



# HHS Public Access

Author manuscript

*Proc IEEE Int Conf Pervasive Comput Commun.* Author manuscript; available in PMC 2021 August 26.

Published in final edited form as:

*Proc IEEE Int Conf Pervasive Comput Commun.* 2021 March ; 2021: . doi:10.1109/percomworkshops51409.2021.9431110.

## Signaligner Pro: A Tool to Explore and Annotate Multi-day Raw Accelerometer Data

**Aditya Ponnada, Seth Cooper, Qu Tang, Binod Thapa-Chhetry, Josh Aaron Miller, Dinesh John, Stephen Intille**

*Northeastern University, Boston, MA, USA*

### Abstract

Human activity recognition using wearable accelerometers can enable *in-situ* detection of physical activities to support novel human-computer interfaces. Many of the machine-learning-based activity recognition algorithms require multi-person, multi-day, carefully annotated training data with precisely marked start and end times of the activities of interest. To date, there is a dearth of usable tools that enable researchers to conveniently visualize and annotate multiple days of high-sampling-rate raw accelerometer data. Thus, we developed Signaligner Pro, an interactive tool to enable researchers to conveniently explore and annotate multi-day high-sampling rate raw accelerometer data. The tool visualizes high-sampling-rate raw data and time-stamped annotations generated by existing activity recognition algorithms and human annotators; the annotations can then be directly modified by the researchers to create their own, improved, annotated datasets. In this paper, we describe the tool's features and implementation that facilitate convenient exploration and annotation of multi-day data and demonstrate its use in generating activity annotations.

### Keywords

raw accelerometer data; data annotation; wearables

### I. INTRODUCTION

Sensor-based human activity recognition algorithms use data from ubiquitous and wearable sensors to detect activities people perform in their everyday lives [1]. These algorithms can detect intensity (e.g., vigorous vs. mild movements) and type (e.g., sitting vs. running) [2] of different activities of interest. Such algorithms could not only enable the design of personalized human-computer interfaces, but also support epidemiological efforts to study physical activity at population scales. Algorithm designers, exercise physiologists, and sleep behaviorists often use wearable sensors such as accelerometers, gyroscopes, and/or magnetometers to gather data to detect activities of interest in the real world. In particular, the use of high-sampling-rate raw accelerometer data in training activity recognition algorithms is becoming popular because of the small size, long battery life, and low maintenance demands of wearable accelerometers. Raw accelerometer data has been

used to classify activities such as sedentary behaviors, walking, running, and cycling [2], as well as sleep quality and sensor non-wear.

Researchers use a variety of machine learning approaches to train activity recognition algorithms. The challenge is assessing the performance of, and providing training data for, algorithms as diverse as decision trees and support vector machines, as well as unsupervised learning approaches such as hierarchical clustering. There has also been a surge of interest in deep learning methods for sensor-based activity recognition [3]. Regardless of the approaches used, most machine learning algorithms require – and perform best with – large, annotated datasets of raw sensor data where the start and end times of different activities are precisely labeled.

Large-scale data are already being collected for physical activity research. The influential National Health and Nutrition Examination Survey (NHANES) [4] has gathered raw accelerometer data from 16K+ individuals for one week; the UK BioBank has gathered data from about ~100K individuals [5]. These efforts, and others like them, collect unlabeled raw data at a high sampling rate of 30–100 Hz, thus generating more than 18 million three-axis measurements from each individual in the study. Such efforts could provide valuable data to improve activity recognition algorithms to study physical activity, sedentary behavior, and sleep patterns at individual, as well as population, levels.

As more such datasets become available for researchers, the need for appropriate data exploration and annotation tools will increase, both to generate or clean labeled data and to better understand the performance characteristics of the algorithms against the raw data. However, annotating such multi-day datasets with varied activities poses some challenges. First, it would require researchers to visualize and explore the raw accelerometer data at vastly different time and resolution scales: studying activities such as face touching might require visualization of data at the resolution of a few seconds, but studying activities such as sleep require visualization at the resolution of several hours at once. Second, researchers need to be able to identify precise activity transitions for start and end times for each activity annotation, especially to design algorithms sensitive to transitions between activities in the real world. Another challenge is allowing efficient, interactive, and multi-scale data exploration with datasets that can have hundreds of millions of datapoints. To the best of our knowledge, no such tool is available for this task, and this work was motivated by our own research group's need for such a tool. Thus, we developed an interactive, free, and open-source tool, called Signaligner Pro, for exploration and annotation of multi-day raw accelerometer data. We describe the tool's features, and the mechanisms that render high resolution raw data on web browsers, especially by trading off disk space for interactivity speed. We discuss how the tool can be used for individual, as well as collaborative, labeling by researchers interested in developing activity recognition algorithms.

## II. BACKGROUND

### A. Raw Accelerometer Data

Raw accelerometer data are collected from body-worn triaxial accelerometers measuring acceleration along X, Y, and Z axes. These devices can be worn on wrist, ankle, hip or thigh.

The raw data provides information on intensity (i.e., amplitude of acceleration), frequency (i.e., cycles of repetition), and the orientation of the sensor on the body. Most accelerometers (e.g., the Actigraph GT3X+ devices used in the NHANES) generally sample data up to 100 Hz with up to a  $\pm 8$  g range ( $g = 9.8 \text{ ms}^{-2}$ ). Different activities generate different visual patterns of acceleration (Figure 1). For instance, high intensity activities such as running will have high amplitudes with a rhythmic pattern when measured from a wrist-worn device indicating ambulation. Whereas sleep-like activities will have lower amplitude and occasional orientation changes (e.g., from rollovers during sleep) when measured on the same wrist. The nature of an activity determines the time scale at which it is most easily visually identified by viewing accelerometer data. For instance, eating or face touching activities might require viewing data at short intervals of few seconds, and thus, visualization of the raw, 50–80 Hz data is useful. On the other hand, activities such as sleep require several hours of data to be viewed at once; thus, a visually informative downsampling of the data to a lower sampling rate (e.g., 1–2 Hz) is useful.

## B. Existing Tools for Raw Sensor Data Annotation

Throughout this work, we use the word “annotation” and “label” interchangeably. In typical, lab-based data collections, participants wear accelerometers on different parts of the body and carry out a set of predetermined activities. Research staff then label these activities with start and end times either in real-time or retrospectively, typically using manual notes, a custom application built specifically for the research group’s needs (e.g., [6]), or video annotation. This approach may not be practical for annotating larger datasets. In addition, when machine learning algorithms are deployed on multi-day datasets such as the NHANES dataset, a considerable amount of visualization of results will be required to visually check the algorithm’s outliers.

Even though researchers have developed visual analytics tools for machine learning generally (e.g., [7]), and human activity recognition specifically (e.g., [8]), these tools primarily focused on visualizing feature contributions to the algorithm’s output. There is a dearth of tools that permit convenient visualization and annotation of multi-day raw accelerometer data, an important step in multiple stages of algorithm development. In early stages, researchers like to visually explore accelerometer data to identify relevant features to compute. In training stages of algorithm development, researchers must verify or add labels of the data. And in algorithm validation stages, researchers must inspect outliers in their results, in order to understand the conditions under which the algorithms fail and to gain insight into how to revise algorithms to avoid those failures. Among the tools published to date, Barz et al. [9] and Mitri et al. [10] developed a multi-sensor annotation tool with video reference (recorded during data collection) to annotate data from different on-body sensors. Similarly, Diete et al. [11] developed an image-supported, semi-supervised annotation tool that enables researchers to annotate wrist-based grabbing actions lasting a few seconds using images as ground truth to guide the annotation. Martindale et al. developed an algorithm-assisted annotation tool to label pressure intensity measured from raw acceleration in gait data, which allowed annotators to correct algorithm output on short bouts of movement [12]. However, these tools are not optimized to support visualizing multi-day raw data at varied resolutions for users to annotate.

While it is possible to explore time-series sensor data using existing libraries (e.g., D3.js [13]), our experience shows that they often pose load latencies when visualizing multi-day, high-sampling-rate raw accelerometer data. For example, BokehJS supports visualizing time-series data on web-browsers with interactive annotation plug-ins, but requires additional libraries to render large-scale datasets [14]. Some of these libraries (e.g., Datashader [15]) convert the data into fixed-size raster images to draw in the web-browser suitable for geo-spatial data. However, raw accelerometer data annotation demands flexibility to interactively zoom between high and low resolutions, especially when annotating a diverse set of activities or activity transitions. Inspired by these tools and the need for fast, interactive visualization of multi-day raw accelerometer data for data annotation, we developed Signaligner Pro.

### III. SIGNALIGNER PRO: SYSTEM OVERVIEW

We developed Signaligner Pro by modifying an existing crowdsourcing game, called Signaligner, designed to gather labels on raw accelerometer data from casual game players [16]. This approach has been adopted in the past, where researchers repurposed an existing crowdsourcing game about protein folding (e.g., Foldit [17]) into a research tool for molecular biologists (e.g., Foldit Standalone [18]). Signaligner Pro improves upon the Signaligner game by: 1) enabling multiple resolutions of multi-day, high-sampling-rate raw data to be explored continuously; 2) providing algorithm and user annotations to edit in addition to annotating the dataset from scratch; and 3) providing flexibility of being used as a cross-platform web-interface for collaborative annotation.

#### A. Data Management

The Signaligner Pro system consists of a launcher, a server, and a browser-based client to interact with the raw data (Figure 2), which allows the tool to be run locally on a computer as well as via a web-based collaborative tool. The launcher provides users with a GUI (Figure 3 Left) to import the raw accelerometer dataset(s) and precompute intermediate data structures required for fast, interactive rendering at different resolutions. The launcher also starts the server that serves the pre-computed data along with algorithm (and user) annotations to the client for rendering on the browser. Browser client(s) can connect to the server to receive data for visualization, and the client is used to explore and annotate the raw data; these annotations are sent back to the server for storage. The launcher, server, and client(s) can all be run locally for a single user. However, as the web-based client communicates with the server over the network, data can be stored remotely to allow multi-user collaboration.

#### B. Raw Data Visualization and Exploration

Signaligner Pro visualizes the raw data as a time series with acceleration along the X, Y, and Z axes (Figure 3 Right A). The default view of the data is the most zoomed-out view, because for the multi-day datasets, it is useful to view the patterns across days before progressively exploring the details of each day-, hour-, minute-, or even second-level changes in the acceleration [19]. These lower zoom levels could help identify large bouts of high intensity vs. low intensity movements (e.g., sleep, sensor non-wear, and sedentary

behaviors). The minimum zoom level shows the entire dataset (e.g., seven-days), the maximum zoom shows the raw data-level (e.g., each sample for the 80 Hz NHANES data, typically displaying about ~10 s at once).

### C. Manual Raw Accelerometer Data Annotation

To manually annotate raw accelerometer data (i.e., without editing existing labels), users can select an annotation from the label list and then highlight a segment of the data with that annotation, creating a new layer of user annotations (Figure 3 Right B), where annotations are rendered as color-coded blocks. As the user annotates more segments of the data, new blocks are added to the user's annotation layer and are saved with start and end time stamps for each annotation and sent to the server for storage. Users can edit their own annotations by dragging the edges of the colored blocks in either direction. Finally, users can also select which annotations to view across the dataset (e.g., if interested in checking *only* the instances of walking activities across seven-days). If the user resumes the program after quitting, the annotations from the previous sessions are reloaded allowing users to explore and annotate data in multiple different sessions. Users can change the color scheme for each annotation as well as customize the available annotations by editing a per-dataset configuration file.

### D. Annotation using Existing or Algorithm Labels

Signaligner Pro includes three example activity recognition algorithms (run from the launcher) that classify the imported data into: 1) common physical activities such as walking, running, or sitting [2]; 2) instances of sleep, sensor wear, and non-wear; and 3) valid and invalid sensor data [20]. These annotations are displayed in the same way as user annotation layers as a time series of color-coded blocks (Figure 3 Right C). The algorithm annotations are rendered by drawing a rectangle between the start and end timestamp of the classification. Users can copy and edit the output from the algorithms, removing errors, and improving timing of transitions. Similarly, users can import annotations generated by other Signaligner Pro users for the same dataset, which are also considered as new label sets and are added as new annotation layers. When used as a web-based tool, the server can also serve annotations from different users and allowing users to copy and edit each other's annotations – for collaborative annotation.

### E. Raw Data Pre-computation and Rendering

Signaligner Pro is designed to visualize multi-day raw accelerometer data from sensors collected at a high sampling rate (~48 million samples for just one individual from a seven-day dataset). Annotating multi-day datasets can require users to revisit the datasets multiple times; viewing data must be optimized to save the user time. Thus, in Signaligner Pro, we trade off precomputation time and disk space requirements to minimize the time required for real-time rendering of data on web browsers. This is performed by downsampling the data for different zoom levels, and pre-fetching and loading the appropriate downsampled data only when triggered by a user's action.

**Downsampling Raw Accelerometer Data:** Traditional approaches of downsampling sensor data often compute the summary value (e.g., median or clustering) for a chosen

time-window, to reduce the number of samples needed to visualize and preserve as much information as possible [21]. However, this approach can alter the visual pattern of raw data significantly, especially when there is high variability of movement (e.g., sedentary as well as running activities) in a time-window. Instead, in Signaligner Pro, when downsampling data for a time-window, we compute the minimum and maximum value of acceleration along each axis within that window. We then render each sample of data at this zoom level as the area (a mesh of triangles) between the minimum and maximum values for each axis (Figure 5).

Data for lower zoom levels is downsampled using larger time windows (thus lower resolution or sampling rate) than higher zoom levels. The thickness of the mesh is proportional to the difference between minimum and maximum accelerations within that window. At the raw data level (i.e., sampling rate of the original data at the time of data collection), the minimum and maximum for a window of one sample are equal rendering only a line. Zoom levels vary by a factor of four in terms of the sampling rate. If the raw data are sampled at 80 Hz, then the subsequent lower zoom levels (lower resolution) will be sampled at 20 Hz, 5 Hz, 1.25 Hz, and so on. Thus, when downsampling, the number of samples selected in the downsampling window also increase by a factor of four (i.e., 4, 16, 64, and so on). For a typical seven-day raw accelerometer dataset, this creates eight zoom levels ranging from the raw sampling rate of 80 Hz to 0.005 Hz.

**Rendering Raw Data at Different Zoom Levels:** To reduce real-time load latencies in Signaligner Pro, the downsampled data at different zoom levels are further divided into smaller files of the same size, referred to as “data tiles.” Each data tile represents 1024 samples of data (i.e., 1024 of minimum and maximum values along X, Y, and Z axes), approximately ~30KB each. For a typical seven-day dataset collected at 80 Hz, Signaligner Pro generates ~60K such data tiles. In that case, a data tile at raw zoom level represents 12.8 s of raw data, whereas, a data tile at the lowest zoom level (0.005 Hz) represents ~2.4 days of raw data. These data tiles are generated during pre-computation in tandem with downsampling and are indexed using the file name containing the zoom level and the tile index so that they are fetched and placed at the appropriate location on the time series.

When Signaligner Pro starts, the data tiles of the lowest zoom level (i.e., the most downsampled level) are loaded first and those that fit in the display window are rendered. In parallel, the neighbouring data tiles for panning are pre-fetched and rendered as soon as the user pans in either direction. While zooming in, the initial zooming takes place by scaling the time-axis while fetching the data tiles for next zoom level (i.e., less downsampled data and of higher resolution) in the background. If the zoom scaling exceeds a pre-set threshold to that of the original zoom level, the data tiles of the next zoom level are rendered immediately (Figure 4). Because each zoom level is downsampled by a factor of four, the next zoomed-in level will load four data tiles for each of the data tiles from the lower zoom level. Each data tile being of the same size keeps the loading time consistent allowing users to freely navigate through the data. This approach is similar to rendering maps (e.g., Google Maps [22], or Datashader but without converting the data into fixed-sized raster images), where tiles for different zoom levels and pan directions are pre-fetched in real-time.

#### IV. PILOT EVALUATION

To evaluate Signaligner Pro for raw data annotation, one of the co-authors (R1) collected 8.5 days of raw accelerometer data on himself from the non-dominant wrist (similar to NHANES dataset) and annotated it manually with sleep, wear, and non-wear labels using notes taken during data collection; R1's labels were used as ground truth. Two of the other co-authors (R2 and R3) then annotated this dataset (without the ground truth labels). R2 and R3 edited/corrected the labels provided by the sleep, wear, and non-wear algorithm pre-packaged with the tool. The accuracy of the algorithm's labels compared to ground truth was 0.94 (CI: 0.93, 0.94). The inter-annotator agreement (Krippendorff's alpha) between R2 and R3 was 0.90 (CI: 0.90, 0.91,  $p < 0.001$ ). In fact, when compared to ground truth, R2 and R3 had an annotation accuracy of 0.98 (CI: 0.97, 0.98) and 0.96 (CI: 0.95, 0.96) respectively – improving over the algorithm, especially when the algorithm could not identify any non-wear instances in the raw data, and suggesting high labeling reliability and potential for collaboration between researchers. Table I summarizes the label contributions by R1 (ground truth), R2, R3, and the sleep, wear, non-wear algorithm (*Algo*).

R1 took the longest time to annotate the data, possibly because R1 did not use the input from the algorithm and was referring to notes while annotating and ensuring all label transitions were precisely marked. Even though R1, R2, and R3 only had to label three classes of prolonged activities, they explored the raw data at different zoom levels – from the lowest zoom levels all the way to the raw level when using Signaligner Pro (Figure 6). While R1 and R3 constantly zoomed in (to zoom 6 or 7, 80 Hz displaying ~12.8 s of data or one data tile at a time) and out (to zoom 0, 0.005 Hz displaying ~6.5 days of data or three data tiles at a time) while annotating the data, R2 was able to annotate majority of the data at zoom level 3 (at sampling rate of 0.3125 Hz displaying ~3 hrs of data or three data tiles at a time on the screen). This shows that Signaligner Pro not only allows inspection of several hours of data but also the transitions between the activities at raw data level.

#### V. DISCUSSION AND FUTURE WORK

In this work, we developed Signaligner Pro to enable researchers in activity recognition to explore and annotate multi-day raw accelerometer datasets. Signaligner Pro improves upon the existing sensor data annotation tools by: 1) rendering multiple days of high sampling rate raw accelerometer data that enables exploring data at the week, day, hour, or second level; 2) providing assistance to a human annotator using labels from existing activity recognition algorithms and/or other users; and 3) allowing the flexibility to use the tool as a cross-platform web-based interface for collaborative annotation on large datasets. However, there are opportunities to improve the tool.

##### A. Potential Improvements to Signaligner Pro

Several tools from prior work have used image or video reference to aid raw data annotation for short bouts of movements. Researchers have also started using front-facing cameras to gather additional information to support data annotation. Rendering such images or video synchronized with the raw data would assist with labeling tasks.

In Signaligner Pro, algorithm labels are displayed without the label likelihood or confidence. Often algorithms with similar or partially overlapping activity classifications (e.g., walking and brisk walking) do not classify a single label, but rather output likelihoods of the most probable classifications. Thus, displaying labels with their likelihood scores in Signaligner Pro could assist researchers in adding appropriate annotations to a given fragment of raw data.

Finally, the tool was only assessed among the co-authors for labeling sleep, wear, and non-wear. We would like to evaluate the tool with more users interested in sensor-based activity recognition, especially labeling more complex activities such as sitting, walking, running, and lying down among others, and to improve the tool iteratively. In addition, we would like to formally compare the usability of Signaligner Pro with existing (and freely available) tools discussed in the prior work.

## B. Potential Future Applications of Signaligner Pro

In addition to raw data exploration and annotation, Signaligner Pro could be used to train researchers about raw accelerometer data and to crowdsource annotation from non-experts to support research in activity recognition.

**As a Training Tool:** Researchers often have research staff working to clean and prepare the datasets. However, due to the lack of convenient tools, they rely on using small datasets for training. In such cases, Signaligner Pro could allow researchers to train their teams with multi-day datasets, especially when working with a diverse set of activity-types (e.g., sedentary to sleep patterns) at different resolutions.

**As a Crowdsourcing Tool:** Signaligner Pro was repurposed from a crowdsourcing game designed to gather annotations from casual game players. Signaligner Pro could be adapted to be a crowdsourcing tool where non-experts in the crowd explore and perform data annotation tasks, including data cleaning, quality checking, or algorithm output correction. Crowd assistance would be required for any extensive labeling of a population-scale dataset such as the NHANES or UK-Biobank datasets.

## VI. CONCLUSION

Activity recognition algorithms can detect activities such as sitting, walking, and sleeping, among others, using data from wearable accelerometers. Such algorithms need annotated datasets with precise start and end times of different activities. Signaligner Pro is a tool to explore and annotate high-sampling-rate raw accelerometer data. We invite researchers with high-resolution data annotation tasks to use the open-source tool, and to join us in improving the tool to support new research in activity recognition. Signaligner Pro is available at <http://signaligner.org> under the MIT license.

## Acknowledgment

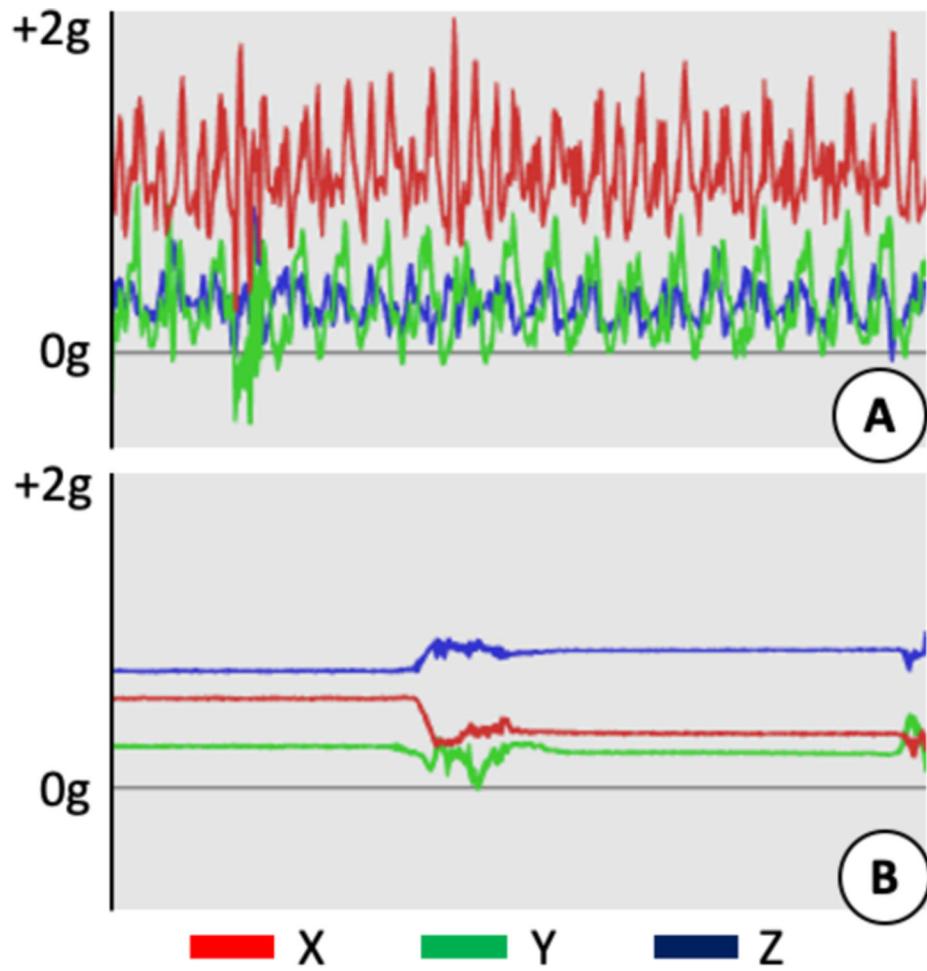
This work was supported by the National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health (NIH) via award number UH2EB024407. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.



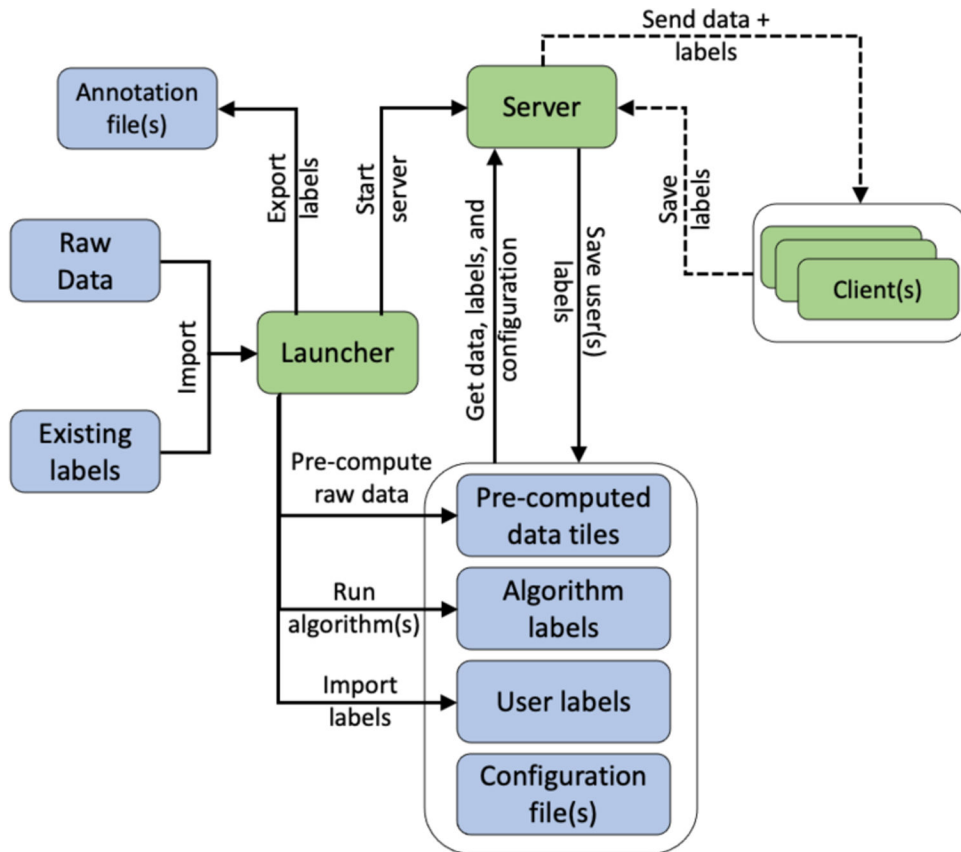
## REFERENCES

- [1]. Munguia Tapia E, Intille SS, and Larson K, "Activity recognition in the home setting using simple and ubiquitous sensors," in Proceedings of PERSASIVE 2004, vol. LNCS 3001, Ferscha A and Mattern F Eds. Berlin: Springer-Verlag, 2004, pp. 158–175.
- [2]. Tang Q, John D, Thapa-Chhetry B, Arguello DJ, and Intille S, "Posture and Physical Activity Detection: Impact of Number of Sensors and Feature Type," *Medicine & Science in Sports & Exercise*, vol. 52, no. 8, pp. 1834–1845, 2020. [Online]. Available: [https://journals.lww.com/acsmmsse/Fulltext/2020/08000/Posture\\_and\\_Physical\\_Activity\\_Detection\\_Impact\\_of.23.aspx](https://journals.lww.com/acsmmsse/Fulltext/2020/08000/Posture_and_Physical_Activity_Detection_Impact_of.23.aspx). [PubMed: 32079910]
- [3]. Nweke HF, Teh YW, Al-garadi MA, and Alo UR, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the Art and research challenges," *Expert Systems with Applications*, 2018, doi: 10.1016/j.eswa.2018.03.056.
- [4]. CDC, "National Health and Nutrition Examination Survey 2011–2012 Overview," 1600 Clifton Rd. Atlanta, GA 30333, USA, 2011, pp. 1–6.
- [5]. Sudlow Cet al., "UK BioBank: An open access resource for identifying the causes of a wide range of complex diseases of middle and old age," *PLoS Med.*, vol. 12, no. 3, p. e1001779, 2015/3 2015, doi: 10.1371/journal.pmed.1001779. [PubMed: 25826379]
- [6]. Anguita D, Ghio A, Oneto L, Parra X, and Reyes-Ortiz JL, "A public domain dataset for human activity recognition using smartphones," in *ESANN*, 2013, vol. 3, p. 3.
- [7]. Wexler J, Pushkarna M, Bolukbasi T, Wattenberg M, Viégas F, and Wilson J, "The What-If Tool: Interactive probing of machine learning models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 1, pp. 56–65, 2019. [PubMed: 31442996]
- [8]. Röhlig Met al., "Supporting activity recognition by visual analytics," in 2015 IEEE Conference on Visual Analytics Science and Technology (VAST), 25–30 Oct. 2015 2015, pp. 41–48, doi: 10.1109/VAST.2015.7347629.
- [9]. Barz M, Moniri MM, Weber M, and Sonntag D, "Multimodal multisensor activity annotation tool," presented at the Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, Heidelberg, Germany, 2016. [Online]. Available: <https://dl.acm.org/citation.cfm?doid=2968219.2971459>.
- [10]. Mitri DD, Schneider J, Klemke R, Specht M, and Drachslers H, "Read Between the Lines: An Annotation Tool for Multimodal Data for Learning," presented at the Proceedings of the 9th International Conference on Learning Analytics & Knowledge, Tempe, AZ, USA, 2019. [Online]. Available: 10.1145/3303772.3303776.
- [11]. Diete A, Sztyler T, and Stuckenschmidt H, "A smart data annotation tool for multi-sensor activity recognition," in 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 13–17 March 2017 2017, pp. 111–116, doi: 10.1109/PERCOMW.2017.7917542.
- [12]. Martindale CF, Roth N, Hannink J, Sprager S, and Eskofier BM, "Smart annotation tool for multi-sensor gait-based daily Activity data," in 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 19–23 March 2018 2018, pp. 549–554, doi: 10.1109/PERCOMW.2018.8480193.
- [13]. Bostock M, Ogievetsky V, and Heer J, "D<sup>3</sup> data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, 2011.
- [14]. Jolly K, *Hands-on data visualization with Bokeh: Interactive web plotting for Python using Bokeh*. Packt Publishing Ltd, 2018.
- [15]. Cottam JA, Lumsdaine A, and Wang P, "Abstract rendering: Out-of-core rendering for information visualization," in *Visualization and Data Analysis 2014*, 2014, vol. 9017: International Society for Optics and Photonics, p. 90170K.
- [16]. Ponnada A, Cooper S, Thapa-Chhetry B, Miller JA, John D, and Intille S, "Designing videogames to crowdsource accelerometer data annotation for activity recognition research," presented at the Proceedings of the Annual Symposium on Computer-Human Interaction in Play, Barcelona, Spain, 2019. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6876631/>.

- [17]. Khatib Fet al., “Crystal structure of a monomeric retroviral protease solved by protein folding game players,” (in eng), *Nat Struct Mol Biol*, vol. 18, no. 10, pp. 1175–7, 10 2011, doi: 10.1038/nsmb.2119.
- [18]. Kleffner Ret al., “Foldit Standalone: A video game-derived protein structure manipulation interface using Rosetta,” *Bioinformatics*, vol. 33, no. 17, pp. 2765–2767, 2017. [PubMed: 28481970]
- [19]. Shneiderman B, “The eyes have it: A task by data type taxonomy for information visualizations,” in *Proceedings 1996 IEEE symposium on visual languages*, 1996: IEEE, pp. 336–343.
- [20]. Troiano R, Intille S, John D, Thapa-Chhetry B, and Tang Q, “NHANES and NNYFS wrist accelerometer data: Processing 7TB of data for public access,” in *Journal of Physical Activity & Health*, 2018, vol. 15, no. 10, pp. S19–S19.
- [21]. Uchida Y and Itoh T, “A Visualization and Level-of-Detail Control Technique for Large Scale Time Series Data,” in *2009 13th International Conference Information Visualisation*, 15–17 July 2009 2009, pp. 80–85, doi: 10.1109/IV.2009.33.
- [22]. Kontkanen J and Parker E, “Earth in Google maps: Rendering trillions of triangles in JavaScript,” presented at the *ACM SIGGRAPH 2014 Talks*, Vancouver, Canada, 2014. [Online]. Available: 10.1145/2614106.2614194.



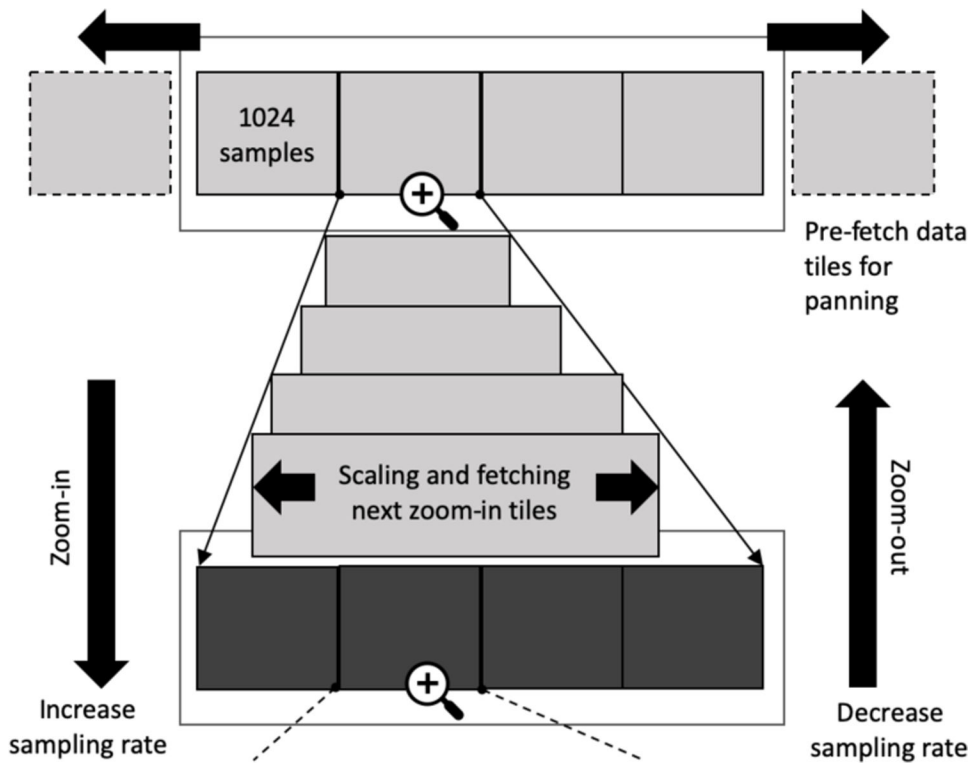
**Fig. 1.** A 10 s sample of raw accelerometer data in g ( $g = 9.8 \text{ ms}^{-2}$ ). (A) Walking activity. (B) Sleeping activity. With only 10 s of data, this sleeping activity is indistinguishable from a different sedentary behavior.



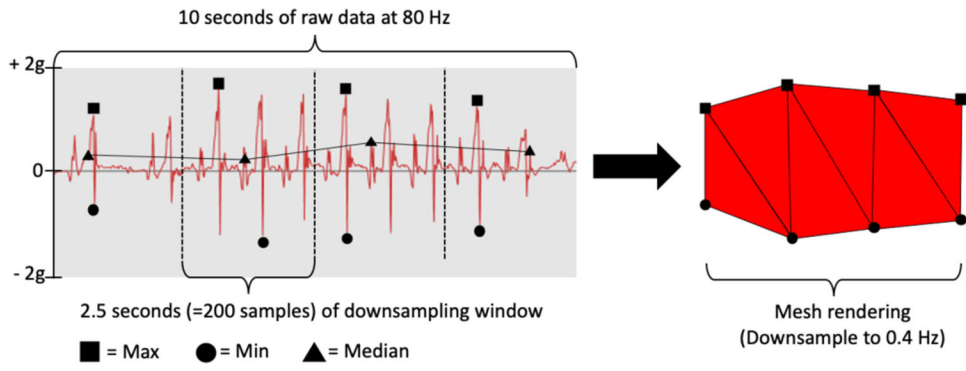
**Fig. 2.** Signaligner Pro system with launcher, server, and client(s). The server and client(s) communicate over a network.



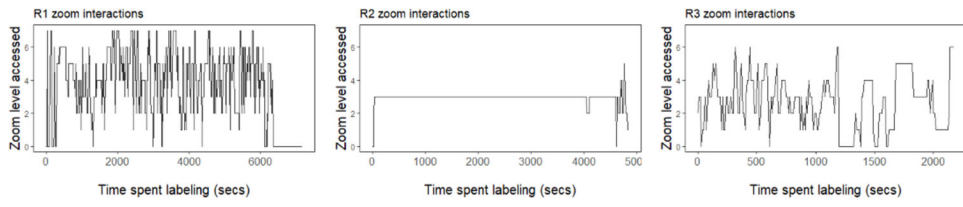
**Fig. 3.** (Left) Signalaligner launcher interface to import raw data and existing labels. (Right) The Signalaligner browser client displays (A) a five-day snippet from seven-days of tri-axial accelerometer data collected at 80Hz, (B) user-added activity annotations, (C) annotations from an activity recognition algorithm, and (D) data-viewing controls.



**Fig. 4.** Loading data tiles at different zoom levels. Data tiles are pre-fetched when zooming in at the pointer and are loaded if the zoom exceeds a threshold. The darker tiles are higher zoom levels (i.e., higher sampling rate) than the lighter tiles.



**Fig. 5.** Downsampling 10 s of walking raw data (X-axis) to 0.4 Hz. The resulting data is rendered as a mesh of triangles.



**Fig. 6.** Zoom levels accessed by R1 (left), R2 (middle), and R3 (right). Zoom level ranges between 0 (0.005 Hz) and 7 (80 Hz)



**TABLE I.**

ANNOTATION CONTRIBUTION USING SIGNALIGNER PRO

	<b>R1</b>	<b>R2</b>	<b>R3</b>	<b>Algo</b>
Annotation time (min)	119.3	80.8	44.3	N/A
Sleep labels (hrs.)	49.0	51.7	44.3	55.3
Wear labels (hrs.)	131.6	130.9	140.9	132.3
Non-wear labels (hrs.)	7.0	5.0	2.4	0

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript