```
*******************************************************************************
**Updated: 2014-AUGUST-15
**Project: Fully automated waist-worn accelerometer algorithm for
children's sleep period time separate from 24-hour physical activity or
sedentary behaviors
Tudor-Locke C, Barreira TV, Schuna JM Jr, Mire EF, Katzmarzyk PT.
Fully automated waist-worn accelerometer algorithm for children's
sleep period time separate from 24-hour physical activity or
sedentary behaviors, Appl Physiol Nutr Metab. 2014 Jan;39(1):53-7.
doi: 10.1139/apnm-2013-0173.
** Source Code: a commonly used and publicly available non-wear
algorithm developed by the National Cancer Institute
(http://riskfactor.cancer.gov/tools/nhanes_pam/); most modifications
are indicated in CAPITAL LETTERS
** Coding: Emily F. Mire, Pennington Biomedical Research Center, 2013
** Notes: Some of the variables used are defined in the documentation
(http://www.cdc.gov/nchs/nhanes/nhanes2005-2006/PAXRAW_D.htm), and the
rest are created using the code from the website
http://riskfactor.cancer.gov/tools/nhanes_pam/. Below are the
variables specific to this program:
**PID = subject identification number
**AccelTime = time output by the accelerometer, formatted to include
hour and minute
**Noon = sequential noon-to-noon day variable
**Inclinometer = indicator for position of accelerometer at time of
output (0 = Off, 1 = Standing, 2 = Lying down, 3 = Sitting)
**USERS ARE REFERRED TO AN UPDATED AND REFINED VERSION LOCATED AT
http://www.pbrc.edu/pdf/PBRCSleepEpisodeTimeMacroCode.pdf AND
DESCRIBED IN Barreira, T.V., Schuna Jr., J.M., Mire, E.F.,
Katzmarzyk, P.T., Chaput, J-P., Leduc, G., & Tudor-Locke, C.
Identifying children's nocturnal sleep using 24-hour waist
accelerometry. Medicine and Science in Sports and Exercise, in press.

*******************************************************************************

/*CREATING ALGORITHM 1 - SADEH*/

data setvar;
set sample;
by pid;

/*Natural log of current minute*/
LogAct = log(paxinten+1);

/*Indicates if current minute intensity 50-99, to be totaled later*/
if paxinten ge 50 and paxinten < 100 then SedInd = 1; else if paxinten = .
then sedind = .; else sedind = 0;

/*Rolling STD of current minute and 5 preceding*/
STDpre5 = std(paxinten, lag(paxinten), lag2(paxinten), lag3(paxinten),
lag4(paxinten), lag5(paxinten));

IF PAXN < 6 then STDpre5 = .;
```

```
run;

/*Rolling mean of previous and next 5 minutes and total count of minutes with
intensity between 50 and 100*/

proc expand DATA = setvar OUT = rollmean;
convert paxinten = MeanW5 / METHOD = none TRANSFORMOUT = (cmovave 11);
by pid;
run;

data rollmean2;
set rollmean;
format meanw5 8.2;
drop time;
run;

proc expand DATA = rollmean2 OUT = countout;
convert sedind = SumInd / METHOD = none TRANSFORMOUT = (cmovsum 11);
by pid;
run;

/*Set mean and sum of first 5 and last 5 of total wear time since there are
no future minutes to use in calculation*/
data lastmin;/*8*/
set sample;
by pid;
if last.pid then do;
      lastpax = paxn;
      output;
end;
keep pid lastpax;
run;

data sets;
merge lastmin countout;
by pid;
if (paxn > (lastpax - 5)) or (paxn < 6) then do;
meanw5 = .; sumind = .;
end;
run;


data sadeh;
set sets;
ProbSleep = 7.601 - (0.065*MeanW5) - (1.08*sumind) - (0.056*STDpre5) -
(0.703*logact);
if probsleep = . THEN DO; PROBSLEEP = 1; END;
if probsleep < 0 and probsleep ne . then SleepSadeh = 0; if probsleep ge 0
then SleepSadeh = 1;



/*ALGORITHM 2 - INCLINOMETER ADJUSTMENT*/
if probsleep < 0 and probsleep ne . then SleepIncOff = 0; if (probsleep ge 0)
or inclinometer = 0 then SleepIncOff = 1;
drop lastpax time;
run;
```

```
data SadehNoon;
retain PID  Date AccelTime NOON Day PAXDAY PAXN PAXINTEN Inclinometer
SleepSadeh SleepIncOff;
  set sadeh;
keep PID  Date AccelTime NOON Day PAXDAY PAXN PAXINTEN Inclinometer
SleepSadeh SleepIncOff;
run;



/*ALGORITHM 3 - 160 MINUTE MACRO*/

data monitors;
set SadehNoon;
by pid;
run;

proc sort data=monitors;
by pid noon paxn;
run;

%macro nw(nwperiod=);
data nw_all;
  set monitors;
  by pid NOON paxn;

  if first.NOON then nw_num=0;    /*non-wear period number*/

  if first.NOON or reset or stopped then do;
     strt_nw=0;       /*starting minute for the non-wear period*/
     end_nw=0;        /*ending minute for the non-wear period*/
     start=0;         /*indicator for starting to count the non-wear period*/
     dur_nw=0;        /*duration for the non-wear period*/
     reset=0;         /*indicator for resetting and starting over*/
     stopped=0;       /*indicator for stopping the non-wear period*/
     cnt_non_zero=0; /*counter for the number of minutes with SLEEP = 0*/
       END_MIN = 0;           /*LITERAL ENDING MINUTE (TIME THEY WOKE UP)*/
       CNT_SLEEP = 0;    /*COUNTER FOR NUMBER OF MINUTES SLEEPING*/
       BEDTIME = 0;
  end;
  retain nw_num strt_nw end_nw stopped reset start cnt_non_zero dur_nw
END_MIN CNT_SLEEP BEDTIME;

  /*Counting sleep minutes so that it will start a sleep period*/
    if SLEEPINCOFF = 1 then
     cnt_sleep=cnt_sleep+1;
      /*before reaching the 5 consecutive minutes of sleep, if encounter one
      minute with wake, reset the counter*/
  if SLEEPINCOFF = 0 then do; cnt_sleep=0; end;


  /*The SLEEP period starts with 5 CONSECUTIVE MINS OF SLEEP*/
  if start=0  and cnt_sleep ge 5 then do;
        strt_nw=paxn-4;    /*assign the starting minute of SLEEP*/
```

```
        start=1;
        BEDTIME = ACCELTIME - 240; /*4 MINUTES PRIOR*/
  end;
FORMAT BEDTIME TIME8.2;

  /*accumulate the number of the SLEEP minutes*/
  if start and SLEEPINCOFF = 1 then do;
       end_nw=paxn;         /*keep track of the ending minute for the SLEEP
period*/
       END_MIN=ACCELTIME; END;        /*keep track of the LITERAL TIME FOR
THE END OF the SLEEP period*/
FORMAT END_MIN TIME8.2;

/*keep track of the number of AWAKE MINUTES*/
  if SLEEPINCOFF = 0 then
     cnt_non_zero=cnt_non_zero+1;


/*before reaching the 10 consecutive minutes of AWAKE, if encounter one
minute with SLEEP, reset the counter*/
  if SLEEPINCOFF = 1 then do; cnt_non_zero=0; end;

  /*duration of SLEEP period*/
  dur_nw=end_nw-strt_nw+1;

  /*A SLEEP period ends with 10 consecutive minutes of SLEEP = 0 (AWAKE)*/
  if (cnt_non_zero=10) then do;
    if dur_nw<&nwperiod then reset=1;       /*reset if less than &nwperiod
minutes of SLEEP*/
    else stopped=1;
  end;

  /*last minute of the day*/
  if last.NOON and dur_nw>=&nwperiod then stopped=1;

  /*output one record for each SLEEP period*/
  if stopped=1 then do;
       nw_num=nw_num+1;
       keep pid NOON nw_num strt_nw end_nw dur_nw END_MIN BEDTIME ;
       output;
  end;
run;

*------------------------------------------------------------------------*;
*summarize the SLEEP periods to one record per day                       *;
*------------------------------------------------------------------------*;
proc summary data=nw_all ;
  by pid NOON;
  var dur_nw ;
  output out=sum_nw
       sum=tot_dur_nw
run;

*------------------------------------------------------------------------*;
*summarize the total number of valid minutes for everyone in the analysis.*;
*------------------------------------------------------------------------*;
proc summary data=monitors;
```

```
  by pid NOON paxday;
  var PAXINTEN;
  output out=sum_all
         n=tot_min;
run;


/*create a dataset with one record per minute, for the SLEEP periods only*/
data nw_minutes(keep=pid NOON paxn);
  set nw_all;
  by pid NOON nw_num;
  do i=strt_nw to end_nw by 1;
     paxn=i;
     output;
  end;
run;
DATA NW_MINS(keep=pid NOON paxn paxinten SLEEPINCOFF);
  merge monitors(in=in_all) nw_minutes(in=in_nw);
  by pid NOON paxn;
  if in_nw;
run;


/*create a dataset from the original data, restricted to the AWAKE periods*/
data wear_minute(keep=pid NOON paxn paxinten SLEEPINCOFF);
  merge monitors(in=in_all) nw_minutes(in=in_nw);
  by pid NOON paxn;
  if in_all and not in_nw;
run;

/*summarize the SLEEP minutes*/
proc summary data=nw_minS;
  by pid NOON;
  var SLEEPINCOFF;
  output out=sum_SLEEP
         n=tot_min_SLEEP;
run;

*------------------------------------------------------------------------*;
*final data for one record per day for everyone in the analysis.      *;
*------------------------------------------------------------------------*;
data nw&nwperiod;
  merge sum_all(in=in_all) sum_nw(in=in_nw) sum_SLEEP;
  by pid NOON;
  if in_all;

  if tot_dur_nw=. then tot_dur_nw=0;
  if tot_min_SLEEP=. then tot_min_SLEEP=0;

  keep pid paxday NOON tot_min tot_min_SLEEP tot_dur_nw;
run;

%mend nw;

%nw(nwperiod=160);
```

```
*------------------------------------------------------------------------*;
*Using the bed time from the first sleep period and the wake time       *;
*from the last sleep period.                                            *;
*------------------------------------------------------------------------*;
data newsleep;
set nw_all;
by pid;
if first.pid and last.pid then do; Bed = bedtime; Wake = end_min; end;
      else if first.pid then do; Bed = Bedtime; wake = 0; end;
      else if last.pid then do; wake = end_min; bed = 0; end;
format wake time8.2 bed time8.2;
run;

proc summary data=newsleep ;
  by pid NOON;
  var bed wake ;
  output out=sumtimes
       sum = BedTime WakeTime;
run;

data sleep2;
set sumtimes;
if bedtime < '13:00:00't then SleepMins = (waketime-bedtime) + '00:01:00't;
else sleepmins = (waketime + ('24:00:00't-bedtime))+'00:01:00't;
format sleepmins mmss5.;
if sleepmins = . then do; sleepmins = 0; end;
drop _type_ _freq_;
run;

data reintegrate;
merge monitors sleep2;
by pid noon;
if bedtime < '13:00:00't then do; if acceltime ge bedtime and acceltime le
waketime then ASLEEP = 1; else asleep = 0; end;
else do;
      if (acceltime ge '13:00:00't and acceltime < '24:00:00't) then do;
            if acceltime ge bedtime then ASLEEP = 1; else asleep = 0; end;
      if acceltime < '13:00:00't then do;
            if acceltime le waketime then asleep = 1; else asleep = 0; end;
end;
run;
```