Analyzing and Reporting Wearable Sensor Data Quality in Digital Biomarker Research

Hui Zhang[1], Regan Giesting[1], Guangchen Ruan[1], Leah Miller[1], Neel Patel[1], Chakib Battioui[2], Ju Ji[2], Ming Zhong[1], Andrew David Kaczorek [1], Tianran Zhang[1,3], Yi Lin Yang[1,4]

[1]*Digital Health Office, Eli Lilly & Company, Indianapolis, Indiana, USA*

[2]*Advanced Analytics and Data Science, Eli Lilly & Company, Indianapolis, Indiana, USA*

[3]*Department of Computer Science, Brown University, Rhode Island, USA*

[4]*Department of Computer Science, Purdue University, Indiana, USA*

email: {zhang_hui, rgiesting, ruan_guangchen, miller_leah, patel_neel_k, battioui_chakib, ji_ju, zhong_ming, kaczorek_andrew_david}@lilly.com

tianran_zhang@brown.edu

yang2501@purdue.edu

***Abstract*—Digital Health Technologies (DHT) utilize a combination of computing platforms, connectivity, software, and sensors for healthcare-related uses. Today, these technologies collect complex digital data from participants in clinical investigations, including a large amount of wearable sensor signals. These collected data are used to develop digital biomarkers (dBMs), which can act as indicators for health outcomes for monitoring life quality and measuring drug efficacy. One essential step towards realizing the full potential of these complex digital data is to define the fundamental principles and methods to demonstrate sufficient data quality and fidelity needed for the research. This paper aims to develop a digital data quality assessment framework across the complete data life cycle in dBM research, including data quality metrics and methods to analyze and report digital data quality. Aggregating and reporting digital data quality is often challenging and error-prone. We developed Magnol.Ai, a data platform equipped with data quality assessment and reporting tools that allow us to define data compliance criteria and view data quality reports at different levels in a consumable fashion.**

***Keywords*—*digital health technology*; *connected clinical trial*; *sensor data*; *data quality assessment*; *data visualization*; *digital biomarker*.**

I. Introduction

Digital biomarkers (dBMs) are patient-generated physiological and behavioral measures collected through connected digital devices. The collected data are then used to explain, influence, or predict an individual's health-related outcomes (see [1], [2]). While the development of dBMs invests heavily in advanced analytics, effective results depend on trusted and understood data collected from digital devices. An established data quality assessment framework is thus needed to define the expectation of data, monitor the data for conformance to expectations throughout trials, and report various measures to assess the data quality (see, *e.g.*, [3]). Establishing a meaningful data quality function will help reduce risk throughout the dBM research activities, ultimately ensuring that the criteria for success are met.

Today, we use DHT (see, *e.g.*, [4]) to collect some of the most complex digital data from patients for dBM research. There has been an overall need for better understanding of data, as well as easier access to both data quality and trusted digital data to support operational and analytical activities in the research. Establishing a data quality assessment framework and building software tools to facilitate the assessment is an emerging industry capability. Some unique challenges for this class of data quality strategy include:

- **Complexity of digital data** — We collect some of the most complex digital data in the dBM context, including sensor signals from wearables, patient-reported outcomes from hand-held devices, and labels and annotations processed and used as ground truth information for algorithm development. Handling the wearable sensor data can be challenging. For example, with a sampling frequency of 50Hz, over 4 *million* 3-axial data points are collected from an accelerometer sensor for a single day to understand a patient's daily activities. Similar sensor data streams include, *e.g.*, continuously collected photoplethysmography (PPG) and electrocardiogram (ECG) signals from trial participants.
- **Full-spectrum quality expectations** — Defining quality expectations for digital data and monitoring their conformance to expectations are full-spectrum in the data life cycle. For example, given that data can be collected in a free living environment, scanning the invalid values and noises in wearable sensor signals is often the first profiling step. Identifying the wearable sensor signal's useable (wear-compliant) portions is also a leading data quality function. The ultimate answer to the digital data quality question is the extent that our digital data satisfies the specific requirements needed for dBM analysis.
- **Aggregation and reporting** — Generating various measures to assess digital data quality is not trivial. For example, aggregating compliance information from signal level to the number of analyzable digital measures at the visit and study levels can often be tedious and error-prone. More challenging is reporting data quality in an efficient and effective means across the data life-cycle, and

with more difficulty, at individual participant level, which requires tools to extract and report quantified compliance information, including patterns.

In this paper, we build upon our prior work [1] in the field of data quality assessment. Our contributions can be summarized into two significant parts:

- **Introduction of Magnol.Ai Data Platform**: We introduce Magnol.Ai, a comprehensive data platform tailored to support digital dBM research. At its core, Magnol.Ai incorporates an enhanced data quality assessment framework as an integral component. We begin by offering an overview of the primary digital data categories central to our research focus. Subsequently, we delve into the various metrics employed for profiling digital data quality and their application at multiple aggregation levels.
- **Data Quality Reporting Dashboard**: This paper also presents our innovative Data Quality Reporting Dashboard integrated within Magnol.Ai. The dashboard unifies and streamlines all data quality assessment functions into a cohesive system. It empowers data stewards and quality analysts by allowing them to work seamlessly with specific study data. Users can run processes through interactive workflows and effortlessly generate consumable data quality reports within a centralized, cloud-based ecosystem.

This paper is organized as follows. Section II presents the related work. We present our digital data quality assessment framework and the visual interfaces and tools we developed in Section III. In Section IV, we showcase the data quality reporting portal and the underlying data infrastructure of Magnol.Ai and finally, we conclude the paper in Section V.

II. Related Work

Developing dBMs requires conducting studies in a lab or free-living settings to collect raw sensor data, often with appropriate labels and annotations (*e.g.*, reported patient outcomes). Collection and analysis of wearable sensor data, together with other digital data sets, has thus become an emerging capability needed in dBM development.

Industry players have begun exploring cost-effective and purpose-built solutions in the past few years. For example, the Medidata sensor cloud [5] is used to manage wearable sensor and DHT data for clinical trials. The Koneksa platform [6] provides support to improve compliance monitoring and patient engagement, and other representative efforts to store and deliver raw or processed data from devices in trials, including Evidation [7] and DHDP [8].

Meanwhile, good data is more important than big data in dBM development. Given that wearable sensor data can be collected from participants in a free-living environment, noises, missingness, and invalid values in wearable sensor signals are inherent. To extract and leverage useful and meaningful sensor data, we need to monitor the quality and eventually standardize and process them to support dBM research, as digital data quality is of fundamental importance to developing algorithms for new dBMs (see, *e.g.*, [9] [10] [11]).

In this paper, we are mainly concerned with digital data sets that fall into four general categories:

- *Raw Sensor Signals*. A device typically collects data from multiple sensor signals at varied pre-configured sampling frequencies to minimize study participants' burden under free living conditions. In most cases, the sensor signals are collected in a nonstop $24 * 7$ fashion throughout the entire study, which generally runs between weeks to months. Therefore, assessing potential issues, such as sensor malfunctioning, or wear non-compliance due to participants' behaviors, is critical to ensure data quality can satisfy the downstream analytics needs. Meanwhile, the quality and coverage of sensor data directly correlate to the dBMs derivation, which will be discussed in the later sections of this paper.
- *Scored Data, or Digital Biomarkers*. In addition to raw sensor signals, device companies usually have their proprietary algorithms to analyze sensor data and derive dBMs from it. For example, heart rate and blood volume pulse can be derived from the raw photoplethysmography (PPG) sensor signal. Derived dBMs are at a much lower resolution than the sensor signal, often at the minute or half-minute level.
- *Labels/Annotations*. As algorithms and machine learning models used in developing dBMs become more complex, requirements for large annotated data sets grow. Annotating data for machine learning applications is especially challenging in the biomedical domain as it requires the domain expertise of highly trained specialists to perform the annotations. Annotations can come as interval-based events, with precise timestamps to label the onset and offsets of disease events.
- *Clinical Records*. Apart from raw sensor data and derived dBMs, one yet important piece of data is clinical records that provision key mappings, *e.g.*, device ID to participant ID, participant ID to the treatment cohort, visit dates to treatment phases, *etc*.

Unique challenges arise from these digital data and have made a case for us to develop a data quality assessment framework to define the expectation of these digital data (*e.g.*, completeness, uniqueness, validity, integrity), to monitor the data for conformance to expectations throughout the dBMs trials, and, finally, a user interface as the front-end of Magnol.Ai to display the findings to support operational and analytical activities.

III. Digital Data Quality Assessment Framework

The key functions in our data quality assessment framework should now be clear in Figure 1. The logical series of modeling steps, the problems they induce, and the ultimate resolution of the problems are in the rest of this section as follows.

A. Signal Data Quality Metrics

In the pre-study phase, we establish the Data Transfer Agreement (DTA), to clearly define data quality metrics regarding signal data, including raw sensor signals and dBMs.

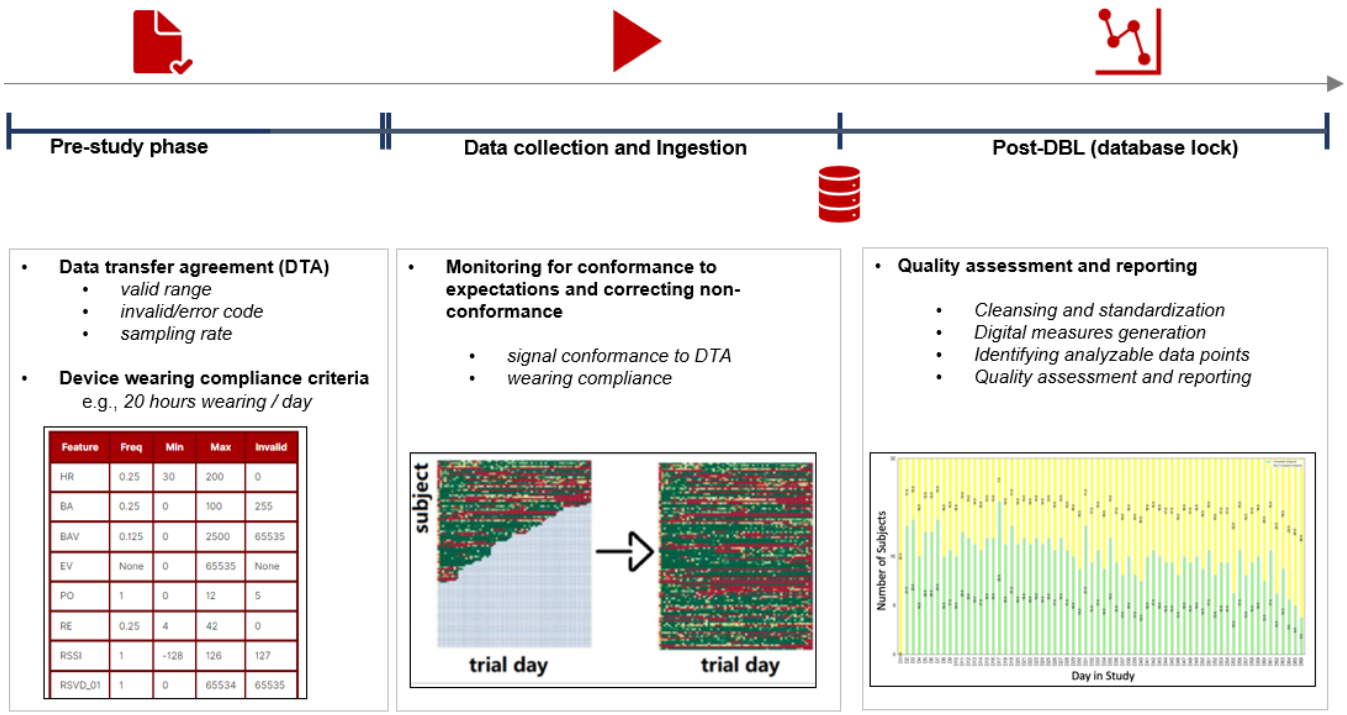

Figure 1: The overall data quality assessment scenario — from establishing the data transaction report (DTA) in the pre-study phase, to compliance monitoring in the live phase, and finally to the quality assessment and reporting in the post-Database Lock (DBL) phase.

TABLE I: EXAMPLE OF A SIGNAL DATA QUALITY METRICS TABLE FOUND IN A TYPICAL DTA DOCUMENT.

Channel	Description	Units	Min Value	Max Value	Invalid Value	Sampling Frequency (Hz)
$accel_x$	Accelerometer X Vector	gravity/1024	-32768	32767	None	50
$accel_y$	Accelerometer Y Vector	gravity/1024	-32768	32767	None	50
$accel_z$	Accelerometer Z Vector	gravity/1024	-32768	32767	None	50
ec	ECG signal	μV	-10000	10000	32767	125
st	Step count	Steps	0	65535	None	1
hr	Heart rate	beats/min	30	200	0	0.25
re	Respiration rate	beats/min	4	42	0	0.25
po	Posture • Laying Down = 0 • Standing = 2 • Walking = 3 • Running = 4 • Unknown = 5 • Leaning = 11	Enum	0	11	5	1

Below we list the typical quality metrics, and Table I gives an example of the data quality metrics table we find in the DTA, where $acce_x$, $accel_y$, $accel_z$ and ec are raw sensor signals, st, po (categorical) are derived dBMs (or, scored data) from accelerometry data, and hr and re are the scored ones from ec.

- **Sampling Frequency** — For raw sensor signals, the sampling frequency is the preconfigured average number of samples obtained in one second. For derived dBMs, it is the resolution of resultant features from analyzing raw sensor data.
- **Valid Range** — For numerical variables (*i.e.*, sensor signals and dBMs), the valid range is indicated by minimum and maximum values that can be measured. For enumerated variables, the valid range is a list of predefined categorical values. One example is the rest classification biomarker, which has the following classes: "awake", "sleep", "toss and turn" and "interrupted".
- **Invalid Value/Error Code** — In addition to the valid range, devices often provision specific invalid values or error codes to indicate different statuses of malfunctioning, which help pinpoint the underlying issue.

B. Signal Data Quality Assessment

Connected clinical trials for dBM research often are conducted under a free living condition, *i.e.*, participants wear sensor devices on a best effort basis using instructions communicated during study enrollment. Inevitably, the free living conditions, potential for device failure or malfunction, and device wearing compliance introduce data issues such as

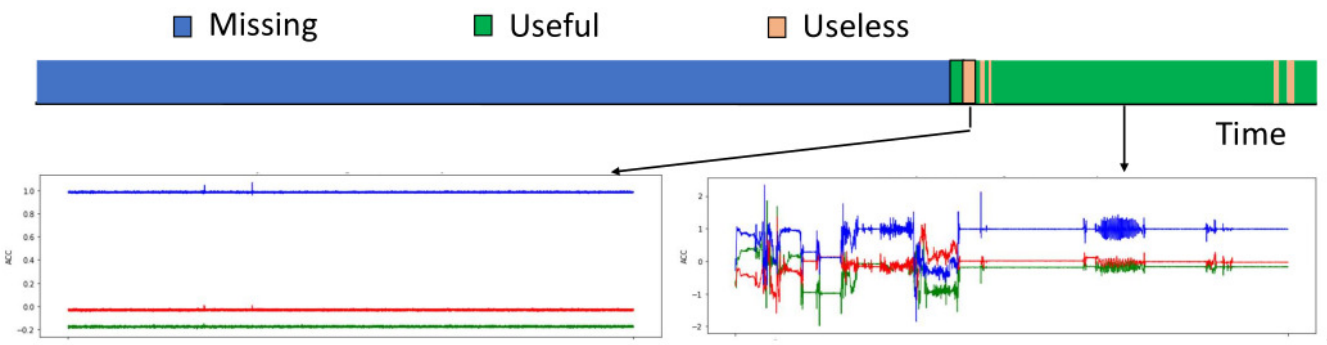

Figure 2: Illustration of sensor signal data issue. Visualized sensor data show different patterns when worn correctly versus incorrectly.

missing data or invalid data collected when participants do not wear or incorrectly wear the devices. Figure 2 illustrates how valid signals (*i.e.*, correctly worn signals) can mix with invalid signals (*i.e.*, incorrectly or not worn signals) in the data collection and how they differ when plotted. Therefore, a *qualitative* means is needed to tell whether a device was operating normally and worn correctly (*i.e.*, **data usefulness**).

To fulfil this goal, the quality assessment is performed in two stages, as discussed in the following.

- **Validity Check**. Data validity checks leverage signal data metrics, as discussed in Section III-A. We immediately know how many valid data points we expect to receive for a sensor signal or dBM using its pre-configured sampling frequency. We can filter out invalid values with a valid value range to get valid data coverage, *i.e.*, coverage of valid data points.
  Since raw sensor signal directly correlates with derived dBMs, we can perform a validity check against the two independently and then align their valid data coverage to check the consistency. We may further overlay device incident events to understand the root cause of observed issues better.
- **Non-wear Detection**. After dropping out invalid data through the validity checking process, the subsequent task is to detect moments when the devices were not correctly worn. The non-wear detection can be challenging as data from such moments can be entirely valid in terms of falling within its valid data range. Instead of reinventing the wheel, we rely on Biobank [12] [13], an accelerometer data processing pipeline whose non-wear detection module is widely adopted as a standard. Below are two key concepts in non-wear detection.
  - *Epoch* — Although data points are collected initially at a high resolution, *e.g.*, 50Hz sampling frequency, the processing is conducted on aggregated values (*e.g.*, 1 or 5 second *short epochs* or 15 minutes *long epochs*) due to the following reasons: (1) collapsing data to epoch summary measures helps to standardize differences in sample frequency across studies; (2) there is little evidence that raw data is an accurate representation of body acceleration, and all scientific evidence so far has been based on epoch averages; (3) collapsing data to epoch summary measures also helps to average out different noise levels making results more comparable across sensor brands.
  - *Non-wear Detection* — Accelerometer non-wear time is estimated based on the standard deviation and the value range of the raw data from *each* accelerometer axis. Classification is done per 30-second epochs based on the characteristics of a larger window centered at these 30-second epochs. Specifically, Biobank identifies stationary periods in 10-second windows where all three axes have a standard deviation of less than $13.0mg$ ($1mg = 0.0098\,m \cdot s^{-2}$). These stationary periods are then used to define whether a window is stationary or not.

C. Signal Data Quality By Granularity

In addition to *qualitative* assessment as discussed in Section III-B, *quantitative* measures that define how much usable data is in a specific period (*i.e.*, **data quality** at different levels) are required before statisticians can begin analysis.

The Data Quality Model. Based on Biobank's non-wear classification on 30-second epoch level, we can further generate data quality that can be used for analysis at different time resolutions. Each phase in our data quality derivation flow is illustrated in Table II to Table V and expanded upon below.

- **Epoch Level** — This table is generated from Biobank's 30-second epoch classification. It serves as the working basis for subsequent data quality tables. Note that we have one additional column, "Subject," to indicate participant ownership of an epoch.
- **Hourly Level** — From the epoch quality table, we can apply a filter to only keep correctly worn epochs and in turn infer hourly data coverage in terms of compliant minutes. This hourly data quality table is the source for data quality reporting at the finest granularity.
- **Daily and Intraday Window Level** — From the hourly data quality table we can summarize the total coverage for each day and produce daily level data quality tables. In addition, for analysis purposes, we are often interested in specific intraday windows from which digital endpoints

TABLE II: EPOCH LEVEL QUALITY.

Subject	Timestamp	Non-wear
1002	2021-09-15 19:15:00	false
...	...	...
1005	2021-10-18 09:45:30	true

TABLE III: HOURLY LEVEL QUALITY.

Subject	Date	Hr	Coverage $(minute)$
1002	2021-09-15	19	45
...	...	...	...
1005	2021-10-18	09	60

TABLE IV: DAILY AND INTRADAY LEVEL QUALITY.

Subject	Date	Coverage $(minute)$	Window
1002	2021-09-15	1440	pa_daily
...	...	...	...
1005	2021-10-18	720	sleep_night

TABLE V: EXTENDED QUALITY WITH EXTERNAL MAPPINGS.

Site	Subject	Date	Trial Day Index	Visit	Coverage $(minute)$	Window
101	1002	2021-09-15	1	0 $(PreTreatment)$	1440	pa_daily
...	...	...	...	...	...	...
103	1005	2021-10-18	32	4	720	sleep_night

are derived — for instance, walking time or step count during the daytime (*i.e.*, daily **p**hysical **a**ctivity) and sleep hours during the nighttime. Thanks to the "Hour" column in the hourly quality table, intraday window coverage can be easily derived by applying filters.

- **Extended Quality with External Mappings** — We can further extend the data quality table with additional mappings when they become available as the study progresses, for instance, mapping between patients and sites/visits, as reported from the clinical operation site. These extra fields allow analysis-specific filtering and aggregation, *e.g.*, to find out which participants have sufficient data and set up individual baselines. We use this table to look for the patients with at least three valid days ($>= 20$ hours of data for a day to be qualified as a valid day) during a pre-treatment visit.

D. Representing Digital Data Quality

Fully understanding the quality of a large dataset, especially one that contains data from wearable device sensors, is not always a trivial undertaking. With numerous considerations to be cognizant of, as discussed in Section III-C, the most logical first step is to present the data with visualizations. Thoroughly understanding the data coverage and quality requires more than one visualization, simply because there is more than one aspect to check. This section presents a family of commonly used visualization examples in our data quality strategy.

- **Identifying Outliers and Missing Data.** Certain metrics must fall between threshold ranges depending on the study and associated data sources. One example is heart rate, which falls within a specified range of 30 to 200 beats/minute for one study. This range is outlined in the DTA for the study and must be applied to all heart rate data points collected. By plotting these signals against the specified thresholds, outliers can be immediately detected by viewing a plot. If outliers exist, further investigation will be completed for that participant's data to see if there are outliers for other metrics. Further, gaps in data can be identified within the same visualization, as demonstrated in Figure 3(a). Detailed data quality reports are generated in conjunction with the visualizations created for displaying outliers and missing data. For example, we convert the signal data from 3(a) to a sequence of colored blocks in Figure 3(b), with green blocks indicating valid sensor signal value in the corresponding period and red indicating missing or invalid signal value identified. In Figure 3(c), we compute the valid data ratio, and therefore can represent the data quality with a numeric value, or with a color from the color palette, keyed to the valid data ratio (see *e.g.*, Figure 3(d)).
- **Data Quality Map with Levels of Detail.** The quality of sensor signal data must be examined on various levels, each offering a specific level of detail. While certain levels are more useful for identifying distinct patterns, we will focus on the hourly, daily, and study levels on both a patient and population level:
  - *Minute-by-Minute Quality Map for a Day* — Examining signals on a minute level can help to identify the minutes where a device may have intermittent connectivity, or more minor issues can be identified and further inspected, as seen in Figure 4(a).
  - *Hour-by-Hour Quality Map for a Trial* — Zooming out, we can look at each hour across all days in the study. The hourly level aggregation mentioned in Section III-C is used to configure the day level plot, shown in Figure 4(b). This figure shows minutes of data coverage for each hour across all study days. This type of visualization allows us to look at compliance trends for a patient that may persist during certain hours of each day. Figure 4(b) shows an interesting device wearing pattern for the participant — taking off the wearable device to charge the battery for a couple of hours in the middle of each day of the trial has resulted in *missing data*, visualized as a sequence

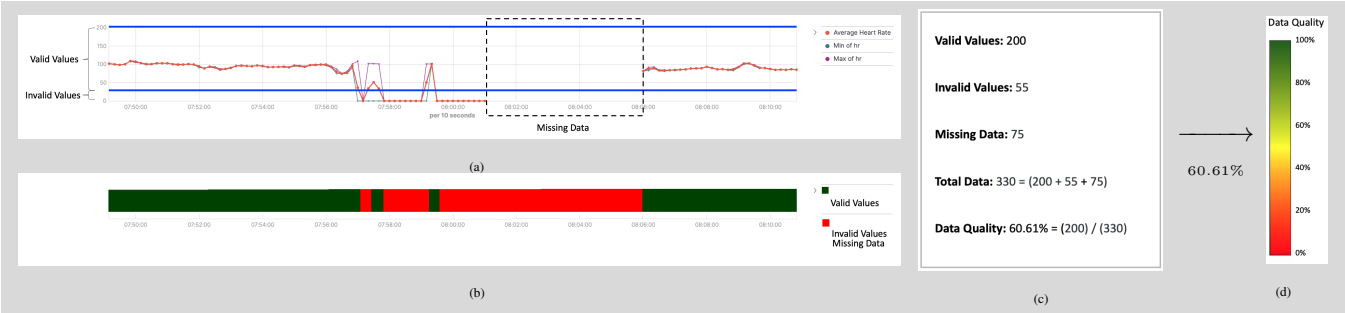

Figure 3: Visualization for sensor data quality. (a) Heart rate data (beats/minute) observed for one participant between 2021-02-15 07:49:00.000 and 2021-02-15 08:11:00.000. Valid range between 30 - 200 beats/minute, as denoted by threshold lines. Invalid data was observed multiple times. Missing data was observed between 2021-02-15 08:01:08.994 and 2021-02-15 08:06:09.000 with nearly 5 minutes of no data. (b) Use colored blocks to represent sensor signal data quality. (c) Deriving numeric representation of the data quality, *i.e.*, valid data ratio. (d) Interpreting data quality with color.

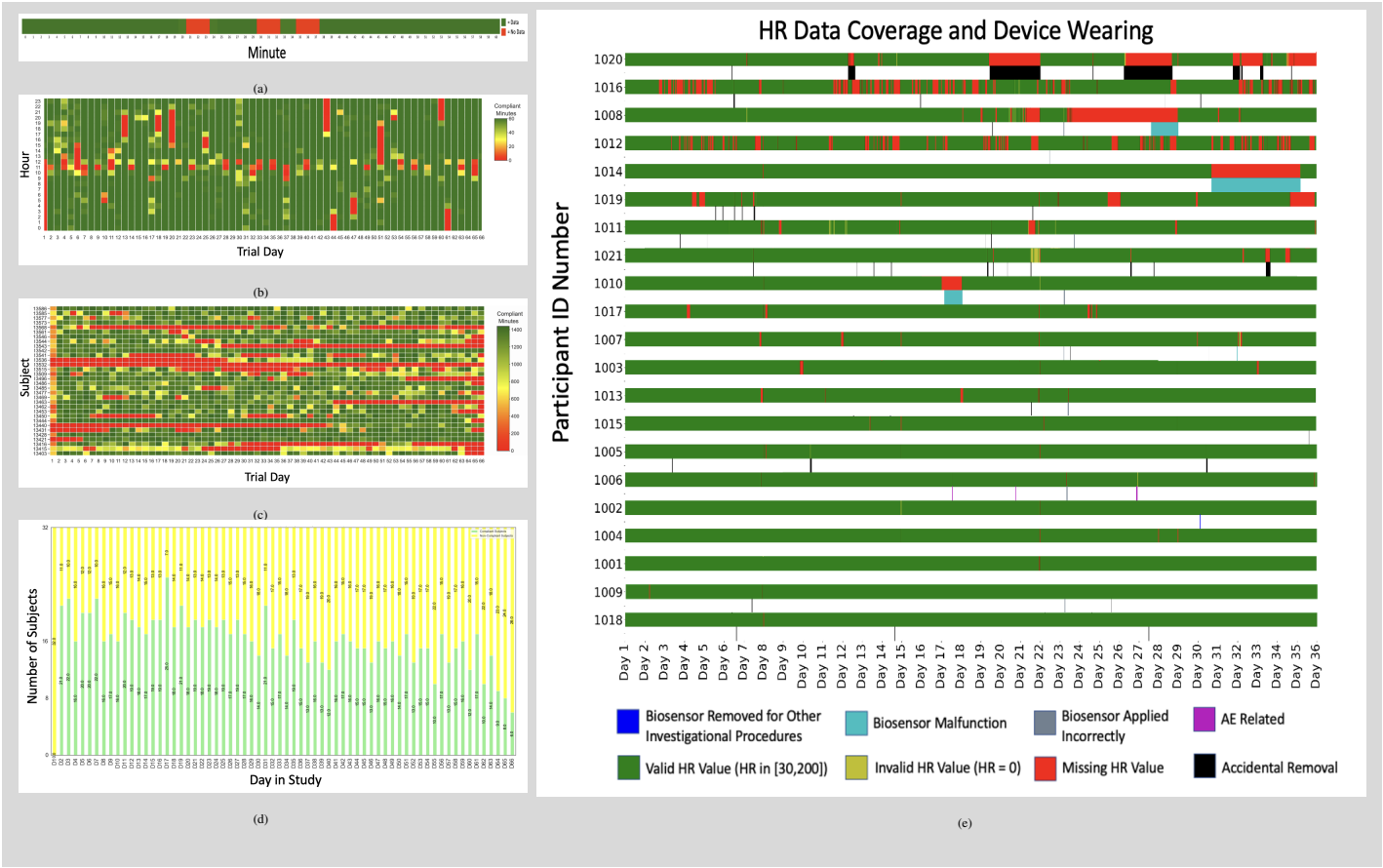

Figure 4: Plots showing (a) minute-level quality representation throughout a participant day, (b) hourly-level quality representation for a participant throughout an entire trial, (c) daily-level quality for a population throughout the entire trial, (d) number of compliant days across all days in a study and (e) data coverage and device wearing issues observed throughout a study.

of red blocks in the center area of the map.

– *Day-by-Day Population-level Quality Map for a Trial* — Plotting data quality for all hours, days, and participants in a study yields the observation of data quality patterns seen in Figure 4(c). This study-level visualization can help us gain insights into the overall data quality at the population level and the compliance trends at the participant level throughout the trials.

– *Compliant Days Throughout a Trial* — In addition to the number of hours per day, it is also useful to view the number of *compliant* In addition to the number of hours per day, it is also useful to view the number of *compliant* days throughout the study, with a definition of compliance dependent on a study's protocol. One

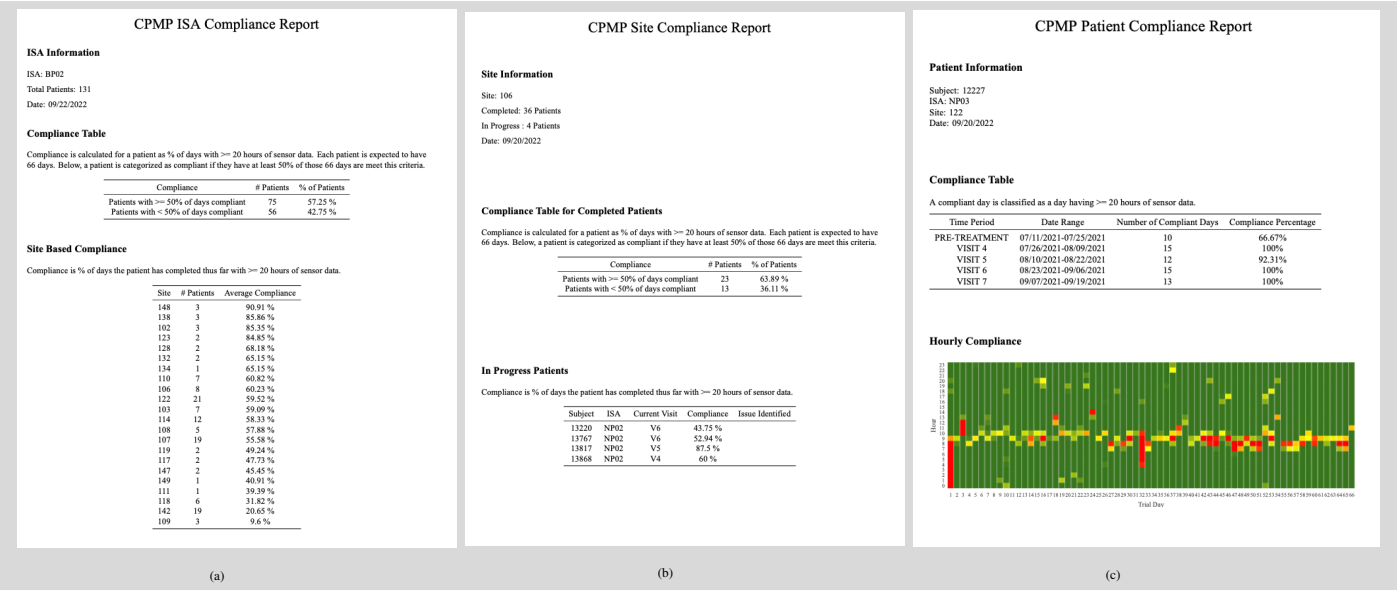

Figure 5: Putting together compliance reports for Intervention-Specific Appendices (ISAs) under Chronic Pain Master Protocol (CPMP). (a) Generated compliance reports on the patient level. (b) Compliance by visit. (c) Customizable compliance report at patient level.

can recognize device-wearing patterns by plotting the number of patients compliant daily in a given study. As seen in Figure 4(d), the number of compliant days in a study decreased due to reduced device wearing as the study progressed.

- **Identifying and Aligning Data Issues.** In many clinical trials, it is a requirement that patients visit a site periodically. Whether it be for receiving dosing of a drug, having their vitals checked, or obtaining a device, information is collected by the sites and stored in various reports. One type of report, device reports, are used during data processing and can help understand the device's overall performance, specifically if any device issues exist. Additionally, information derived from these reports can be used to populate visualizations such as Figure 4(e). By combining this visualization with the information received in site reports, patterns specific to potential device issues and wearing patterns can be derived.
  From the aforementioned data visualizations, various issues and patterns can be identified. When these are paired with actionable recommendations and delivered to the study team promptly, the study team can notify the corresponding site and participant to ensure the issue is rectified. This process leads to a quick turnaround time for potential improvements to data collection and can resolve the challenges that create low compliance in studies.

E. Generating Compliance Reports

Visualizing data is key to understanding data quality, as discussed in Section III-D. However, it is equally important to have a standardized reporting system for compliance to distribute quality and compliance information. Such systems generate reports that outline compliance on three levels: trial, site, and patient. In addition, automated generation allows systems to be configured at the start of a trial and run at set cadences to produce consistent quality assessment reports efficiently.

For each report, regardless of the level or contents, the thresholds used to configure and derive data metrics and visualizations are based on the expectations outlined in the study protocol. Each report aims to give insights into the population's compliance behavior:

- **Trial Summary**: A single comprehensive trial report can be generated and contains metadata regarding the number of patients, sites, and overall compliance percentages.
- **Study-Level Compliance**: A study-level report, such as Figure 5(a), will typically contain metrics displaying overall enrollment and compliance on a site level. These can allow a clinical trial team to gauge the progress of a specific study easily, *i.e.*, the number of patients who have completed their time in the study and the number of patients still in progress.
- **Site-Level Compliance**: Generating reports based on sites, as seen in Figure 5(b), allows clinical teams to efficiently identify which sites may be experiencing issues regarding low compliance across their assigned patients. Typically, site reports contain information for overall performance, with specifics for patients that may fall below a set compliance threshold. The patients with low compliance are labeled with a potential issue- such as low compliance during the nighttime. The potential issues are derived from the hourly compliance for that patient. From here, sites can identify which of their patients contribute most to low compliance and attempt to resolve the issues linked to the low compliance.
- **Patient-Level Compliance**: Reports on a patient level

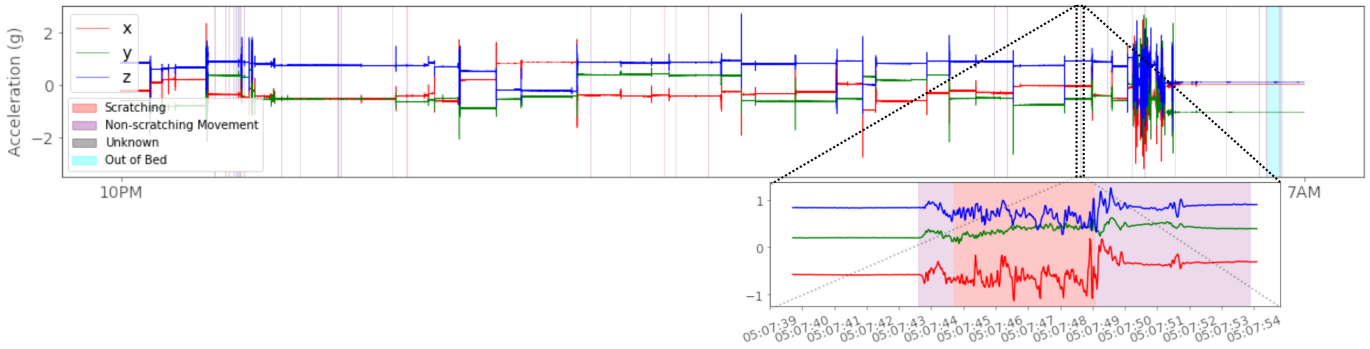

Figure 6: A plot of sensor signals overlaid with annotation labels is used to assess the data quality of annotations in conjunction with sensor signals.

can give insight into their specific patterns of device wearing. In these reports, as seen in Figure 5(c), the number of visits, compliant days within each visit, and compliance percentage per visit are displayed. In addition, an hourly compliance heatmap is visible, allowing for further understanding of when patients wear their devices across the study duration.

F. Data Quality in Novel Digital Endpoint Development

For novel digital endpoint development, raw sensor signals are collected along with annotations or labels, considered the ground truth. Annotations describe events explaining the status of the patient. As such, it is critical to assess the data quality of annotations and sensor signals to identify and address as many defects as possible.

Assessing Annotation Quality. Annotations are typically collected through patient reporting via a survey system or are labeled via software by trained clinicians who observe patient behavior. We first check for defects in the annotations. Defects may include improper data structure, invalid label categories, incomplete annotations, duplicates, and impossibly overlapping annotations. Defects could be caused by bugs in the annotation software or improper training on how to label.

Assessing Annotation Quality with Sensor Signals. Evaluating annotation quality in isolation is insufficient because digital endpoint development requires both annotations and raw sensor signals. So, we must also assess the data quality of annotations and raw sensor signals in conjunction. Therefore, we plot annotated time segments along with raw sensor signals (*e.g.*, Figure 6) to facilitate the data quality assessment.

Discrepancies in the alignment of annotations and raw sensor signals can vary considerably due to time tracking configurations and device properties in each step of the data collection process. Misalignment between annotation and raw sensor signals can be caused by improper device time configuration or the precision of the sensor device's initial time configuration. In addition, if the sensor device's time tracking is not periodically synced, the device's internal Real-time clock (RTC) will slowly drift over time. We measure drift using the sensor signal overlaid with annotation plots. Once the misalignment from the initial configuration time and RTC drift are measured, we align the raw sensor signals to the annotations.

After the annotations and sensor signals have been properly aligned, we observe the plots to identify possible defects in annotation quality. Defects could include improper labels, annotated events that are not apparent in the sensor signals, and time segments that appear to be missing annotations or sensor signals. Specific time segments of concern are selected and validated with the source to determine if further action is needed.

Lastly, depending on study-specific requirements, we may apply other methods to assess data quality. For example, output from movement detection algorithms can be compared to annotated time segments that describe the movement to check annotation validity and coverage. Using various methods to assess data quality from different approaches is essential to maintain the data quality needed for novel digital endpoint development.

Throughout a clinical trial, accessing data quality metrics is critical to upholding our outlined principles. Therefore, in addition to the compliance reports generated, an interactive data quality assessment tool is needed to monitor data quality throughout a trial. We are thus motivated to establish a data platform, *i.e.*, Magnol.Ai, that allows users to customize the plots to view digital data and the associated data quality reports through various lenses, utilizing filters and other user controls. For example, users may want to view the raw sensor data at the scale and resolution they desire, review derived compliance reports on a day, visit, or patient level. Figure 7(a) is such a typical screen image of the dashboard of Magnol.Ai where users can select the level and the metric for which the visualization will show accordingly. A user wants to view compliance for all patients in a study on the visit level, as seen in Figure 7(b). They define *compliance* as having at least 12 hours of data daily, with 3 days each visit comprising a compliant visit. By selecting the compliance type, which in this case is visit, and inputting the number of hours and days for defining compliance, the user can see the population's compliance report with these specific thresholds, as seen in Figure 7(a). Additionally, they can easily compare and contrast different levels and compliance thresholds within Magnol.Ai's dashboard.

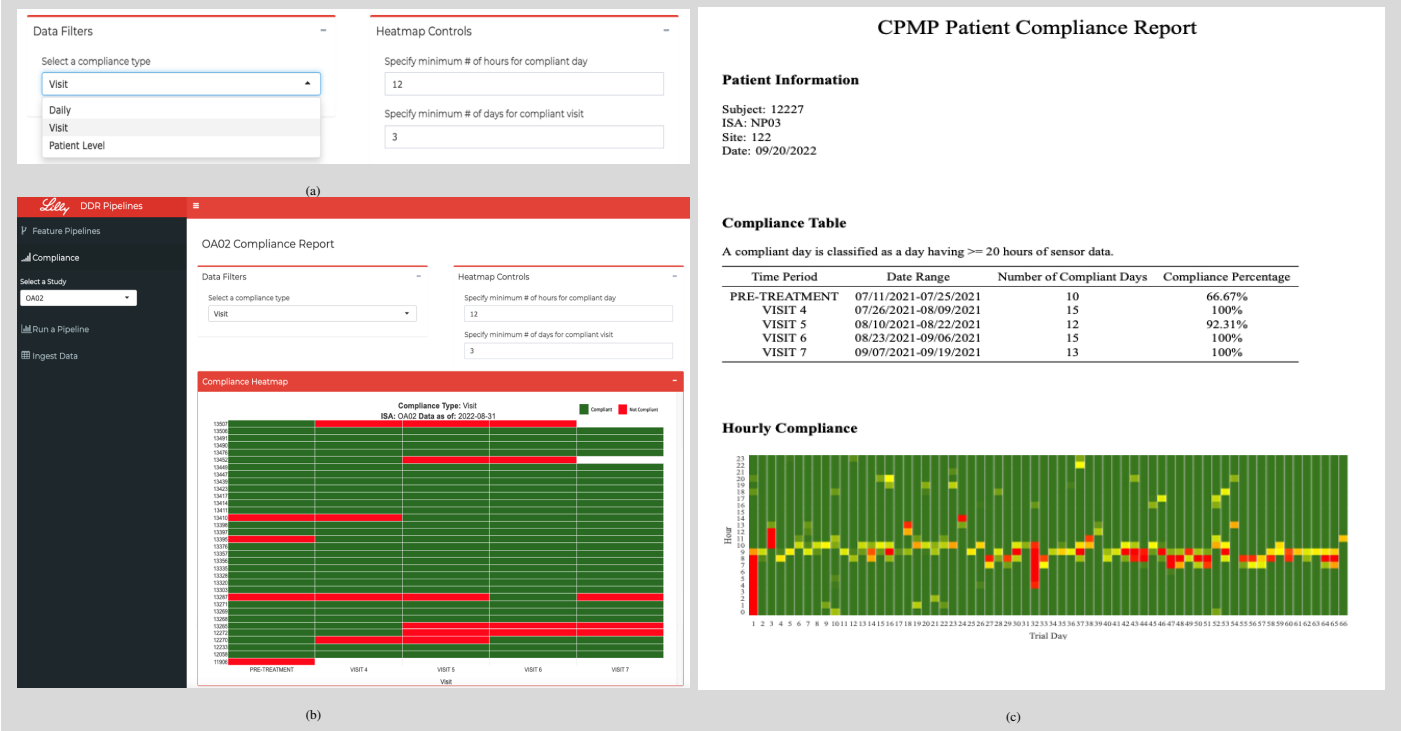

Figure 7: The platform features displaying (a) filters for customizable compliance reports, (b) compliance by visit, and (c) generated compliance reports on the patient level.

In addition to the compliance assessment, data quality visualizations, such as Figure 4, are created and customized within the platform. For example, as seen in Figure 7, a user can select a specific time range or time level to view the data. This zoom in and out can be used to identify and trace patterns of device wearing. The sensor data visualization, supplemented by the data quality assessment capability allows for customizable, real-time, informative visualizations that enable insights into patient compliance and device-wearing data patterns. The study team can process and act upon these key insights with these visualizations housed in a centralized, consistent, and efficient platform.

IV. Magnol.Ai — Putting Tools Together for Digital Biomarker Research

In this section, we focus on a few key building blocks of our data platform, *i.e.*, Magnol.Ai, which continuously ingests, visualizes, and profiles digital data for quality analysis and digital measure derivation. We introduce how we organize the various digital data sets collected from studies, and then how we present these digital data sets with interactive dashboards to help researchers navigate and explore these digital data, view the data quality reports and uncover data insights. Finally we focus on the technologies we leverage for cloudifing, versioning, and parallelizing Magnol.Ai's computing jobs for quality analysis and digital measure derivation.

A. Organizing Digital Data in Magnol.Ai

Organizing and presenting digital data from wearables is vital to the success of using digital technologies in a clinical study, and is a key consideration to regulators. Inside Magnol.Ai these wearable sensor data are stored and organized at study level for authorized visualization and access, with a few key attributes that can be leveraged to further subset or group the digital data in Magnol.Ai's data portal:

- **ISA**: There are cases where an overarching study consists of various unique studies, or a 'suite' of studies. In this case, each individual study is described as being an ISA. Using the aforementioned CPMP study suite, we see that there various ISAs. As seen in Figure 8, Magnol.Ai handles this case by treating each ISA as an individual study, under the umbrella of the overarching CPMP study.
- **Cohort**: A cohort is a grouping of participants in a given study that is specific to an activity or criteria as outlined in the study protocol. For example, in a drug trial, there can be a cohort for each specific drug dosage, as well as a placebo cohort. In an observational study, there can be cohorts that contain subjects who have varying levels of severity for a given disease state, as well as a healthy population cohort.
- **Participant**: Wearable sensor data are collected from participants enrolled in each clinical study. Displaying various sensor data collected from each participant as well as the derived features and data compliance is one typical way for one to explore sensor data using Magnol.Ai's

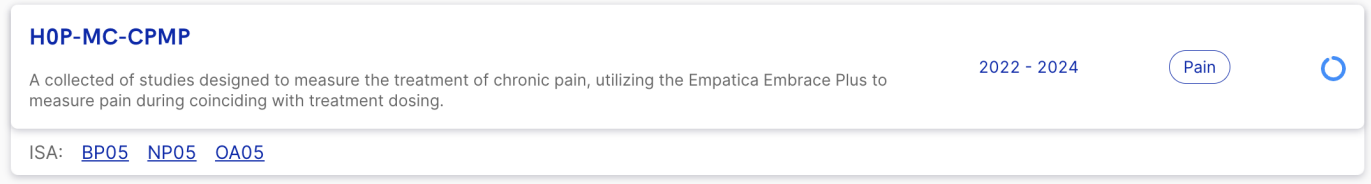

Figure 8: A typical "study suite" view in Magnol.Ai, where there are multiple ISAs, *i.e.*, individual studies, under this overarching "suite".

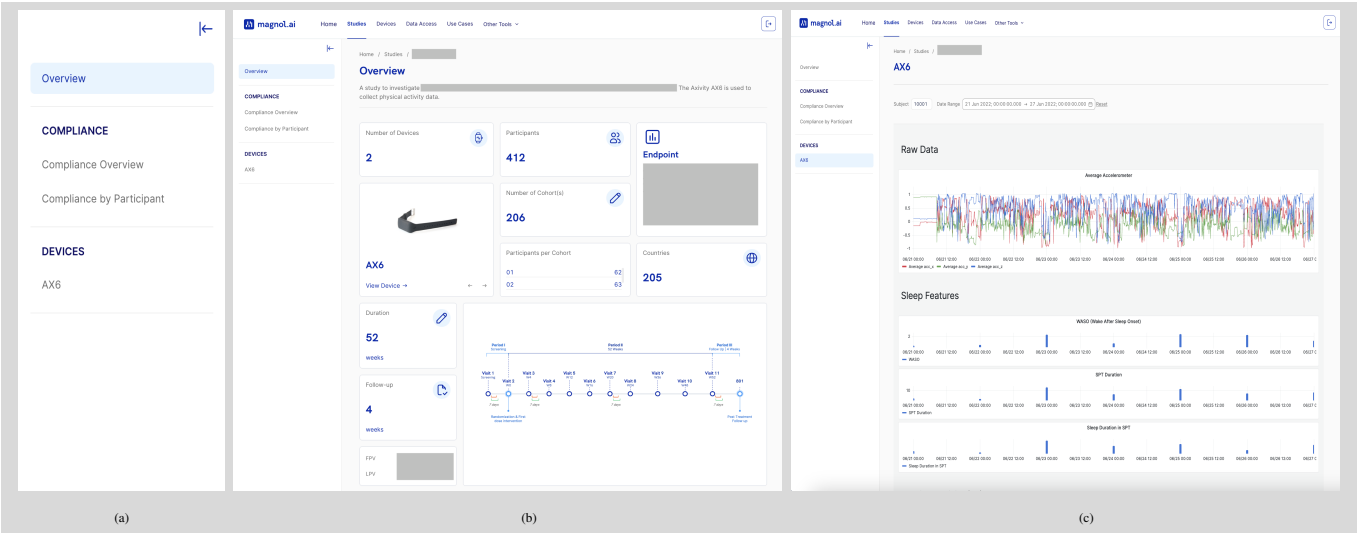

Figure 9: Navigating studies and exploring digital data sets with Magnol.Ai's dashboards. Key visuals: (a) side navigation panel, (b) study overview dashboard example, and (c) sensor data and derived features visualizations.

dashboard (see *e.g.*, Figure 9(c)).

- **Time Range**: Magnol.Ai allows one to view the full spectrum of all digital data collected in studies, or use controls to explore the data at the scale or resolution desired by the user.

By designing Magnol.Ai to work with various organizational hierarchies for study design, it can properly house and display the needed information, including an overview of a study, as seen in Figure 9(b). When beginning data exploration for a given study, users need to know the basic overview of study activities and study population. Magnol.Ai's custom design allows all study overview information to be viewed in one single dashboard. The study overview page contains high-level, universal metrics including the number of devices, endpoint descriptions, and study timeline.

B. Visualizing Wearable Sensor Data and Reporting Quality

In addition to the study overview page, each study contains various dashboards belonging to each type of data, including sensor data, ePRO data, and compliance data. Users can easily navigate to a given dashboard within a study directly from the study overview page. With each tab holding specific information, organizing the information from studies becomes as simple as making a new dashboard for each desired type of data. Taking compliance data as an example, a user would need to see an overview of compliance on a study level, as well as compliance on a participant level. By building dashboards representative of these two aggregation levels, these dashboards can be housed under 'Compliance'.

Compliance Overview The compliance overview dashboard shows aggregated levels of compliance at the study-level. These metrics account for all subjects, regardless of cohort, site, visit, or other grouping and often contain the following:

- *Number of Completed Participants* — The total number of participants who have completed the study.
- *Number of In-Progress Participants* — The total number of participants who are currently in the study.
- *Average Daily Compliance* — The average percentage of daily device wearing.
- *Average Daily Wearing Hours* — The average number of hours where a device was worn.

As shown in Figure 7, various visualizations showing compliance including daily, weekly, or visit heatmaps, distribution plots of average daily wearing compliances, and compliance metrics by site are often included.

Compliance at Participant Level The compliance analysis at participant level displays compliance metrics and detail plots corresponding to each participant with more granularity —- hour by hour throughout the entire trial. Additional filters can be applied including visit information or trial day information. While some kind metrics, such as average daily compliance percentage, are also used on the compliance overview dashboard, visualizations on the 'Compliance by Participant' dashboard

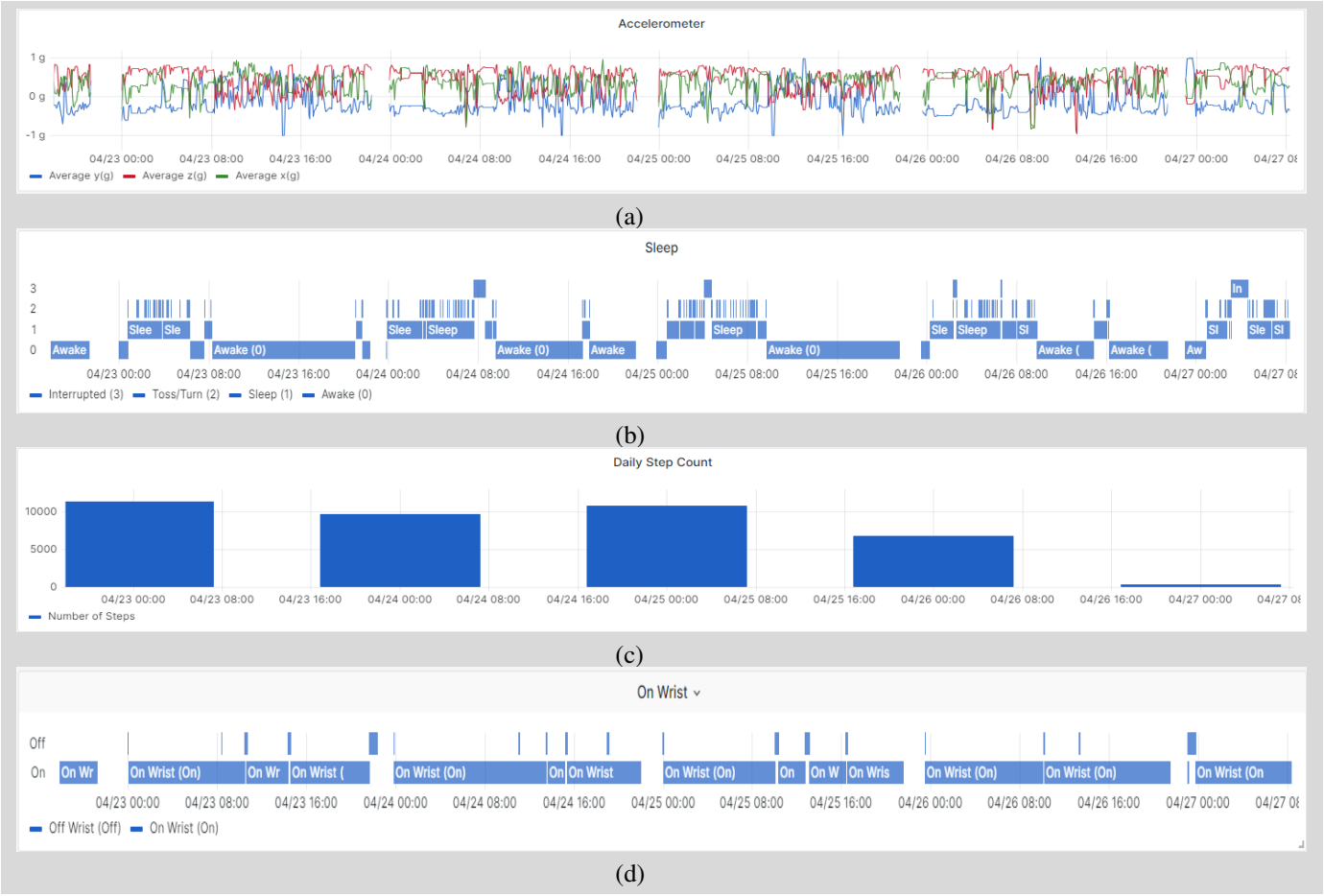

Figure 10: Magnol.Ai offers the power to visualize and directly compare raw sensor signals (a), derived features — sleep measure (b), step count (c), and daily wearing compliance (d) to fully understand the quality of the data.

can show more detailed information about a subject including:

- *Visit Compliance* — With the addition of a visit selection dropdown, users can select a visit and view compliance for each hour during a given visit in a study (Figure 7(b)).
- *Hourly Compliance* — A heatmap showing the number of wearing minutes for each hour across the duration of a study, offering more insights to wearing patterns of a given subject (Figure 7(c)).

Sensor Data Visualization Sensor data can also be organized and displayed similarly to compliance information (see Figure 9(c).) For each device in a study, there is a designated dashboard under the 'Devices' section in the side navigation panel, as seen in Figure 9(a). When navigating to a given dashboard, users can see visualizations of raw sensor data processed sensor data, and the derived features. Across types of visualizations, the data is synchronized to zoom / aggregate to the same level in the same view. This can be accomplished by 1) clicking and dragging to zoom into a specific time range (see *e.g.*, Figure 6), or 2) utilizing the date-picker at the top of the page to select a date on a calendar, or type in a specific date and time (see *e.g.*, Figure 9(c)).

Data Quality The true power of Magnol.Ai comes in the form of visualizing data quality. With the capability to view various types of data aligned to the same date and time range, users can easily identify the story that the data is telling. Examining Figure 10, we see the raw accelerometer data coming from a wrist-worn device. We can compare the raw sensor signals to the derived sleep minute features. Additionally, we can view the device wearing compliance for each day the device was worn, in form of a heatmap. With all three data visualization channels aligned to the same date and time range, users easily view the quality of the data and assess which segments of the data are most useful to analysis.

We have designed Magnol.Ai to view data in this way in order to directly compare data and fully understand the quality of the data. This capability is not limited to only viewing sensor data, but rather, we can apply the same methodology to comparing patient reported events to raw sensor data and derived features, as well as ePRO data, including pain ratings. With the capability to compare virtually any type of data, Magnol.Ai allows unlimited exploration, including data coverage, quality, and compliance.

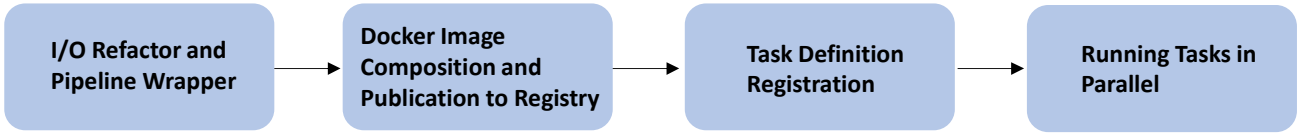

Figure 11: Flowchart of Pipeline Cloudifying Process

C. Cloudifying, Versioning, and Parallelizing Data Pipelines

Motivation Our data processing and digital measure derivation pipelines previously run on company's on-premises High Performance Computing (HPC) systems. As our computational needs grow significantly following limitations are identified:

- *Limited Scalability* — On-premises HPC systems often have fixed hardware resources, which can limit the scalability of data processing pipelines. As data sets grow in size or processing demands increase, it may become challenging to accommodate the additional workload efficiently. This can lead to performance bottlenecks and longer processing times.
- *Dependency Management* — Running pipelines on dedicated on-premises systems might involve managing various dependencies, libraries, and runtime environments manually. Ensuring consistent environments across different systems can be challenging and may lead to compatibility issues and version conflicts.
- *Lack of Elasticity* — On-premises HPC systems have fixed capacity, which means they cannot easily adapt to varying workloads. During periods of low activity, resources may be underutilized, wasting valuable computing power. Conversely, during periods of high demand, the system may struggle to handle the load efficiently, leading to delays in data processing.
- *Reproducibility Challenges* — Reproducing pipeline results can be challenging on on-premises HPC systems due to potential differences in hardware, configurations, and software versions. This lack of standardization can hinder the ability to validate and replicate research findings reliably.
- *High Maintenance Costs* — Maintaining and upgrading on-premises HPC infrastructure can be expensive, requiring significant capital investments and ongoing operational costs. Additionally, specialized personnel are needed to manage and support the infrastructure, which can add to the overall expenditure.

In contrast, containerizing pipelines and migrating to cloud-based environments like AWS using Docker addresses many of these drawbacks. Containerization is a lightweight and portable approach to software development and deployment. It is a method of packaging an application along with all its dependencies, libraries, and configurations into a single unit called a container. This container acts as a self-contained execution environment, allowing the application to run consistently and predictably on any platform that supports the containerization technology. Unlike traditional virtual machines, which require a full operating system for each application, containers share the host OS kernel, making them much more efficient and lightweight. This efficiency not only reduces resource overhead but also facilitates rapid deployment and scaling, making containerization an ideal solution for modern cloud-based architectures.

Cloud-based containerization offers elasticity, scalability, and cost-effectiveness, allowing researchers to efficiently process large datasets, optimize resource utilization, and adapt to varying workloads. Docker's containerization approach ensures consistent environments, simplifies dependency management, and enhances reproducibility, making it easier to validate research findings and collaborate with others seamlessly. Additionally, cloud providers handle infrastructure maintenance, disaster recovery, and offer a pay-as-you-go model, reducing the need for extensive upfront investments and ongoing maintenance costs. In below we detail the advantages of containerization.

Cloudifying Public cloud providers, such as AWS, Google Cloud Platform (GCP), and Microsoft Azure, offer robust support for container-based parallel and distributed processing services. AWS provides Elastic Container Service (ECS) and Elastic Kubernetes Service (EKS), GCP offers Google Kubernetes Engine (GKE), and Azure has Azure Kubernetes Service (AKS) and Azure Container Apps. These cloud-native services enable seamless deployment and management of containerized applications at scale, aligning perfectly with our needs for parallel data processing.

Migrating from on-premises high-performance computing to the cloud aligns with our company's strategy to leverage cloud-based storage and computation. By adopting container-based processing in the cloud, we can take advantage of the cloud's elasticity, scalability, and cost-effectiveness. Cloud infrastructure allows us to dynamically allocate resources based on demand, optimizing utilization and reducing operational costs. Additionally, it reduces the burden of managing and maintaining on-premises hardware, providing us with more flexibility and agility to adapt to changing research requirements and workloads.

Versioning Ensuring the reproducibility of results is crucial in scientific research and data processing. Dockerizing our pipelines allows us to version both the algorithm source code and the entire environment in which the processing takes place. This means that we can track changes to the pipeline code over time, allowing us to roll back to previous versions if needed. Moreover, capturing the entire environment, including

libraries, configurations, and runtimes, guarantees that the data processing pipeline will produce consistent and replicable results, regardless of the underlying infrastructure or platform.

Furthermore, provenance tracking, which involves recording the origin and history of data and processes, is essential for maintaining data integrity and traceability. Docker containers act as self-contained units that encapsulate all the dependencies and configurations required for data processing. By versioning the container images, we can precisely track the exact environment in which data processing occurred, ensuring that any future analysis or audits can be confidently performed based on the same conditions. This level of versioning and provenance tracking enhances the credibility of our research and allows other researchers to reproduce our findings with ease.

Parallelization The data processing pattern for digital measures, such as step count and sleep duration, often follows an embarrassingly parallel computing paradigm. This means that the processing of data for each subject day can be performed independently, without any need for inter-process communication or coordination.

By dockerizing these pipelines and running them in a public cloud environment like AWS, we can easily take advantage of the cloud's ability to handle data processing in parallel and at scale, enabling faster and more efficient analysis of large datasets. To be more specific, docker containerization provides a straightforward way to manage parallel computing tasks efficiently. With the ability to spawn multiple containers simultaneously, we can distribute the data processing workload across a cluster of containers, enabling concurrent execution of tasks. This scalability not only speeds up the overall data processing but also ensures that we can handle large volumes of data without overwhelming the system. Furthermore, container orchestration tools like Kubernetes make it easy to manage the deployment, scaling, and monitoring of containers, simplifying the management of parallel processing in the cloud environment.

Additionally, the elasticity of cloud resources ensures that we can dynamically adjust the number of container instances based on the workload, optimizing resource utilization and cost efficiency.

Pipeline Cloudifying Steps

Figure 11 summarizes the steps of cloudifying a traditional pipeline and below we detail these steps.

- *I/O Refactor and Pipeline Wrapper* — During the process of containerizing the pipeline, a significant aspect that required attention was the handling of file input/output, which originally relied on local storage. To make the pipeline cloud-ready and to ensure seamless data processing in a distributed environment, a crucial refactoring step was undertaken to implement an I/O layer that interacts with cloud storage, specifically an S3 bucket.

  During the I/O refactoring phase, a thoughtful design approach was adopted to seamlessly integrate cloud-based file input/output functionalities without modifying the original pipeline code. The design centered around the creation of a wrapper that serves as an intermediary layer between the pipeline and the cloud I/O services, specifically the S3 bucket. This wrapper encapsulates the necessary code to interact with the cloud storage, enabling the pipeline to leverage the benefits of cloud-based file management while maintaining the integrity of its core functionality. By isolating the cloud I/O layer in the wrapper, the underlying pipeline code remains untouched and agnostic to the storage medium. This design preserves the pipeline's portability and allows it to be run with minimal modifications in diverse computing environments.

  By leveraging Amazon S3's object storage service, the pipeline can now read input data from and write output data to the S3 bucket, providing a scalable and durable storage solution. The refactored I/O layer ensures that the pipeline can effectively handle cloud storage, making it well-suited for deployment in public cloud environments like AWS.

- *Docker Image Building* —

  The Dockerfile presented in Listing 1 serves as a blueprint for building a containerized environment tailored to process sensor data to derive digital measures using GGIR. It starts by utilizing the base image of R, upon which subsequent instructions are layered to set up the required configuration. For instance, setting up GGIR pipeline in R (line 5), installing dependent python libraries required by our driver (line 14). Lastly, the Dockerfile defines the command to run the application (driver 'ggir.py' in our case) upon container launch (line 20). Through this Dockerfile, researchers can create a self-contained environment with all necessary components, enabling seamless and consistent data processing across various platforms and environments. Once Docker image being fully validated, we deposit it to AWS Elastic Container Registry (ECR).

```
1  # Base image for R
2  FROM rocker/r-ver:4.3.1
3
4  # Install GGIR version 2.8-6
5  RUN R -e
       "devtools::install_github('wadpac/GGIR@2.8-6')"
6
7  # Install Python 3.9
8  RUN apt-get -y install python3.9
9
10 WORKDIR /app
11
12 # Install dependencies through requirements.txt
13 COPY requirements.txt .
14 RUN pip install --no-cache-dir -r requirements.txt
15
16 # Copy Python driver program and other files
17 COPY . .
18
19 # Set the entry point as the Python driver
20 ENTRYPOINT ["python", "ggir.py"]
```

Listing 1: Sample GGIR Dockerfile.

- *Task Definition Registration* —

```
family="pipeline-ggir"
task_role_arn="ecs-task"
execution_role_arn="dh-tti-ecs-exec"
# CPU units, 1,024 CPU units per vCPU
cpu=2048
# in MiB
memory=8192
container_name="ggir"
image_uri="dhrd-pipeline-ggir:2.8.6"

response=$(aws ecs register-task-definition \
  --family "${family}" \
  --task-role-arn "${task_role_arn}" \
  --execution-role-arn "${execution_role_arn}" \
  --network-mode awsvpc \
  --requires-compatibilities FARGATE \
  --cpu ${cpu} \
  --memory ${memory} \
  --container-definitions "[{
      "name": "${container_name}",
      "image": "${image_uri}",
      "essential": true,
      "cpu": ${cpu},
      "memory": ${memory} }]" \
  )
```

Listing 2: Sample script to register GGIR task definition.

In the provided sample shell script as shown in Listing 2, the task definition registration is a crucial step in deploying containerized applications on AWS. ECS task definitions define the configuration for individual tasks, specifying essential parameters such as the Docker container image, CPU and memory requirements, network settings, and task roles. By registering a task definition using the AWS CLI (Command Line Interface), we effectively create a blueprint for the containerized application that ECS can use to launch and manage instances of the task. The registration process associates the task definition with a specific family name, such as ”pipeline-ggir” in this case, making it easily identifiable and reusable across ECS services.

In addition, task definitions support the concept of revisions. A revision represents a specific version or iteration of a task definition. Whenever a task definition is updated, either to change container configurations, environment variables, or other parameters, a new revision is created. This approach allows ECS to maintain a historical record of changes to the task definition over time. This helps to maintain a history of configuration changes, which is useful for auditing, rollback purposes, and understanding the evolution of our pipelines over time.

- *Running Tasks in Parallel* — Following the successful registration of the ECS task definition, the next crucial step in our containerized data processing pipeline is to launch ECS tasks for each input/output file pair, as shown in Listing 3. The script employs a for loop to iterate over the lists of input and output file paths. For each pair, a separate ECS task is initiated using the AWS CLI's aws ecs run-task command. The run-task API call submits tasks in an asynchronous fashion, enabling ECS to efficiently run tasks in parallel based on the available resources in the designated cluster, in this case, “dbm-pipeline”. The tasks are executed using the specified Fargate launch type, which manages the underlying infrastructure, allowing us to focus solely on defining the task requirements in the task definition. The ‘–overrides’ flag within the script allows us to pass dynamic input and output directories to the Docker container as command-line arguments. This enables the containerized application to process the corresponding input data from S3 and store the results in the specified output location. By launching tasks in parallel, ECS leverages the underlying cloud infrastructure's scalability, making it a well-suited solution for processing large-scale datasets in an efficient and resource-effective manner.

```
# List of input file paths
input_folders=("s3://...", ...)

# List of output file paths
output_folders=("s3://...", ...)

cluster_name="dbm-pipeline"
task_definition_name="pipeline-ggir"

# Iterate over the input/output lists
# and run the Docker container for each pair
for i in "${!input_folders[@]}"; do
  input="${input_folders[i]}"
  output="${output_folders[i]}"

  # Run the Docker container as a separate ECS task
  aws ecs run-task \
    --cluster $cluster_name \
    --launch-type FARGATE \
    --task-definition $task_definition_name \
    --overrides "{"containerOverrides":
        [{
        "name": "${container_name}",
        "command": [
        "--inputdir", "${input}",
        "--outputdir", "${output}" ]}]}"
done
```

Listing 3: Sample script to run parallel GGIR tasks.

V. Conclusion and Future Work

As DHT continue to evolve and collect more complex digital data in clinical trials, the need for a digital data quality assessment platform is increasing. By defining and implementing the fundamentals of data quality into the digital data quality framework and platform, we can generate automated compliance reports, customizable visualizations, and real-time quality metrics. In addition, the methods for facilitating dBMs research have been simplified with the centralized digital data quality assessment platform. As dBMs research continues, so will the use of the digital data quality assessment platform. Future directions include the use of visual mining and data mining technologies to help identify data quality in a novel way to facilitate data quality assessment.

References

[1] H. Zhang, R. Giesting, L. Miller, G. Ruan, N. Patel, J. Ji, T. Zhang, and Y.-L. Yang, “A framework for digital data quality assessment in digital biomarker research,” in *The Ninth International Conference on Big Data, Small Data, Linked Data and Open Data (ALLDATA 2023)*, 2023.

[2] J. Wright, O. Regele, L. Kourtis, S. Pszenny, R. Sirkar, and C. Kovalchick, “Evolution of the digital biomarker ecosystem,” *Digital Medicine*, vol. 3, no. 4, pp. 154–163, 2017.

[3] R. Y. Wang, V. C. Storey, and C. P. Firth, “A framework for analysis of data quality research,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 4, pp. 623–640, 1995.

[4] A. Sharma, R. A. Harrington, M. B. McClellan, M. P. Turakhia, Z. J. Eapen *et al.*, "Using digital health technology to better generate evidence and deliver evidence-based care," *Journal of the American College of Cardiology*, vol. 71, no. 23, pp. 2680–2690, 2018.

[5] R. Lyons, G. R. Low, C. B. Congdon, M. Ceruolo, M. Ballesteros, S. Cambria, and P. DePetrillo, "Towards an extensible ontology for streaming sensor data for clinical trials," in *Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, 2021, pp. 1–6.

[6] C. M. Rey, "Wearable data revolution: Digital biomarkers are transforming research, promising a revolution in healthcare," *Clinical OMICs*, vol. 6, no. 2, pp. 10–13, 2019.

[7] I. Clay, "The future of digital health," *Digital Biomarkers*, vol. 4, no. 1, pp. 1–2, 2020.

[8] M. Chen and M. Decary, "Artificial intelligence in healthcare: An essential guide for health leaders," in *Healthcare management forum*, vol. 33, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2020, pp. 10–18.

[9] S. M. Hossain, T. Hnat, N. Saleheen, N. J. Nasrin, J. Noor, B.-J. Ho, T. Condie, M. Srivastava, and S. Kumar, "Mcerebrum: A mobile sensing software platform for development and validation of digital biomarkers and interventions," ser. SenSys '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1–14.

[10] A. Dillenseger, M. L. Weidemann, K. Trentzsch, H. Inojosa, R. Haase, D. Schriefer, I. Voigt, M. Scholz, K. Akgün, and T. Ziemssen, "Digital biomarkers in multiple sclerosis," *Brain Sciences*, vol. 11, no. 11, pp. 1519–1544, 2021.

[11] M. M. Rahman, V. Nathan, E. Nemati, K. Vatanparvar, M. Ahmed, and J. Kuang, "Towards reliable data collection and annotation to extract pulmonary digital biomarkers using mobile sensors," in *Proceedings of the 13th EAI International Conference on Pervasive Computing Technologies for Healthcare*, 2019, pp. 179–188.

[12] A. Doherty, D. Jackson, N. Hammerla, T. Plötz, P. Olivier, M. H. Granat, T. White, V. T. Van Hees, M. I. Trenell, C. G. Owen *et al.*, "Large scale population assessment of physical activity using wrist worn accelerometers: the uk biobank study," *PloS one*, vol. 12, no. 2, p. e0169649, 2017.

[13] C. Sudlow, J. Gallacher, N. Allen, V. Beral, P. Burton, J. Danesh, P. Downey, P. Elliott, J. Green, M. Landray *et al.*, "Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age," *PLoS medicine*, vol. 12, no. 3, p. e1001779, 2015.