MATLAB codes to extract key health metrics from raw accelerometers data

Introduction

Accelerometery has become an increasingly popular method to quantitatively measure physical activity (PA) and sedentary behavior in free-living environments. Tri-axial accelerometers, which are typically worn on the waist, thigh or wrist, capture the wearer's movement acceleration in three spatial axes. With this movement data, researchers can process the acceleration signals to make meaningful interpretations about the wearer's time spent in sedentary or light, moderate, or vigorous intensity PA. Currently, there is an extensive variety of research-grade accelerometers available to scientists (Burchartz et al., 2020). Each of these devices operate by utilizing the same underlying principle of detecting the amplitude and frequency of the device's acceleration along the Cartesian coordinates, converting the "kinematic component" (signal due to body movement) to a voltage proportional in size, and filtering out excess "noise" due to bouncing of the sensor or external vibrations (Bouten et al., 1997). Researchers have several options of how to interface with this data in order to convert the voltage signals into activity intensities, including analyzing the raw acceleration data (Vähä-Ypyä et al., 2015) or relying on the device manufacturer's proprietary algorithms.

The activPAL monitor (PAL Technologies), which is a thigh-worn tri-axial accelerometer, is particularly useful to measure postural positions. By measuring the position and acceleration of the thigh, the activPAL can detect the wearer's time spent sitting/lying, standing and stepping; additionally, it can measure the wearer's cadence, and this stepping speed is used to indirectly estimate energy expenditure (Harrington et al., 2011). Validation studies have found the activPAL to have excellent reliability in detecting different postures and postural transitions (Grant et al., 2006; Lyden et al., 2012) and can accurately estimate energy expenditure of free-living behaviors (Lyden et al., 2017).

Although the AP, along with other brands of accelerometers, have become widely relied upon by PA researchers as the "state-of-the-art" data collection method, analyzing the collected data is not always user-friendly or intuitive. For example, after collecting PA data with the AP, the manufacturer's software creates eleven different Excel file outputs to present the data in alternate forms. While users have the option to work with the *"Average Valid Day Summaries"* file to obtain averages for various activity metrics, one limitation with this output is that it does not distinguish between average weekday and weekend activities, which can exhibit different activity profiles (Evenson et al., 2015). Thus, researchers seeking to fully capture the nuances of activity may lose this variability. Alternatively, researchers have turned to their own lab-developed software codes to meet their research needs and study aims. The lack of currently available interface-friendly AP data processing methods, and often advanced coding skills required to work with individually developed codes, can present a challenge to the researcher seeking a streamlined method to process AP data.

Thus, we sought to meet the need of the researcher seeking a simple step-by-step method to score activPAL data. We present an open-source MATLAB code that would allow researchers to easily obtain weekday and weekend averages of daily sitting, standing, and stepping time, as well as averages of time spent in light PA (LPA) and moderate-vigorous PA (MVPA). Use of this code does not require any additional coding by the user, unless they wish to add additional features, which is discussed later. Here, we explain the development of the code, its basic functions and a step-by-step guide for users regardless of their experience using MATLAB or coding.

Methods

Code Development and Overview of the Features

Our code uses the 15-second epoch file generated by PALanalysis (Version 8.11.2.53). This output file sums up all events occurring in each 15-second interval throughout the entire period the AP has been initialized. The second essential component needed with the code is a participant wear-time log, which records their daily wake times, bedtimes and any periods of non-wear or removal of the device. Detailed instructions on placement and log recording of activPal devices have been previously published by researchers (REF). Using this 15-second epoch file and the wear-time log, our code outputs daily summaries and weekday and weekend averages of the wearer's daily activity profile (i.e. time spent sedentary, standing, and moving at light or moderate-vigorous intensities). Our code does not detect and report specific nuances of behavior, such as bout lengths and postural transitions. However, users familiar with MATLAB who seek to incorporate these elements in the code have the ability to do so.

Key considerations and decisions made when developing the code are listed in **Table 1**. Our code was developed based on previously published data analysis procedures and methods of classification (REF). It was tested and validated on a sample of n= ## young adults (ages #-#) who wore the AP following a 24-hour protocol for an average of ## days. While optimal wear-time protocols have been previously recommended (Edwardson et al., 2017), the code can be easily modified to meet the researcher's study-specific wear-time protocol and valid day criteria.

Step 1: Identifying number of days and length of each day

Our code defines a day as the wearers' reported wake time (out of bed) to bed time (time in bed). This operational definition of a day, which takes a "person-orientated" approach, has been previously suggested to most accurately capture a person's daily activity profile by not removing periods or breaking up bouts of activity just because they occur during or after an arbitrary time point (i.e. midnight) (Edwardson et al., 2017). The code reads each row in the "Participant Wear Log" to identify each "wake time" and "bed time" over the course of the wear period. Additionally, if the participant removes the device during the day, this "off period" can be accurately accounted for by logging the blocks of time the device was removed and re-worn. This addition of user interface differs from the AP software, which does not allow the user to manually input times the device may have been removed during the day. Rather, the AP's proprietary algorithm sets the minimum duration of non-wear time as 60 minutes and relies on "non-varying" accelerometer data. However, if an individual removes this device but places it on an unstable surface or carries it with them (i.e. in a backpack, left on a table that is moved a lot), then that movement may be inaccurately recorded as wear time. Thus, our code eliminates the reliance on automatic algorithms by allowing the user to manually enter any off-on time periods recorded by the wearer.

Criterion for a valid day has been previously explained elsewhere (Edwardson et al., 2017), but the most common criteria is accumulating at least 10 hours of wear-time per day. Thus, 10 hours has been coded into the program as the minimum hours wear-time per user-day to be considered valid. If the code identifies less than 10 hours, that day will be removed and not used in analyses. However, the code can be easily modified to any user-selected wear-time criteria.

Step 2: Identifying and removing nighttime sleep periods

To properly classify non-wear periods as nighttime sleep and indicate in the code that this is the end of the wear-defined day, we have programmed the code to recognize sleep bouts as any removal period greater than 5 hours. This time was selected based on previous work by (Winkler et al., 2016) to develop an automated algorithm to identify sleep/nonwear periods and valid days. The study authors classified bouts of sleep/non-wear if either the bout was ≥ 5 hours or the longest bout of ≥ 2 hours in a 24 hour period. This parameter can be easily modified in our code to meet the user-specific sleep time criteria. Identifying this bout as “sleep time” is important, as it indicates the end of one-user defined day and the beginning of the next. By not restricting this bout to occur between an arbitrary time frame, the code can accommodate any wearer’s sleep schedule.

Step 3: Calculate MET values for each 15-second epoch

In the 15-second epoch file, the “activity score column” is the total sum of METs accumulated during each 15-second sampling period. This summed total does not allow the user to apply standard guidelines to classify the activity levels which occurred during each epoch as sedentary (≤1.5 METs), light (2-2.9 METS), moderate (3-5.9 METS) or vigorous (≥ 6 METs) intensity (American College of Sports Medicine, 2018). By converting this data column to an average rate of energy expenditure which occurred during each 15-second epoch, standard guidelines can be applied to classify the approximate intensity of each specific 15-second epoch. This classification is later used when distinguishing the amount of time spent in LPA and MVPA.

To calculate this average rate of energy expenditure, the code divides each activity score, (which is the sum of METs accumulated over 15 seconds) by 15 to create a new variable of "average METs." For example, for an activity score of 18.75 outputted in the 15-second epoch file, dividing this by 15 equals an average energy expenditure rate of 1.25 METs per second- which is formally classified as "sedentary behavior" (Tremblay et al., 2017)**.**

Step 4: Create "sitting time" variable - remove primary lying time.

Sedentary behavior is formally defined as "any waking behavior characterized by an energy expenditure ≤1.5 metabolic equivalents (METs), while in a sitting, reclining or lying posture" (Tremblay et al., 2017). In the 15-second epoch file, sedentary time is represented among four columns – "sedentary time," "primary lying time," "secondary lying time" and "seated transport time." The "sedentary time" column contains the amount of sedentary time in each 15-second epoch. In our MATLAB code, the *total daily sedentary time* variable is calculated by summing all sedentary time (both sitting and lying) accumulated during waking hours (time out of bed – time in bed). This new variable is the sum of two columns from the 15 second epoch file, "sedentary time" and "secondary lying time." "Primary lying time" is not included in the calculation of this new variable, as it is defined by AP as the longest continuous duration of non-upright (i.e. lying) detected per day, allowing for brief interruptions of upright time for < 15 minutes. For the average participant, this period typically occurs during nighttime sleep. Nighttime sleep is distinct from sedentary behavior, as the latter only includes inactive behaviors which occur during waking hours. Additionally, sleep time is a distinct behavior that is widely studied for its independent effects on health outcomes (Buman et al., 2014; Fanning et al., 2017). In our code, the nighttime sleep period will be identified by the wearer's log (time in bed to time out of bed) and removed from further data reduction steps. However, the researcher is encouraged to cross check the log time on the participant's log with the 15-second epoch file to ensure consistent reporting on their log.

"Secondary lying" time is vaguely described by AP as any non-upright event lasting longer than one hour, but shorter in duration than the *primary lying* period. Examples of secondary lying periods may be mid-day naps or extended TV watching. A wearer is classified to be in a lying position if the non-upright posture is detected for at least 60 minutes. Time spent in "secondary lying" is not counted in the "sedentary time" column in the 15-second epoch file. However, to account for extended periods during the day engaged in a lying position (i.e. napping, sitting on a couch) our code sums this into the outputted total for sedentary time.

"Transport time" is the number of seconds in seated transport (i.e. sitting in a car, on a bus). However, this time is already accounted for in the "sedentary time" column in the 15-second epoch file and thus does not need to be separately added to our code.

Step 5: Create a "standing" variable

The "upright" time computed in the AP 15-second epoch file is a sum of both time spent stepping and standing. Standing, which has a MET value of approximately 1.7 (Mansoubi et al., 2015), has also been investigated as a distinct behavior which may have independent effects on health outcomes (Healy et al., 2015; Van der Ploeg et al., 2014). Thus, it may be of interest to the researcher to independently examine this variable. In order to extract the time spent standing, our code subtracts the "stepping time" from the "upright time" to create a new "standing" variable.

Step 6: Create "moving time" variable – seconds spent either stepping or cycling

A "moving time" variable in our code is computed by adding together the "stepping" time and "cycling time" columns from the 15-second epoch file. This new variable encompasses all physical activity engaged in by the wearer, which will further be categorized into LPA and MVPA. Time spent in LPA and MVPA is calculated using the American College of Sports Medicine-recommended cutoffs (American College of Sports Medicine, 2018). Using the newly calculated activity score variable, our code classifies time spent in LPA if any 15-second epoch has an average energy expenditure rate of 2-2.99 METS, and time in MVPA if the epoch has an average energy expenditure rate of ≥ 3 METs. For each wearer-defined day, our code sums up the time spent in LPA and MVPA and reports these totals in our output file.

Step 7: Outputs- week summary; weekday averages; weekend averages; activity times

The code outputs daily totals, total averages over the entire recording period, weekday (Monday – Friday) averages, and weekend (Saturday – Sunday) averages for time spent sedentary, standing, and moving. Additionally, time spent in LPA and MVPA is reported for each parameter. A sample output is displayed in **Table 2.**

Step 8: Batch scoring multiple ActivPal raw data files

Researchers who use accelerometer and likely to collect data from multiple participants and often in the context of longitudinal studies which involve multiple time-points. With the batch scoring option in our code, researchers can score multiple activPAL raw data files. Once the researcher has completed steps 1-7 to enter all the files from their study participants, a new excel file will be outputted with the summary scores and averages of each participant for the entire sample. A sample batch-scored output is displayed in **Table 3.**

Summary/Conclusion

Previous calls for open-source algorithms to process accelerometer data have aimed at reducing reliance on proprietary algorithms and increasing the researcher's freedom when processing data (Quante et al., 2015). This has prompted a surge in proposed algorithms and independent validation studies, however many of these codes may require advanced coding and technical knowledge. Our proposed MATLAB code offers the user a simplified approach to process activPAL data. By processing participants' specified wear time, the code's output provides a more accurate reflection of the participant's waking day and thus their daily sedentary and activity profile. Our code generates the key parameters researchers are interested in.

Strengths:

- Code allows user to customize the wear's day, does not rely on algorithms to detect wear/non-wear/sleep periods
- Gives breakdown of LPA and MVPA per day and averages
- Can batch score to get sample averages

However, our code is not without limitations

Limitations:

- code is not designed to output bout lengths or sit-to-stand transitions. However, users familiar with MATLAB have the ability to modify the code to fit their specific research aims and desired outputs. Third, the code was tested on young adults.
- Relies on activPAL algorithm to classify EE, which has been found to possibly over or underestimate actual EE (Harrington et al., 2011; Lee & Dall, 2019).

As the AP continues to grow in its utility and application in the exercise research field, this simplified code will allow the researchers to seamlessly interface with the data for quick, accurate reporting.

Consideration	**Observations from the literature Edwardson, 2017**	**Decision applied to code**
1. How to define a "day"	• Defining days by calendar days (midnight-midnight) may disrupt a single bout of behavior which extends past midnight, thus dividing a single bout into two behaviors which occurred on separate days • A "person-orientated" approach, which defines a day as the participant's reported wake to bed time or wake time to following day wake time (if examining entire wake and sleep period) accounts for the variability in individuals	Each day is defined using the "person-orientated approach" using the time when the wearer reported getting out of bed to the time reported going to bed. The code only analyzes waking hours (wake time-bed time), and does not include nighttime sleep in analyses.

	wake schedules and avoids dividing up behaviors which may cross over into the following day.	
2. Number of days analyzed	• At least 7 days is ideal to reliably capture the wearer's habitual pattern of behaviors • 14 days of monitoring is ideal monitoring time frame, but this length of time is not always feasible • Number of days needed can also depend on the study design and purpose	The code can process any number of days per file, allowing the researcher flexibility to work with their selected protocol. The number of days worn, indicated by the participant's wear log, will be detected by the code.
3. How to define a valid day	• 10 hours of wear during waking hours is the most commonly-reported protocol • Other criterion previously reported following 24 hour wear time protocols include: a full 24 hour period of wear; wear time was comprised of $\geq$ 80% of waking hours; the monitor is removed for < 2 hours in a 24 hour period • Other criterion previously reported following waking-hour wear-time protocols include: 4-6 hours of data waking day; monitor worn for $\geq$ 80% or $\geq$ 90% of waking time	It is up to the user to determine if a day is valid or not, using their own selected criteria. If a day is deemed invalid, then the user will not enter that day into the participant's wear log excel file
4. How to handle non-wear time	Studies that have not utilized participant logs primarily rely on automated detection methods. Specific criteria that these automated methods have been based on include: • Classifying long bouts (>8 hours of sitting/lying as non-wear • Identifying 60 min of no movement (with the exception of 2 min or less) • >3 h of continual sitting/lying with 0 or 1 count in the accelerometer channel	The code uses the wear-time log to identify periods of sleep, wake and non-wear time. When the user enters the participant's wear log into the excel file, the user will indicate times when the device was removed and replaced (must occur on the same calendar day). The code will remove these periods from analyses.

	However, another study found that only about 10-15% of accelerometer non-wear time is > 60 minutes (Jaeschke et al., 2017), and that algorithms set to the above-listed criteria may miss these shorter periods.	