IEEE Open Journal of Engineering in Medicine and Biology

DISPEL: a Python Digital Signal Processing Library for Calculation of Sensor Derived Measures from Wearables and Smartphones

A. Scotland, G. Cosne, A. Juraver, A. Karatsidis, J. Penalver de Andres, E. Bartholomé, C.M. Kanzler, C. Mazzà, D. Roggen, *Member, IEEE,* C. Hinchliffe, S. Del Din, *Member, IEEE,* and S. Belachew.

Abstract— ***Goal:*** **This paper introduces DISPEL, a Python DIgital Signal ProcEssing Library developed to standardize extraction of sensor-derived measures (SDMs) from wearables or smartphones data.** ***Methods*****: DISPEL supports custom and third-party data formats and essential end-to-end processing steps from raw data to structured SDM datasets. DISPEL uses an object-oriented codebase for data import, data modelling and SDM extraction and export, with source-to-outcome traceability.** ***Results*****: DISPEL is publicly available under MIT license. It is a flexible, modular framework with practical examples in extracting SDMs from structured tests and continuous monitoring scenarios (e.g. performance outcome assessments of cognition, manual dexterity, and mobility). Embedded data quality checks ensure robustness of SDMs for remotely collected data. The analysis of a smartphone-based balance and gait turn test illustrates the library's capabilities.** ***Conclusion*****: DISPEL provides a highly standardized and robust analysis framework to support traceability and reproducibility in SDM development. We encourage contributions of new processing modules.**

Index Terms—**Signal processing, digital biomarkers, digital health technology, Python, gait, balance, cognition, drawing, smartphone, wearable sensing, inertial sensor.**

Impact Statement— **DISPEL offers a standardized way to extract sensor-derived measures from wearables and smartphone signals with traceability from source to outcome. Its open-source availability aims at supporting multi-disciplinary collaborations to accelerate and scale development and adoption of digital biomarkers/endpoints.**

I. Introduction

REDUCING duration, size and costs of clinical trials and/or augmenting their probability of success is a compelling need in the development of therapeutic solutions for neurodegenerative diseases. Digital Health technologies (DHTs) and sensor-derived measures (SDMs) of motor and cognitive function are promising solutions to this problem. The path to their full adoption, however, remains elusive, long and complex.

SDMs development is hindered by the lack of tools to standardize analysis of data, often leading developers to initiate their own codebase. Furthermore, limited transparency and reproducibility, associated to insufficient algorithmic implementation details in scientific literature, challenges trust within the field. To overcome these hurdles, open-source tools with validated algorithms have been deployed by the community, but often for very specific applications [1-3]. Recent remarkable efforts towards a solution for integration of diverse biomarkers [4] or for effective standardization [5,6] do not yet seem to have achieved the needed level of traceability and generalizability. These stumbling blocks hinder SDM adoption, especially when faced with the complexity of datasets collected within clinical trials. This complexity lies in the amount and variety of 1) the data that can be recorded, 2) the experimental protocols and digital assessments implemented in a study, and 3) the contextual and user-related factors affecting the execution of performance outcome assessments.

This paper presents a signal processing framework (Digital Signal ProcEssing Library, DISPEL) designed to deal with this complexity. Coded in an object-oriented manner in Python, DISPEL has been developed and used in the biopharmaceutical industry. It offers a standardized and traceable process for the extraction of SDMs from data collected with wearables and smartphones during structured assessments/tests or continuous monitoring. Our ambition is to make DISPEL a widely adopted and community-driven tool, to facilitate collaboration in digital biomarker/endpoint development communities and to support collaborative efforts between industry, academia, and regulators to drive DHT literacy and adoption.

II. Materials and Methods

DISPEL's main functionalities are depicted in Fig.1. Reading of datasets into DISPEL's data model is supported via the *io* module. This was originally designed to process data from three mobile data collection applications (Ad Scientiam, Biogen Digital Health, and SensorLog - .json) and then expanded to read data from a wearable sensor provider (APDM Wearable Technologies - .h5). and the Mobilise-D standard [7]. DISPEL expansions will require users to contribute new data types and/or code modules to its functionalities, following examples provided with the library.

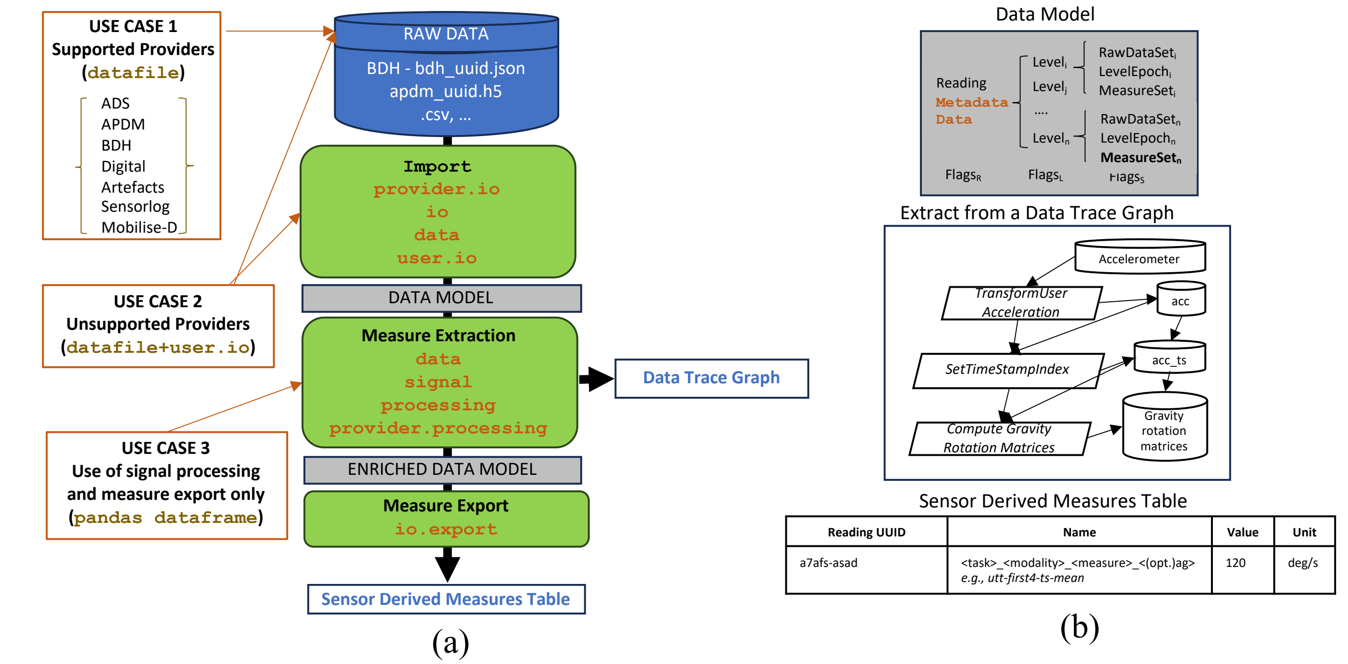

Fig. 1. Panel (a) shows the core functionalities of DISPEL (green blocks) and the modules implementing them (orange text). The brown boxes represent the different scenarios in which a user could interact with DISPEL and the type of data and modules they would need to implement and provide as an input in each case. Panel (b) details the content of the Data Model, the Data Trace Graph and the output table, including an example of automatically standardized measures naming. The Data Model consists of a Reading (including Data and Metadata) structured in a series of levels ($Level_i$) that may contain different sensor data (RawDataSet), custom-defined windows of analysis (LevelEpoch) and calculated measures (MeasureSet). The Model also includes the Flags generated if technical issues or text execution deviations are detected.

DISPEL is designed to process data to assess different functional domains by means of structured performance outcome assessments/tests (e.g. six-minute walk test, drawing test, etc.). These tests are typically composed of sub-tasks, which in DISPEL are structured in the so-called *levels*. For example, if in a smartphone-based drawing task, the subject is asked to draw four different shapes twice with both hands, each attempt is a *level* and the shape and hand are *modalities* (see Fig. 2). This data modelling approach allows for high flexibility and simplifies coding when consuming one or more raw data sets at a time and extracting one or more SDMs. Moreover, it enables extracting SDMs only for specific tasks and sub-tasks and processing only specific *levels*.

Base classes are used to handle a standardized implementation of how SDMs are named and stored: automatically generated measure names directly link to the referring task, level, and modality (Fig.1 (b)).

Measure extraction functionalities are organized in modules per test and relevant processing methods are typically comprised of two generic steps: transforming raw signals and extracting a measure. The available measures extraction algorithms have been optimized based on data from experimental studies[1] in able-bodied controls and persons with neurological disorders. New processing modules can be contributed adding the relevant processing function to a test module.

Measures are wrapped in a class (*measure collection*) to simplify the handling of multiple data collections and aggregations (e.g. multiple measures from same tests, average over different *levels*, etc.). Measure collections can be exported into a Python dictionary, a JSON file, or a CSV file.

A data trace graph documents all the analyses, tracking all transformation and extraction steps (Fig.1(b)).

III. Results

DISPEL is available at https://github.com/newcastleuniversity/DISPEL/. It currently allows to extract SDMs from 16 structured wearable- or smartphone-based tests (including 6 questionnaires), assessing overall disability and quality of life, and measuring performance outcomes in the domains of cognition, manual dexterity and mobility (Table I, Fig. 2). Technical and behavioral data quality checks (*flags*, Table I) have been developed to vet the quality of remotely collected data and ensure robustness of the SDMs.

[1] See e.g., following studies registered at ClinicalTrials.gov: NCT05109637, NCT04756700, NCT03681015.

TABLE I - TESTS, MODALITIES, FLAGS AND MEASURES

Domain	Test	Input data	Sub-Tasks (Modalities)	Data quality flags	Example measure
Quality of life & disability	Mood	Activity sequence (questions, timestamps, responses)	-	-	Median Total Score
	Physical state				Minimum value for score
	MSIS-29				Spasms in limbs
	ALSFRS				Total Score
	PDQ-39				Mobility Score
	Neuro-QOL				Anxiety t-Score
Cognition	Cognitive Processing Speed	Acceleration, Angular Velocity, Touch Events, Activity Sequence	Digit-to-digit (Random/Pre-defined) Symbol-to-digit (Random/Pre-defined) Key (Random/Fixed)	Phone not held in portrait mode	Mean reaction time (predefined sequence, random key, symbol 9)
Upper limb function	Pinch	Acceleration, Angular Velocity, Touch Events	Bubble sizes (Small/Medium/Large Extra-Large) Hand (Left/Right)	No recorded pinch attempt	Mean accuracy normalized by duration (square clockwise, right hand, first attempt)
	Drawing	Acceleration, Angular Velocity, Touch Events	Shape (Circle/Spiral/Square) Direction (Clockwise/Counter-clockwise) Hand (Left/Right)	Phone not held in portrait mode	Mean accuracy normalized by duration (square clockwise, right hand, first attempt)
	Grip Force	Pressure	Hand (Left/Right)	-	Median reaction time (left)
	Finger Tapping	Acceleration, Angular Velocity, Touch Events	Hand (Left/Right)	Simultaneous tapping with both fingers	Total number of valid taps
	Typing	Key Typed, Word Sequence, Touch Events	Correctness (Correct/Incorrect)	Usage of autocomplete	Standard deviation time interval (correct)
	Pronation Supination	Acceleration, Angular Velocity	Hand (Left/Right)	Rotation range smaller than expected	Median movement amplitude (left)
Mobility	5 U-Turn	Acceleration, Angular Velocity, Magnetometer	Static Balance	Excessive motion detected	95% Ellipse area after 5s adjustment
			U-Turn (Number of turns)	Not enough turns performed	Mean absolute turn speed (first 4 turns)
	Walking	Acceleration, Angular Velocity, Magnetometer, GPS (relative)	Six-minutes walking (Continuous, Breaks, Turns)	Too many breaks and turns	Mean walking speed (excluding periods with GPS displacement <1m).
			Two- minutes walking (2 Minutes, 6 Minutes)	Phone not in expected position	Gait variability (first two minutes)
	Gait continous monitoring	Acceleration, Angular Velocity, Magnetometer	-	-	Minutes of activity recorded during the day

Table I: Overview of structured tests, data quality flags associated with incorrect test execution, and example measures that can be extracted with DISPEL. MSIS-29: Multiple Sclerosis Impact Scale. ALSFRS: Amyotrophic Lateral Sclerosis Functional Rating Scale. PDQ-39: Parkinson's Disease Questionnaire. QOL: Quality of Life.

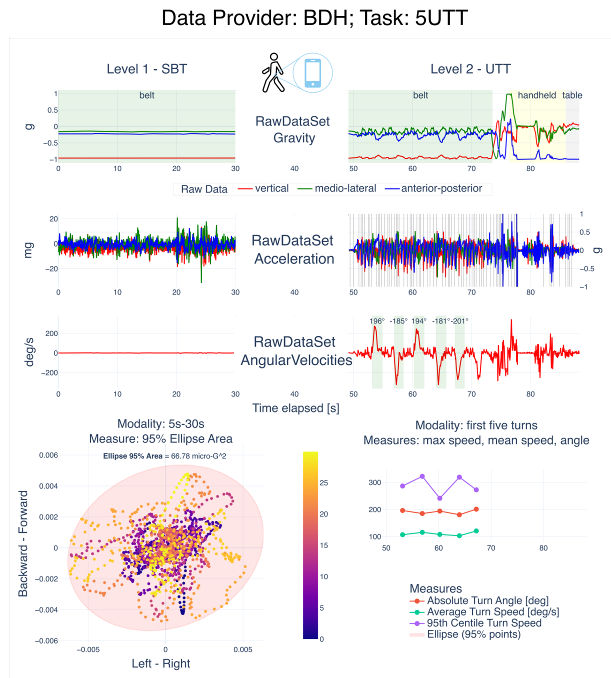

Fig.2 illustrates the use of DISPEL to compute measures from data collected from a 5 U-Turn Test (5UTT). In its BDH implementation, the 5UTT includes two sub-tasks (*levels* in DISPEL), the standing balance (SBT) and the U-turn (UTT). They are performed in sequence and the subject is asked to first stand for 30 seconds (*modality*: process window 5s-30s) and then perform at least five consecutive 180 degrees turns (*modality*: investigate first 5 turns only). Raw datasets, detection of events of interest (e.g., steps and turns), example of context identification (phone handling) and *measures* are also shown.

IV. Discussion

This paper aims to provide the DHT research community with a tool to facilitate the development of digital endpoints in the form of SDMs with superior longitudinal properties for treatment effect detection in clinical trials. To this end, DISPEL was developed and optimized as a framework to extract SDMs from raw signals. DISPEL is freely available as a Python library, together with detailed documentation and step-by-step examples.

Compared to other available tools, its novelties and strengths are that DISPEL allows full standardization and traceability in every step of the analysis process while still ensuring a high degree of flexibility. As a result, DISPEL can deal with essential aspects of the complexity associated with data acquired in real-life unsupervised settings. First, DISPEL ingests and formats data from several providers. Second, DISPEL leverages the advantages of an object-oriented codebase to centralize common functionalities for data processing. Third, DISPEL has a fully modular and customizable pipeline of processing steps, designed to allow for programming simultaneous SDM computations and handle problematic steps (e.g., data missingness or signal corruption during measures extraction). Lastly, DISPEL includes innovative solutions for detecting contextual and user-related factors affecting the obtained SDMs when standardized digital tests are deployed out-of-clinic in self-administered and unsupervised daily life scenarios.

V. Conclusion

DISPEL offers a framework to standardize the generation of SDMs using input from a variety of data providers and easily extendable to incorporate additional providers and processing modules from the DHT community. DISPEL's flagging framework, the automated measure naming, and the visual mapping of all the processing steps ensure transparency and interpretability of the output. This is of the essence especially when operating in the highly regulated intended use of clinical trials.

Acknowledgment

The contributions of Ali Neishabouri, Oussama Tchita, Clément Dulong, Loïc Carment, Angéline Plaud, Kevin

Bouaou to the original design and development of the library is gratefully acknowledged.

References

[1] B. Bent, M. Henriquez, and J. Dunn, ‘Cgmquantify: Python and R Software Packages for Comprehensive Analysis of Interstitial Glucose and Glycemic Variability from Continuous Glucose Monitor Data’, IEEE Open Journal of Engineering in Medicine and Biology, vol. 2, pp. 263–266, 2021.

[2] M. Czech and S. Patel, ‘GaitPy: An Open-Source Python Package for Gait Analysis Using an Accelerometer on the Lower Back’, Journal of Open Source Software, vol. 4, p. 1778, 11 2019.

[3] D. Makowski et al., ‘NeuroKit2: A Python toolbox for neurophysiological signal processing’, Behavior Research Methods, vol. 53, pp. 1689–1696, 8 2021.

[4] B. Bent et al., ‘The digital biomarker discovery pipeline: An open-source software platform for the development of digital biomarkers using mHealth and wearables data’, Journal of Clinical and Translational Science, vol. 5, 2021.

[5] A. Küderle, R. Richer, R. C. Sîmpetru, and B. M. Eskofier, ‘tpcp: Tiny Pipelines for Complex Problems - A set of framework independent helpers for algorithms development and evaluation’, Journal of Open Source Software, vol. 8. The Open Journal, p. 4953, 2 2023.

[6] L. Adamowicz, Y. Christakis, M. D. Czech, and T. Adamusiak, ‘SciKit Digital Health: Python Package for Streamlined Wearable Inertial Sensor Data Processing’, JMIR mHealth and uHealth, vol. 10, 4 2022.

[7] L. Palmerini et al., ‘Mobility recorded by wearable devices and gold standards: the Mobilise-D procedure for data standardization’, Scientific Data, vol. 10, 12 2023.