mhealthtools: A Modular R Package for Extracting Features from Mobile and Wearable Sensor Data

Phil Snyder[1], **Meghasyam Tummalacherla**[1], **Thanneer Perumal**[1], and **Larsson Omberg**[1]

1 Sage Bionetworks

DOI: 10.21105/joss.02106

Software

- Review
- Repository
- Archive

Editor: Tania Allard

Reviewers:

- @Ebedthan
- @AKuederle

Submitted: 07 January 2020
Published: 30 March 2020

Introduction

As life expectancy continues to rise with the increasing prevalence of modern medical practices, the costs associated with caring for an increasingly large elderly population have climbed as well. In response, there has been an evolving interest in using relatively inexpensive remote monitoring methods outside of the clinic to aid in the early detection and real-time assessment of Alzheimer's disease (Kourtis, Regele, Wright, & Jones, 2019), Parkinson's disease (Monje, Foffani, Obeso, & Sánchez-Ferro, 2019), mood disorders (Jacobson, Weingarden, & Wilhelm, 2019), and a wide array of other diseases (Majumder & Deen, 2019). Consumer-grade sensors embedded in smartphones and wearables have the potential to provide clinicians with remote *digital biomarkers* that can aid in the early detection and monitoring of disease via frequent, objective, and unobtrusive assessments (Coravos, Khozin, & Mandl, 2019). The extraction of relevant digital biomarkers from raw sensor data requires a combination of signal processing, data science and biological expertise in addition to extensive validation data (Goldsack, Chasse, & Wood, 2019; Monje et al., 2019). Software libraries to preprocess and extract features from such data are becoming increasingly numerous and varied (Czech & Patel, 2019; Saez-Pons, Stamate, Weston, & Roussos, 2019; van Gent, de Bruin, Kris, & Fernandes, 2019). But without an established set of preprocessing techniques and features for capturing disease signal, it becomes necessary to make use of a broad spectrum of methods and tools when exploring potential biomarkers (Kumar et al., 2013). To this end we have developed mhealthtools: an R package that aids in the construction of data pipelines for remote sensor data analysis.

mhealthtools is an open-source, modular R package for preprocessing and extracting features from remote sensor data collected using mobile and wearable devices. The package is written in a functional style so that various components of the preprocessing and feature extraction pipeline can be inserted, omitted, moved around, extended, and shared between multiple libraries with minimal overhead. Depending on the data source, the package includes a default set of preprocessing and feature extraction behavior for convenient exploratory analysis. The inertial measurement unit (IMU) (e.g., accelerometer and gyroscope) features extracted using this package have been used by Perumal et al. (2018) and the tapping features by Neto et al. (2019), both using data from the mPower study (Bot et al., 2016). The heart rate features have been used as part of a validation study (Omberg, n.d.).

Design

Data collection from sensors embedded in wearables and mobile devices can occur through active tasks, where participants perform a specified task according to a protocol, or passive tasks, where sensor measurements are collected discreetly in the background during daily living conditions. Features computed from these measurements are typically used as the basis of statistical models designed to explore potential patient outcomes or measures of disease. mhealthtools provides functionality to construct processing pipelines of raw data sourced from sensor measurements recorded during these tasks. This includes data transformations, filtering and feature extraction. In addition to these generalizable pipeline-building functions, mhealthtools provides a set of curated, high level functions that compute features using already constructed pipelines designed for specific active tasks.

In the case of accelerometer and gyroscope sensors, mhealthtools provides functions (`accelerometer_features` and `gyroscope_features`) for extracting general features — agnostic of whether data is being collected as part of an active or passive task. Higher level functions, `get_heartrate`, `get_tapping_features`, `get_walk_features` and `get_tremor_features` are designed to compute features for specific active tasks that use the camera, touch screen sensors, and the underlying `accelerometer_features` and `gyroscope_features`, respectively. Unlike accelerometer and gyroscope, touch screen and camera sensors do not have their own task agnostic feature extraction functions. This hierarchical organization of modules is illustrated in the figure below.

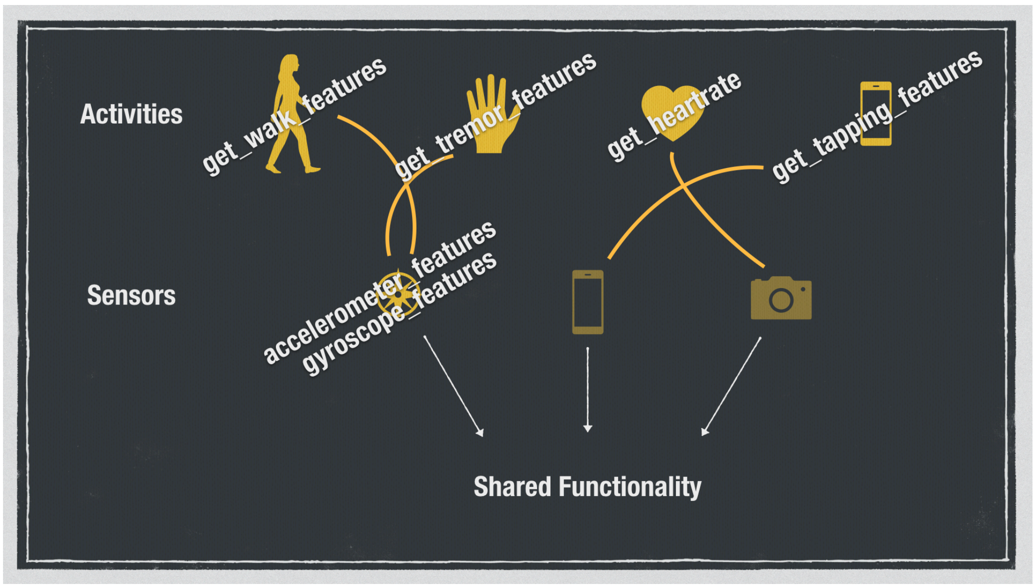

Figure 1: The heirarchical organization of mhealthtools. Note that touch screen and camera sensors do not have their own task agnostic feature extraction functions.

All modules, whether they operate at the activity or sensor level, provide some level of parameterization to modify the default feature extraction pipeline. This concept is best illustrated by an activity-level module which makes use of both the accelerometer and gyroscope feature extraction functions. For example, `get_tremor_features` is sufficiently straightforward to compute features when provided with only the input data as a dataframe in a standardized format — while being flexible enough to accept an assemblage of parameters that control signal detrending, frequency filtering, windowing, which feature sets to compute and more. Additionally, it's possible to modify the order in which certain preprocessing steps are applied or to include external functions within the pipeline.

This flexibility in parameterization is an intentional design choice of the mhealthtools package, made possible by adhering to a consistent functional paradigm. The user is able to easily mix and match components of pipelines which may be designed for extracting features from different activities without needing to worry about potentially inconsistent output formats from those components.

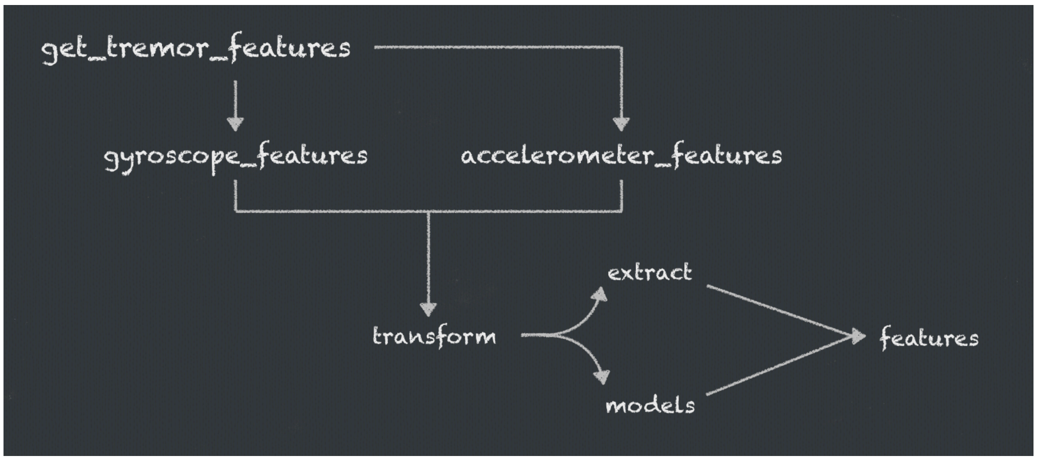

Figure 2: An expanded view of the hierarchical and modular organization of mhealthtools. `get_tremor_features` calls both `accelerometer_features` and `gyroscope_features`, which in turn call a sequence of functions that preprocesses (transforms) then extracts features from the inputted raw sensor data. Each of the components above can be internally modified, extended, or omitted altogether, depending on the desired output.

In the case of the IMU sensors, the paradigm is as follows:

- Input — raw sensor data, in a standardized format.
- Transform — raw sensor data (by, e.g., tidying (Wickham, 2014), computing rates of change, windowing).
- Extract — features by computing statistics upon individual columns, typically grouped on an index such as axis or axis/window.
- Return — features for each group.

The *transform* and *extract* steps may be modified relatively easily, allowing functions from external libraries to be included as part of the pipeline without needing to modify the source code of the package. In addition, to handle the processing of large datasets, mhealthtools provides a robust error-handling mechanism in the case of corrupted input data or other data processing issues — guaranteeing a consistent output.

For more information, including how to use custom preprocessing functions as part of the *transform* step and how to include your own features or machine learning models within the feature extraction pipeline, see the package vignette *Extending mhealthtools*.

Features

Default feature sets are provided for both time and frequency domains. For a full list of features and their definitions, see the *Feature Definitions* vignette.

Comparison with Related Packages

There exist a number of other packages that were built to work with data collected from sensors in a digital health research setting. HeartPy (van Gent et al., 2019) is a Python package focused on extracting features from PPG data collected through PPG or camera sensors. GaitPy (Czech & Patel, 2019) is a Python package designed to extract gait features collected by an accelerometer sensor on the lower back. It includes functionality to identify gait bouts (for measurements collected during free-living conditions), implementations of a number of published algorithms for deriving gait features, and a convenience function to visualize gait events. PDkit (Saez-Pons et al., 2019) is the package with the most overlap in functionality with mhealthtools. Focused on Parkinson's disease, PDkit aims to provide a broad suite of tools, ranging from convenience functions for loading data from popular Parkinson's digital health datasets, feature extraction on measurements collected during commonly performed tasks in Parkinson's studies, and even functions which attempt to map sensor measurements to Unified Parkinson's Disease Rating Scale (UPDRS) Part III scores, a clinically validated metric for measuring Parkinson's disease progression.

In comparison to these packages, mhealthtools is focused solely on the feature extraction process and tries to offer a transparent and generalizable framework that makes minimal assumptions about how the sensor data was collected and how it should be processed. This is reflected in a function design that caters to both ease of use and extensibility. mhealthtools adopts robust, functional programming practices, such as predictable error handling, function interfaces, and output formats, making the package a good fit for ETL workloads and streaming data. The default features for IMU sensors were chosen to work well in both clinical and free-living contexts, but are otherwise signal agnostic and not chosen with one particular phenotype or activity type in mind (task specific feature extraction functions, such as `get_walk_features`, may make use of default argument values to these otherwise task agnostic feature sets). For these reasons, mhealthtools is useful for both quickly generating feature sets for exploratory analysis as well as coordinating the extraction of features across multiple packages for more advanced data pipelines.

References

Bot, B. M., Suver, C., Neto, E. C., Kellen, M., Klein, A., Bare, C., Doerr, M., et al. (2016). The mPower study, Parkinson disease mobile data collected using ResearchKit. *Nature Scientific Data*. doi:10.1038/sdata.2016.11

Coravos, A., Khozin, S., & Mandl, K. D. (2019). Developing and adopting safe and effective digital biomarkers to improve patient outcomes. *npj Digital Medicine*, *2*(1). doi:10.1038/s41746-019-0090-4

Czech, M., & Patel, S. (2019). GaitPy: An open-source python package for gait analysis using an accelerometer on the lower back. *Journal of Open Source Software*, *4*(43). doi:10.21105/joss.01778

Goldsack, J., Chasse, R. A., & Wood, W. A. (2019). Digital endpoints library can aid clinical trials for new medicines. *STAT*. Retrieved from https://www.statnews.com/2019/11/06/digital-endpoints-library-clinical-trials-drug-development/

Jacobson, N. C., Weingarden, H., & Wilhelm, S. (2019). Digital biomarkers of mood disorders and symptom change. *npj Digital Medicine*, *2*(1). doi:10.1038/s41746-019-0078-0

Kourtis, L. C., Regele, O. B., Wright, J. M., & Jones, G. B. (2019). Digital biomarkers for Alzheimer's disease: The mobile/wearable devices opportunity. *npj Digital Medicine*, *2*(1). doi:10.1038/s41746-019-0084-2

Kumar, S., Nilsen, W. J., Abernethy, A., Atienza, A., Patrick, K., Pavel, M., Riley, W. T., et al. (2013). Mobile health technology evaluation: The mHealth evidence workshop. *American Journal of Preventive Medicine*, *45*(2). doi:10.1016/j.amepre.2013.03.017

Majumder, S., & Deen, M. J. (2019). Smartphone sensors for health monitoring and diagnosis. *Sensors (Basel, Switzerland)*, *19*(9). doi:10.3390/s19092164

Monje, M. H., Foffani, G., Obeso, J., & Sánchez-Ferro, Á. (2019). New sensor and wearable technologies to aid in the diagnosis and treatment monitoring of Parkinson's disease. *Annual Review of Biomedical Engineering*, *21*(1), 111–143. doi:10.1146/annurev-bioeng-062117-121036

Neto, E. C., Pratap, A., Perumal, T. M., Tummalacherla, M., Snyder, P., Bot, B. M., Trister, A. D., et al. (2019). Detecting the impact of subject characteristics on machine learning-based diagnostic applications. *npj Digital Medicine*, *2*. doi:10.1038/s41746-019-0178-x

Omberg, L. (n.d.). *Validation of a smartphone measure of vo2max*.

Perumal, T. M., Tummalacherla, M., Snyder, P., Neto, E. C., Dorsey, E. R., Mangravite, L., & Omberg, L. (2018). Remote assessment, in real-world setting, of tremor severity in Parkinson's disease patients using smartphone inertial sensors. *ACM Digital Library*. doi:10.1145/3267305.3267612

Saez-Pons, J., Stamate, C., Weston, D., & Roussos, G. (2019). PDkit: An open source data science toolkit for Parkinson's disease. In *Adjunct proceedings of the 2019 acm international joint conference on pervasive and ubiquitous computing and proceedings of the 2019 acm international symposium on wearable computers*, UbiComp/iswc '19 adjunct (pp. 939–943). New York, NY, USA: ACM. doi:10.1145/3341162.3346277

van Gent, P., de Bruin, J., Kris, & Fernandes, G. (2019). *Paulvangentcom/heartrate_analysis_python 1.2.5*. Zenodo. doi:10.5281/zenodo.3563546

Wickham, H. (2014). Tidy data. *Journal of Statistical Software, Articles*, *59*(10), 1–23. doi:10.18637/jss.v059.i10