                  log(ratios[current to current+detection_width]), 2)
               ENDIF
               IF detection_width > fitpoly_width THEN
                    SET peakHeight = height from fitpoly
                    SET peakPosition = position from fitpoly
               ELSE
                    SET peakHeight = max(ratios[current to current+detection_width])
                    SET peakPosition = time of peakHeight
               ENDIF

               STORE peakHeight, peakPosition in peaks
          ENDIF
     ENDFOR
RETURN peaks
```