

Novel accelerometer-based markers for physical activity in amyotrophic lateral sclerosis based on an open-source summary metric

Elisabeth Niederbacher

5298555

Master's thesis

Supervisors:

Prof. Dr. ir. René Eijkemans – UMC Utrecht

Dr. Ruben van Eijk – UMC Utrecht

Jordi van Unnik – UMC Utrecht



Applied Data Science

Utrecht University

July 2022

Abstract

Accelerometer-based physical activity monitoring has previously been shown to be an objective method for quantifying disease progression in patients with amyotrophic lateral sclerosis (ALS). Currently, accelerometer-based endpoints are based on Activity Counts (AC), a metric provided by accelerometer manufacturers to summarize raw accelerometer data into epochs. However, Activity Counts are proprietary and not calculated the same across manufacturers, making results more difficult to reproduce, less comparable, and potentially slowing down ALS research.

As a result, this thesis project investigates whether markers based on the open-source summary metric Activity Index (AI) can be used to quantify disease progression and thus serve as an alternative to AC-based markers.

This research made use of an existing data set from a previously conducted longitudinal cohort study. In the study, 42 ALS patients wore the ActiGraph during waking hours for seven days every 2-3 months. In addition, they provided additional information about their daily functioning as measured by the ALS functional rating score (ALSFRS-R). The raw accelerometer data was pre-processed, and 61 modified AI metrics for ten different epoch lengths were calculated. These metrics were used in the calculation of three marker types. In total, 183 different markers were defined, each of which aimed to reflect the physical activity of a patient. Finally, the associations of these markers with clinical markers of disease severity, the longitudinal rates of decline, and the associations of AI-based markers with AC-based markers were investigated.

All markers were moderate to strongly associated with the ALSFRS-R total score (0.73, 95% CI 0.56 – 0.88, $p < 0.001$), and their correlations increased with increasing epoch length. Most markers showed less between-patient variability over time than the ALSFRS-R total score (coefficient of variation 0.80 – 1.18 vs. 1.06, respectively). In comparison to some AC-based markers, AI-based markers had slightly lower correlations and a higher between-patient variability over time. AI-based markers were strongly correlated with the AC-based markers.

In conclusion, this thesis project demonstrated that activity markers based on AI metrics can be used to quantify disease progression in patients with ALS and have the potential to be an alternative to markers based on Activity Counts.

Contents

1 INTRODUCTION	5
2 METHODOLOGY	7
2.1 STUDY DESCRIPTION	7
2.2 DATA ACQUISITION	7
2.2.1 Raw triaxial accelerometer data	7
2.2.2 Activity Count data	7
2.2.3 Additional information.....	7
2.3 ACTIVITY COUNT DATA ANALYSIS	8
2.4 RAW ACCELEROMETER DATA PRE-PROCESSING USING AI METRIC	8
2.4.1 File review and missingness	8
2.4.2 Creating epoch-level data using AI.....	9
2.5 AI-BASED MARKERS FOR PHYSICAL ACTIVITY	11
2.6 LONGITUDINAL RATES OF DECLINE IN AI-BASED MARKERS OR DISEASE PROGRESSION.....	12
2.7 LONGITUDINAL CORRELATIONS BETWEEN AI-BASED MARKERS AND CLINICAL MARKERS OF DISEASE SEVERITY	12
2.8 LONGITUDINAL CORRELATIONS BETWEEN AI-BASED MARKERS AND AC-BASED MARKERS.....	13
2.9 FINAL SELECTION OF MARKERS	13
3 RESULTS.....	14
3.1 OVERVIEW OVER FINAL DATA SETS	14
3.2 LONGITUDINAL RATES OF DECLINE IN AI-BASED MARKERS OR DISEASE PROGRESSION.....	14
3.3 CORRELATIONS BETWEEN AI-BASED MARKERS AND CLINICAL MARKERS OF DISEASE SEVERITY	15
3.4 LONGITUDINAL CORRELATION BETWEEN AI ACTIVITY MARKERS AND AC-BASED MARKERS	16
3.5 FINAL MARKERS.....	16
4 DISCUSSION AND CONCLUSION	18
REFERENCES	20
APPENDIX A: R PACKAGES.....	22
R PACKAGES.....	22
APPENDIX B: AC-DATA R-CODE	23
ACTIVITY COUNT DATA PRE-PROCESSING AND ANALYSIS	23
APPENDIX C: AI-DATA R-CODE	35
MISSINGNESS	35
VAN HEES NON-WEAR ALGORITHM.....	37
SYSTEMATIC NOISE CALCULATION.....	38
EPOCH-LEVEL AI DATA.....	41
DEFINING AI-BASED MARKERS AND GENERATING ANALYSIS SET	45
LONGITUDINAL RATES OF DECLINE IN ACTIVITY MARKERS OR DISEASE PROGRESSION	51
LONGITUDINAL CORRELATIONS BETWEEN AI-BASED MARKERS AND CLINICAL MARKERS OF DISEASE SEVERITY	53
LONGITUDINAL CORRELATION BETWEEN AI-BASED MARKERS AND AC-BASED MARKERS	55
FINAL SELECTION OF MARKERS.....	57
APPENDIX D: SUPPLEMENTARY DATA	60
RAW ACCELEROMETER FILES FOR ANALYSIS.....	60
OVERVIEW OVER SYSTEMATIC NOISE.....	61
AI CONTRIBUTION WEIGHTS	62

FULL LIST OF LONGITUDINAL RATES OF DECLINE, BASELINE VALUES AND CoV	64
FULL LIST OF LONGITUDINAL CORRELATIONS OF FINAL MARKERS WITH CLINICAL MARKERS OF DISEASE SEVERITY	67
APPENDIX E: SUPPLEMENTARY TABLES, FIGURES AND EQUATIONS	70
ADDITIVITY OF AI METRIC.....	70
BOXPLOTS OF BASELINE VALUES AND SLOPES OF THE THREE MAIN MARKER TYPES AND CLINICAL MARKERS FOR D1-D10.....	71
BOXPLOT OF CORRELATIONS OF THE THREE MAIN MARKER TYPES WITH A1 MARKER FOR D1-D10.....	72
TABLES CONTAINING FINAL SELECTION OF UPPER AI AND 90P MARKERS M6-M14 AND THEIR CORRELATIONS WITH DISEASE PROGRESSION AND CoV	73
BOXPLOT OF CORRELATIONS OR MARKERS M1-M14 WITH AC-BASED MARKERS	75

List of Figures

FIGURE 1: MISSINGNESS FOR ID1, ID3, AND ID5 IN MEASUREMENT 2.	8
FIGURE 2: A SINGLE MEASUREMENT SUMMARIZED WITH AI METRIC INTO THREE DIFFERENT EPOCH LENGTHS... ..	10
FIGURE 3: DENSITIES OF MARKERS ON ORIGINAL SCALE AND ON THE LOG SCALE.	11
FIGURE 4: BOXPLOT OF MARKERS FOR ID2 MEASUREMENT 2 AND EPOCH LENGTHS 10-SEC, 30-SEC AND 60-SEC 12	
FIGURE 5: BOXPLOT OF CoV FOR THE THREE MAIN MARKER TYPES FOR D1-D10	15
FIGURE 6: BOXPLOT OF CORRELATIONS OF THE THREE MAIN MARKER TYPES WITH THE ALSFRS-R.	15
FIGURE 7: LONGITUDINAL CORRELATIONS OF THE THREE MAIN MARKER TYPES WITH % ACTIVE MARKER	16
FIGURE 8: CORRELATIONS OF MARKERS M1-M14 WITH AC-MARKERS A1 AND % ACTIVE	75
FIGURE 9: BOXPLOT OF BASELINE VALUES FOR THREE MAIN MARKER TYPES	71
FIGURE 10: BOXPLOT OF SLOPES FOR THREE MAIN MARKER	71
FIGURE 11: LONGITUDINAL CORRELATIONS OF THE THREE MAIN MARKER TYPES WITH A1 MARKER.....	72

List of Tables

TABLE 1: SUMMARY OF WEAR TIME IN MODIFIED AI DATA SETS.	14
TABLE 2: FINAL SELECTION OF MEAN AI MARKERS.....	16
TABLE 3: LONGITUDINAL CORRELATIONS BETWEEN DISEASE PROGRESSION AND MARKERS M1 -M5 (MEAN AI) 17	
TABLE 4: FINAL SELECTION OF UPPER AI AND 90P MARKERS	73
TABLE 5: LONGITUDINAL CORRELATIONS WITH DISEASE PROGRESSION AND CoV FOR MARKERS M6 -M9	73
TABLE 6: LONGITUDINAL CORRELATIONS WITH DISEASE PROGRESSION AND CoV FOR MARKERS M10 -M14.....	74

1 Introduction

Amyotrophic lateral sclerosis, short ALS, is a progressive nervous system disease that affects nerve cells in the brain and spinal cord, resulting in muscle control loss (1). The debilitating and immobilizing nature of the disease makes travel, communication, and clinic visits difficult (2), preventing patients from participating in clinical trials (3). Furthermore, ALS disease progression is highly heterogeneous (4). Because clinical visits are no longer required, remote monitoring of physical activity using triaxial accelerometers is a promising method for tracking disease progression in these patients.

Accelerometers are portable devices that are commonly used in clinical and epidemiological research to measure physical activity in greater detail (5). Triaxial accelerometers measure the acceleration caused by body movements along three axes: horizontal right-left (X), vertical (Y), and horizontal front-back (Z) (6). The devices measure acceleration in G-forces “g” $1g = 9.81ms^{-2}$ at a sub-second level (e.g., 30Hz). Accelerometers are particularly effective at measuring dynamic physical activities and provide a more comprehensive picture of health status in real-time than in-clinic assessments (7). Furthermore, accelerometry data is free of information bias introduced by interviewers or patients, making them a reliable measure of physical activity in free-living conditions (6).

Researchers investigated remote, accelerometer-based physical activity monitoring in ALS patients for the first time in 2019 (3). The data set used by the researchers (van Eijk et al. (3)) is the same data set used in this thesis project. The authors proposed four accelerometer-based endpoints based on the Activity Count metric with low-frequency extension (LFE) (8) generated by the ActiLife (version 6.13.3) software. %Active, the proportion of counts that exceeded the 100 counts per minute threshold, and MET, the counts translated into Metabolic Equivalent of Task estimates (9), measured daily physical activity. The daily VM marker, the average multiplied by the standard deviation of the vector magnitude counts estimated on a log scale, and the very similar daily A1, which is based solely on the variation in the vertical axis, were used to assess a patient’s physical capability. The endpoints were strongly associated with the ALSFRS-R but showed less variation over time than the clinical marker ALSFRS-R, which could positively affect sample size calculations and increase the sensitivity to detect differential disease progression (3).

The Activity Count metric (10) used by van Eijk et al. is a summary metric for raw accelerometer data provided by accelerometer manufacturers for the purpose of measuring physical activity and intensity. The counts are calculated by grouping raw accelerometer data into epochs (11). The counts’ values are determined by the sampling frequency and the intensity of the raw data (10). Although ACs are widely used in health research, many researchers have recently raised concerns about this metric (12) (13) (14) (15). The primary issue with ACs is that they are proprietary and not calculated the same across manufacturers and device models, causing confusion among researchers and making results less comparable across studies and more difficult to reproduce (12) (14). Furthermore, documentation of how ACs are calculated is generally incomplete (14). The numerous limitations (13) (12) (14) emphasize the need for an easy-to-access, open-source summary metric to ensure consistency across studies and results.

The summary metric “Activity Index” (hereinafter AI) is an open-source and user-friendly alternative to ACs. AI, like AC, summarizes triaxial accelerometry data into epochs and was developed by Bai et al. in 2016 (12). It is based on the Activity Intensity (AI0) metric (16). AI uses the variability of the device acceleration minus systematic noise and is expressed in “g”, the same unit as raw accelerometer data. AI has favorable properties such as additivity and rotational invariance (12). Additivity means that the AI for shorter epochs is added together to generate AI in longer epochs. This property ensures that the metric is comparable and generalizable across studies. Moreover, AI is rotationally invariant, which means that it remains unchanged if a patient performs the same physical activity with a rotated accelerometer. This is important because accelerometers can tilt when they are being equipped (12). AI outperformed AC, AC (LFE) (8), and ENMO (17) in terms of sensitivity to sedentary and light

activities. This demonstrates AI's advantage in studies with older adults (12) and ALS patients who are immobile. Furthermore, AI had a better predictive performance for classifying an epoch into various physical activity intensity categories compared to the other metrics (12).

As a result, the goal of this research is to identify novel markers for physical activity in ALS patients based on the AI metric and to investigate the relationship between the newly defined AI-based markers and disease progression. Moreover, the research aims to show the differences between markers based on the AC metric and markers based on AI in terms of correlation with disease progression and between-patient variability over time. The potential advantages and disadvantages of the new AI metric over the existing AC metric will also be discussed.

To pre-process and analyze the data, R version 4.1.1 (2021-08-10) was used. "Appendix A" lists the R packages used in the analysis. "Appendix B" contains the complete reproducible R-Code for pre-processing AC data and statistical analysis of AC-based markers. In addition, "Appendix C" contains the R-Codes for pre-processing AI data and statistical analysis of AI-based markers.

Research question:

To what extent can markers based on the Activity Index metric be used to quantify disease progression in patients with amyotrophic lateral sclerosis and thus be considered an alternative to AC-based markers?

2 Methodology

2.1 Study description

The data stems from a longitudinal cohort study by van Eijk et al. (3). The trial participants were drawn from the Treatment Research Initiative to Cure ALS (TRICALS) database. Patients had to have a diagnosis within the Motor Neuron Disease spectrum. From October 2016 to November 2018, 42 Dutch patients took part in the study. Patients were sent the ActiGraph GT9X Link, a small triaxial accelerometer, by mail. Ten different serial numbers were used in the study. Participants were asked to wear the device on their right hip during waking hours for seven days every 2 to 3 months in order to obtain a maximum of seven measurements, i.e., visits, per patient. They also provided information about the severity of their disease (ALSFRS-R score (18)). Data were collected at a sampling rate of 30Hz. The Medical Ethics Committee of the UMCU (16/606) approved the study. All study participants consented to participate and gave their written informed consent to be approached digitally for research purposes.

2.2 Data acquisition

2.2.1 Raw triaxial accelerometer data

The raw accelerometer data collected from the aforementioned study was extracted from the ActiGraph using the ActiLife software (version 6.13.3) and distributed to the students via SURFfilesender (19). This software allows sending large files such as research data. SURFfilesender stores files on servers in the Netherlands for no more than 21 days and is the recommended data-sharing tool by the UMC Utrecht. The servers are subject to European privacy legislation (GDPR) (20). The students received the data in a GT3X file format. In total, 209 data files from 42 patients were given to the students. Students were asked to treat all data as confidential. Other data security measures were not implemented on their personal computers.

2.2.2 Activity Count data

The raw triaxial accelerometer data was summarized into counts for 10-s epochs with the application of the low-frequency extension (LFE) algorithm (8). The aggregated count data was shared with the students via SURFfilesender. 207 data files from 42 patients were given to the students.

2.2.3 Additional information

An Excel spreadsheet containing additional information about the patients' wear time and disease severity was sent via email. The sheet included each measurement's start and end date, the start and end time, and clinical markers of disease severity, i.e., ALSFRS-R, MOTOR, RESPI, and BULBAR scores. The data on the spreadsheet was anonymized. Students did not have access to the names of the patients. Birth and death dates were also not accessible to prevent re-identification of the patients.

2.3 Activity count data analysis

The R-Code for the pre-processing of the count data and the generation of the analysis set, including the calculation of the activity markers %Active and A1, was provided by the project supervisors Ruben van Eijk and Jordi van Unnik and is included in Appendix A. In addition, the paper by van Eijk et al. (3) contains a detailed explanation of the analysis.

2.4 Raw accelerometer data pre-processing using AI metric

2.4.1 File review and missingness

Before working with the raw accelerometer data, the sent files were compared to the information in the additional Excel sheet. Two of the 209 GT3X files did not contain any data. As a result, those files were excluded from the analysis..

Next, all files were checked for missingness. Since the accelerometers were sent by mail and continued measuring while in transit, missing data outside of the measurements' start and end times specified in the additional Excel sheet was excluded. During the wear period, 1192700790 observations, or 460 days, were missing. Around 14.8 hours of data were missing on average per patient per day. There were 243 hours missing during the day (10AM-10PM), and 217 hours missing at night (10PM-10AM). Figure 1 shows a lot of missing data during the day. However, the number of consecutive missing observations is only a small fraction of the number of consecutive missing values at night. The mean number of consecutive missing observations in the times series during the day is 2849 (1.6 minutes, SD: 7.2 minutes) and 11870 (6.6 minutes, SD: 56.8 minutes) at night, which is more than four times as much.

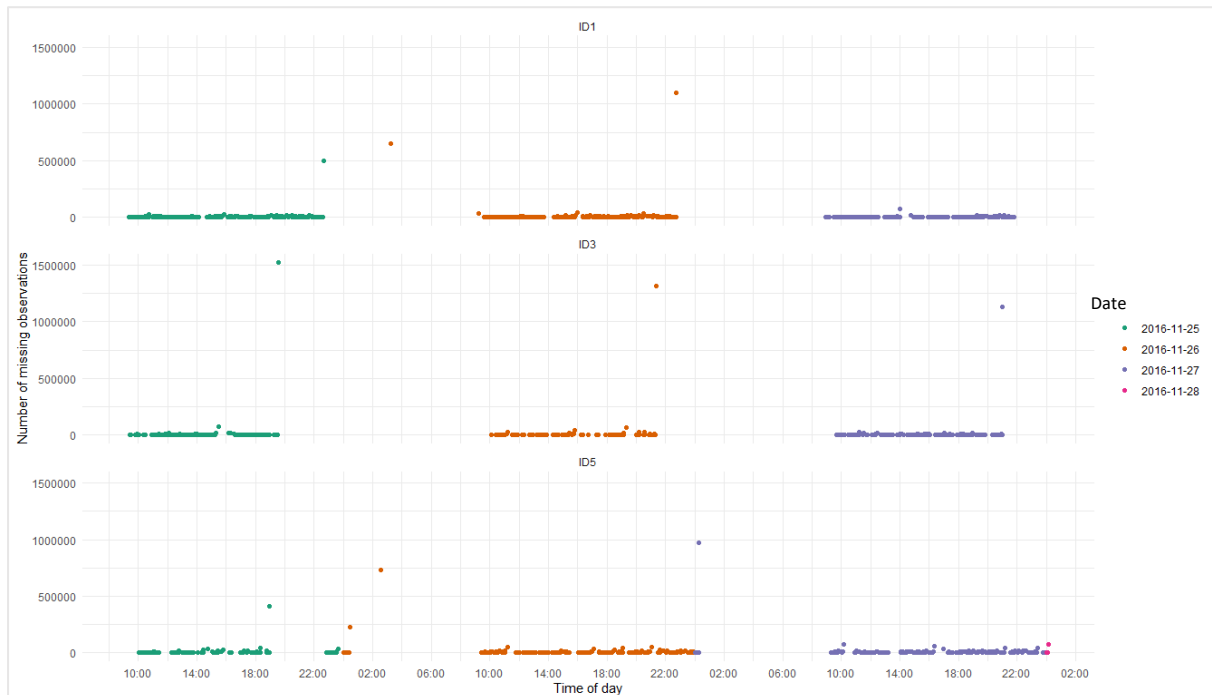


Figure 1: Missingness for ID1, ID3, and ID5 in measurement 2.

Because patients were instructed to wear the device during waking hours, it makes sense that the majority of consecutive missing data comes from the night due to the non-wear times. During the day, missing data may also be due to non-wear times, or the device failed to collect data due to a flat battery,

water damage, or other damages (21). However, because the data was collected in free-living conditions and patients did not keep a daily diary of their wear time, determining which scenario caused daytime missingness is difficult.

A list of the raw accelerometer files can be found in "Appendix D". "Appendix C" contains the R-Code for the missing value analysis as well as the pre-processing steps in section 2.4.1.

2.4.2 Creating epoch-level data using AI

Raw accelerometer data contains millions of observations per measurement. For example, a sampling rate of 30Hz generates 30 observations per second, totaling 1800 observations in just one minute. Therefore, before exploring and working with the data, it is useful to summarize the data into epochs using a summary metric. The original AI metric (12) for patient i , at a specific time point t in the window of length H ($H = \text{epoch length} \cdot \text{sampling rate}$), was defined as:

$$AI_i(t; H) = \sqrt{\max\left(\frac{1}{3}\left\{\sum_{m=1}^3 \sigma_{im}^2(t; H) - \bar{\sigma}_i^2\right\}, 0\right)}$$

Equation 1: Absolute AI

In Equation 1 m are the three accelerometer axes x ($m=1$), y ($m=2$) and z ($m=3$). $\sigma_{im}^2(t; H)$ is the variance of participant i 's acceleration signal for axis m in the window of length H beginning at time point t . $\bar{\sigma}_i^2$ is the noise variance when the device is placed steady which can be calculated using raw non-wear time data. The systematic standard deviation was formally defined as: For every $t/H \in \mathbb{Z}$,

$$\bar{\sigma}_i = \frac{\sum_{t \in \tau_i \text{ and } t/H \in \mathbb{Z}} \sqrt{\frac{1}{3}\{\sigma_{i1}^2(t; H) + \sigma_{i2}^2(t; H) + \sigma_{i3}^2(t; H)\}}}{|\{t \in \tau_i \text{ and } t/H \in \mathbb{Z}\}|}$$

Equation 2: Systematic standard deviation

In Equation 2 τ_i represents the collection of time points t when the device is not moving.

In this research the raw accelerometer data were summarized into epochs using a modified version of the original AI (Equation 1). The modified version of AI was defined as:

$$AI_i^{mod}(t; H) = \sqrt{\max\left(\left\{\sum_{m=1}^3 w_m (\sigma_{im}^2(t; H) - \bar{\sigma}^2)\right\}, 0\right)}$$

Equation 3: Modified AI

Here $w_m \in [0,1]$, $m = 1,2,3$ and the sum of all three axis weights must be one, so $w_1 + w_2 + w_3 = 1$. w_m represents the weight with which each axis contributes to the index. In the original version of the AI metric (12) the weights were set to a fixed value of one-third $w_1 = w_2 = w_3 = 1/3$, implying an equal contribution from all axes and ensuring rotational invariance. Each axis, however, may contribute differently to capturing disease progression. For example, vertical activity (movement against gravity) may be more associated with disease progression than other axes (3). The modified AI allows for experimenting with the weighing of axes during this research.

Another difference between equations 1 and 3 is that a study-specific sigma $\bar{\sigma}$ (12) was used in the modified AI. For simplicity, the modified AI will be referred to as Activity Index or AI in the following sections.

For the AI metric, it is essential to calculate the systematic noise $\bar{\sigma}_i$. Because the noise is calculated using non-wear times, the van Hees raw non-wear algorithm (17) with optimized hyperparameters (22) was applied to each measurement. The algorithm estimates wear and non-wear times based on the standard deviation and the value range of each of the three accelerometer axes in a 135-minute interval. The R-Code of this algorithm is an adaption Shaheen Syed’s Python-Code (23) and can be found in “Appendix C”. Unfortunately, due to the missingness in the data, non-wear periods could only be identified for 8 out of 207 measurements. Luckily, each of the eight measurements came from a unique ActiGraph serial number. As a result, the systematic standard deviation was calculated for eight non-wear periods using the Sigma0 function from the PhysicalActivity R package (24). Because each serial numbers $\bar{\sigma}_i$ was relatively close to the others, the serial number-specific systematic noises could be combined into a single study-specific sigma $\bar{\sigma}$ (12). The values for serial number-specific systematic noises and study-specific noise are listed in “Appendix D”.

Before calculating the AI for each measurement, times outside the measurement time frame specified in the additional Excel sheet were removed. Afterward, time gaps were imputed using the `g.imputeTimegaps` function from the GGIR package (25) to ensure that the wear time adherence results matched those published by van Eijk et al. (3) and to make a summary into epochs possible. The imputation function removes all observations with a zero in each of the three axes and then imputes time gaps with the last recorded value per axis normalized to 1g. The average amount of missing values in a row was about 1.6 minutes. Because most ALS patients are immobile, this imputation method is unlikely to introduce much bias into the statistical analysis since subsequent observations are likely to be similar to previous ones. Following the imputation of missing observations, the van Hees raw non-wear algorithm (17) with optimized parameters was used (22) to remove non-wear times from the raw accelerometry data. Over 70% of the detected non-wear time was between 10PM and 10AM. Next, AIs were calculated. To calculate the AIs in R, the `computeActivityIndex` function from the PhysicalActivity R-package (24) modified to fit Equation was used. 61 different weight combinations found in “Appendix D” were used. This means AIs were calculated using 61 different AI metrics. Finally, to get a more accurate estimate of daily mean activity and make days more comparable, days with less than 8 hours of total wear time were removed from the data set (3).



Figure 2: A single measurement summarized with original AI metric (equal weights) into three different epoch lengths 10-sec, 60-sec, and 300-sec. The figure displays the additivity property of the AI-metric. The equation additivity of the metric can be found in “Appendix E”.

In total, the data was summarized into ten epoch lengths: 5-sec, 10-sec, 30-sec, 60-sec, 120-sec, 300-sec, 600-sec, 1800-sec, 3600-sec, and 7200-sec (Figure 2). This generated ten data sets, D1 to D10, used in the following statistical analysis and each contained 61 AI columns.

2.5 AI-based markers for physical activity

This research looked at markers that reflect a patient’s physical activity (26). The markers summarize the daily activity stored in the AI data sets into a single value. Three marker types were defined; (1) Mean AI, (2) Upper AI, and (3) 90P. The Mean AI was based on the average AI per day. Upper AI was defined as the average of the top 25% of the daily AI values. 90P was defined as the 90th percentile of the daily AI values. Due to right-skewed distributions, all three markers were estimated on log (ln) scale (Figure 3).

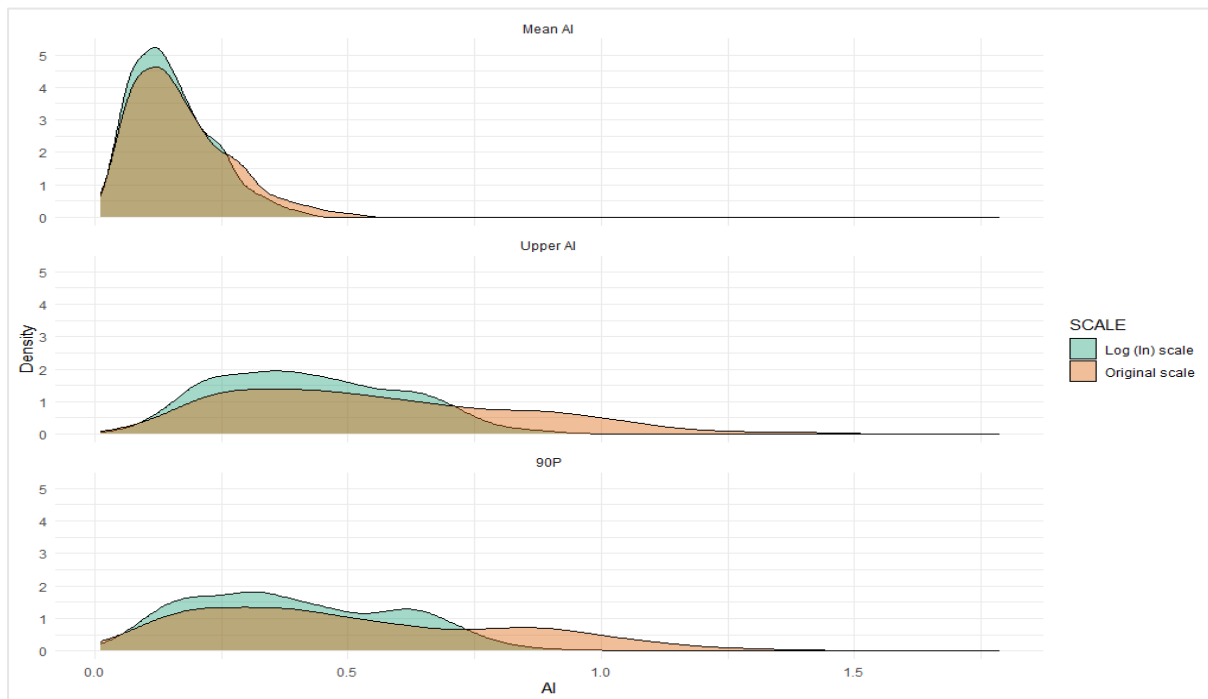


Figure 3: Densities of markers based on original AI with $w_1 = w_2 = w_3 = 1/3$ for 10-sec epoch length on original scale (orange) and on the log scale (blue).

Because there were 61 different AI metrics (61 different axis contribution combinations), 183 (3 x 61) different markers were calculated since every marker type had to be calculated for each AI metric. Figure 4 shows that the marker values increase with epoch length due to the additivity property of the metric (“Appendix E”).

“Appendix C” contains the R-Code for calculating the markers and for the statistical analysis of the markers described in sections 2.6 – 2.9.

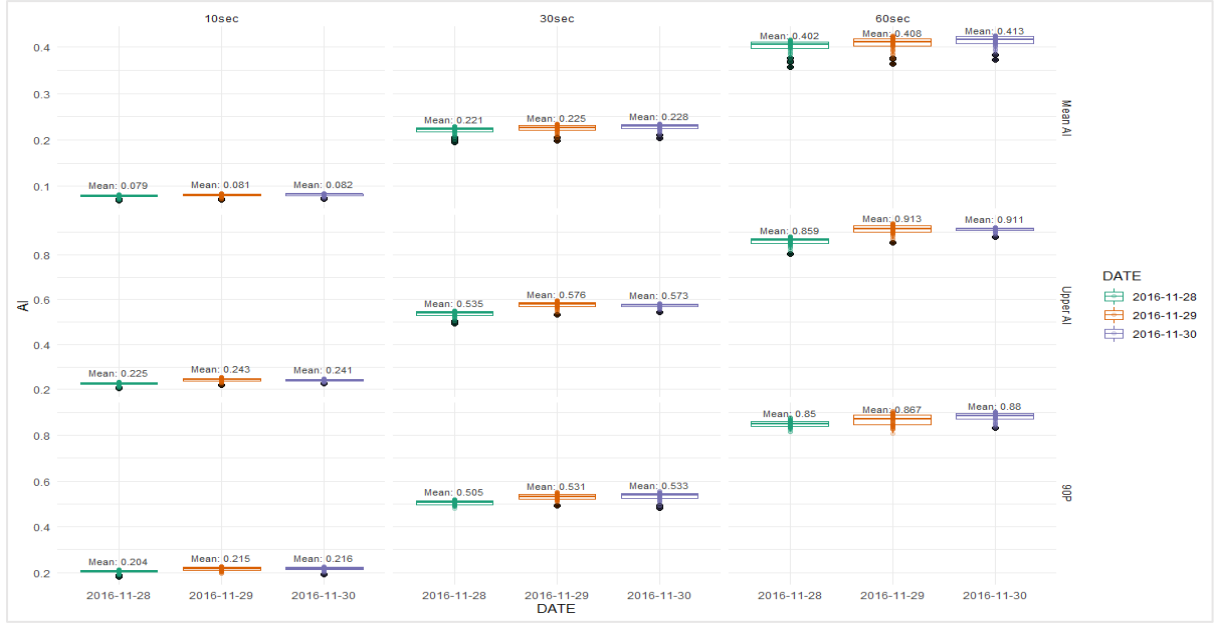


Figure 4: Boxplot of markers for ID2 measurement 2 and epoch lengths 10-sec, 30-sec and 60-sec. The boxplots indicate the spread between the 61 different markers based on the 61 AI metrics. The mean of the 61 markers is shown above the boxplots.

2.6 Longitudinal rates of decline in AI-based markers or disease progression

Mean population rates of decline were estimated using linear mixed-effects models. The models had a fixed effect for time and a random intercept and slope for time per individual.

$$ALSFRSR_{ij} = (\beta_0 + v_{0i}) + (\beta_1 + v_{1i}) \cdot time_{ij} + \varepsilon_{ij}$$

Equation 4: Example of mixed model for calculation of the longitudinal rate of decline in the ALSFRSR-R total score

In Equation 4 β_0 is the marker value at baseline and β_1 , the slope, expresses the mean population rate of decline. Additionally, the coefficient of variation (CoV) was calculated for each outcome, which was defined as the between-patient standard deviation of the slope σ_{v1} divided by the mean rate of change β_1 (3). A lower value indicates less variation among patients, allowing for more accurate detection of disease progression (3).

The longitudinal decline rates of this section and longitudinal correlations described in following sections 2.7 - 2.8 were calculated for all 183 markers in all ten data sets. Furthermore, all mentioned LME models were fitted using the lmer function from the lme4 package version 1.1-27.1 (27).

2.7 Longitudinal correlations between AI-based markers and clinical markers of disease severity

Standardized outcomes in the linear mixed-effects models were used to calculate correlation coefficients with a similar interpretation to Pearson r correlation. The LME models were fitted with a fixed effect for the standardized clinical marker and a random intercept and slope for the standardized clinical marker.

$$scale(meanAI_k)_{ij} = (\beta_0 + v_{0i}) + (\beta_1 + v_{1i}) \cdot scale(ALSFRSR)_{ij} + \varepsilon_{ij}$$

Equation 5: Example of mixed model for calculation of longitudinal correlation between a marker of type Mean AI and the ALSFRSR-R total score. The scale() function standardizes the markers.

In Equation 5 the Pearson r correlation is represented by the fixed slope β_1 .

2.8 Longitudinal correlations between AI-based markers and AC-based markers

Standardized outcomes in the linear mixed-effects models were used to calculate the longitudinal correlations between the AI-based markers and the AC-based marker %Active and daily AI by van Eijk et al. (3). The LME models were fitted with a fixed effect for the AC-based markers as well as a random intercept and slope for the AC-based markers.

2.9 Final selection of markers

A selection was made to give the reader a better overview of the 183 markers. First, the three markers based on the original AI metric (Equation 1) were chosen due to the rotational invariance property of the original metric. Furthermore, the marker with the highest mean for the correlation with the ALSFRS-R score across all ten data sets, i.e., epoch lengths, was chosen for each of the marker types, Mean AI, Upper AI, and 90P. Furthermore, the marker with the highest overall correlation was chosen. Similarly, the marker with the lowest mean coefficient of variation (CoV) across all ten data sets was chosen, as was the marker with the lowest CoV overall.

3 Results

3.1 Overview over final data sets

<i>Data set ID</i>	<i>Epoch length</i>	<i>Wear time days</i>	<i>Wear time adherence</i>	<i>Total monitoring period</i>	<i>Daily monitoring time</i>
<i>D1</i>	5-sec	694	93.0 %	9236 h	13.3 h/day
<i>D2</i>	10-sec	694	93.0 %	9236 h	13.3 h/day
<i>D3</i>	30-sec	693	92.9 %	9228 h	13.3 h/day
<i>D4</i>	60-sec	693	92.9 %	9227 h	13.3 h/day
<i>D5</i>	120-sec	693	92.9 %	9226 h	13.3 h/day
<i>D6</i>	300-sec	693	92.9 %	9222 h	13.3 h/day
<i>D7</i>	600-sec	693	92.9 %	9217 h	13.3 h/day
<i>D8</i>	1800-sec	691	92.6 %	9172 h	13.3 h/day
<i>D9</i>	3600-sec	690	92.5 %	9126 h	13.2 h/day
<i>D10</i>	7200-sec	686	92.0 %	9036 h	13.2 h/day

Table 1: Summary of wear time in modified AI data sets. "Wear time days" are defined as days with more than 8h of wear time. "Wear time adherence" is calculated by dividing wear time days by the 746 total days.

Table 1 summarizes the wear times of the ten data sets created for the ten different epoch lengths of the AI metrics. Because only complete epochs were included in the data sets, the number of wear time days and total monitoring time decreased slightly as epoch length increases. The total follow-up time was 503.3 to 503.2 months for D1 to D10. All ten data sets contained 206 measurements, and each patient produced an average of 4.9 measurements. The high similarity between the data sets allows for comparisons between them.

3.2 Longitudinal rates of decline in AI-based markers or disease progression

The slopes (β_1) of all 183 AI-based markers were negative. Additionally, all 95% CI were also negative and p-values <0.001. A longer epoch length resulted in a smaller slope because marker values increased with epoch length (Figure 4). Because they operate on a larger scale clinical markers of disease progression have smaller slopes than AI-based markers. Furthermore, epoch length had no effect on their slopes. "Appendix E" contains plots of the longitudinal rates of decline and baseline values for the AI-based markers and clinical markers of disease severity for all ten data sets.

Figure 5 depicts the coefficients of variation (CoVs) of the three marker types. The spread between each marker type's 61 AI-based markers was small across all data sets. The CoV for all three marker types was similar and increased with epoch length. When the CoV was averaged across all 61 markers, Mean AI markers had the minimum in D3 with 0.83 (SD:0.01), Upper AI markers in D2 with 0.87 (SD:0.02) and 90P markers in D2 with 0.84 (SD:0.02). In all data sets, the Mean AI markers remained below the CoV of the clinical markers, which could positively affect sample size calculations and increase the sensitivity to detect differential disease progression (3). Upper AI stayed below the CoV of the clinical markers, with a few outliers after D6. CoVs were highest for 90P markers.

AI-based markers had higher between-patient variability over time on average than AC-based markers (3) (coefficient of variation 0.80 – 1.18 vs. 0.64–0.81, respectively).

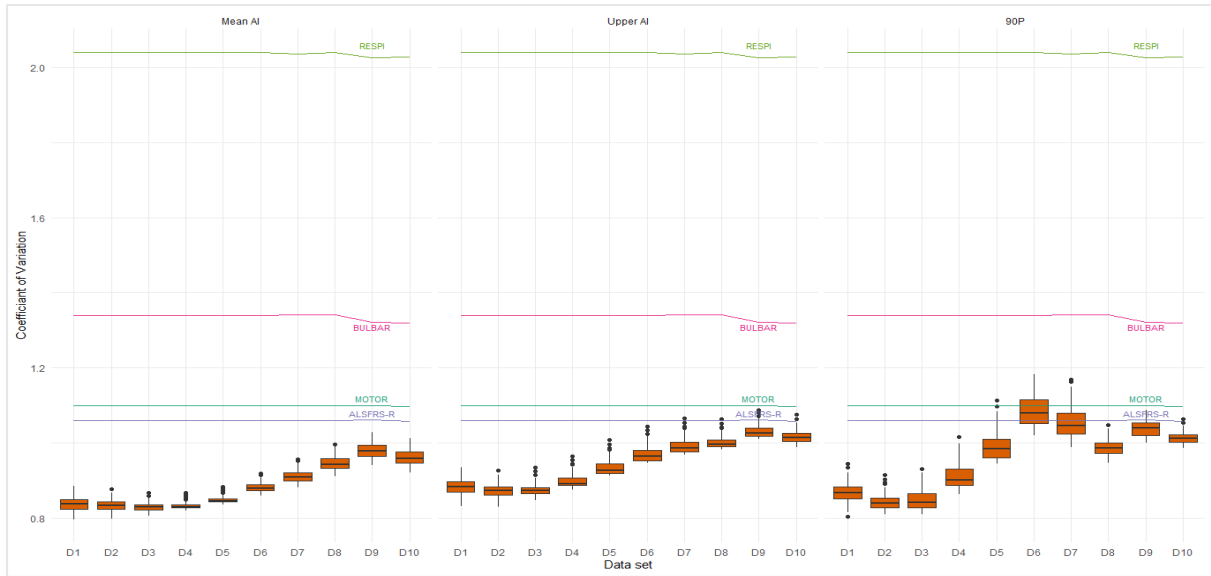


Figure 5: Boxplot of CoV for the three main marker types for D1-D10 (orange). The boxplots show the spread of the 61 markers based on 61 different AI-metrics. The clinical markers of disease severity, ALSFRS-R (blue), MOTOR (green), BULBAR (pink) and RESPI (green) are also shown for comparison.

3.3 Correlations between AI-based markers and clinical markers of disease severity

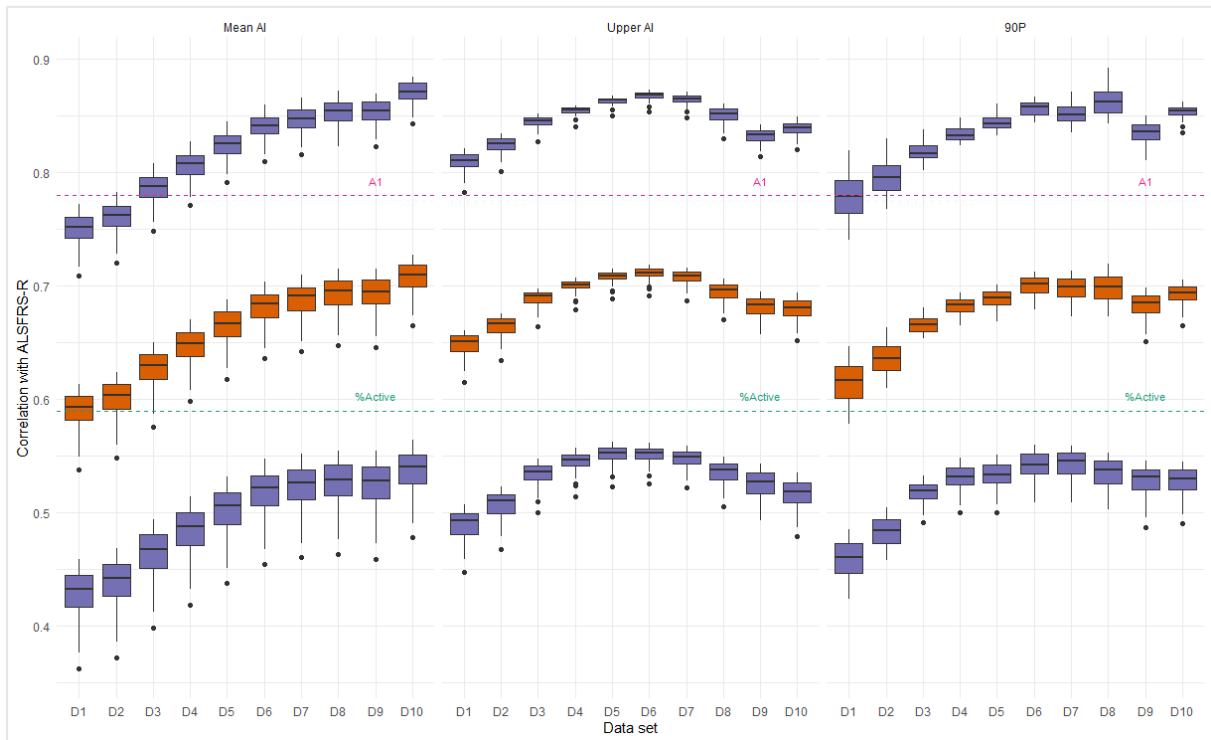


Figure 6: Boxplot of correlations between the three marker types and the clinical marker ALSFRS-R for data sets D1-D10 (orange). Boxplots of the upper and lower bounds of the correlations' 95% CI are shown in blue. The longitudinal correlation of the %Active and AI markers by van Eijk et al. (3) are shown in green and pink, respectively. The boxplots show the spread of the 61 markers based on 61 different AI metrics.

Figure 6 shows the correlations between the three marker types and the ALSFRS-R total score across all data sets. The spread between the 61 markers based on the different modified AI metrics was small for all marker types and data sets. Markers of all three marker types had moderate to strong associations (28) with the ALSFRS-R score. For Mean AI, the correlation averaged across all 61 markers

found its maximum in D10 with 0.71 (SD:0.015). 90P had the maximum in D6 with 0.70 (SD:0.01), and Upper AI in D6 with 0.71 (SD:0.01). For all markers, the p-values for the correlations were <0.001 . The 183 AI-markers had a weaker correlation with the ALSFRS-R than the A1 marker, but most markers had a higher correlation than the % Active marker.

3.4 Longitudinal correlation between AI activity markers and AC-based markers

The longitudinal correlations of the three marker types with the A1 marker (3) ranged from 0.80 (95% CI: 0.72 – 0.88) to 0.95 (95% CI: 0.91 – 1.00) with all p-values being <0.001 . There were no discernible differences between marker types in terms of correlation with the AC-based A1 marker. “Appendix E” contains a graph depicting the correlations of the AI-based markers with A1.

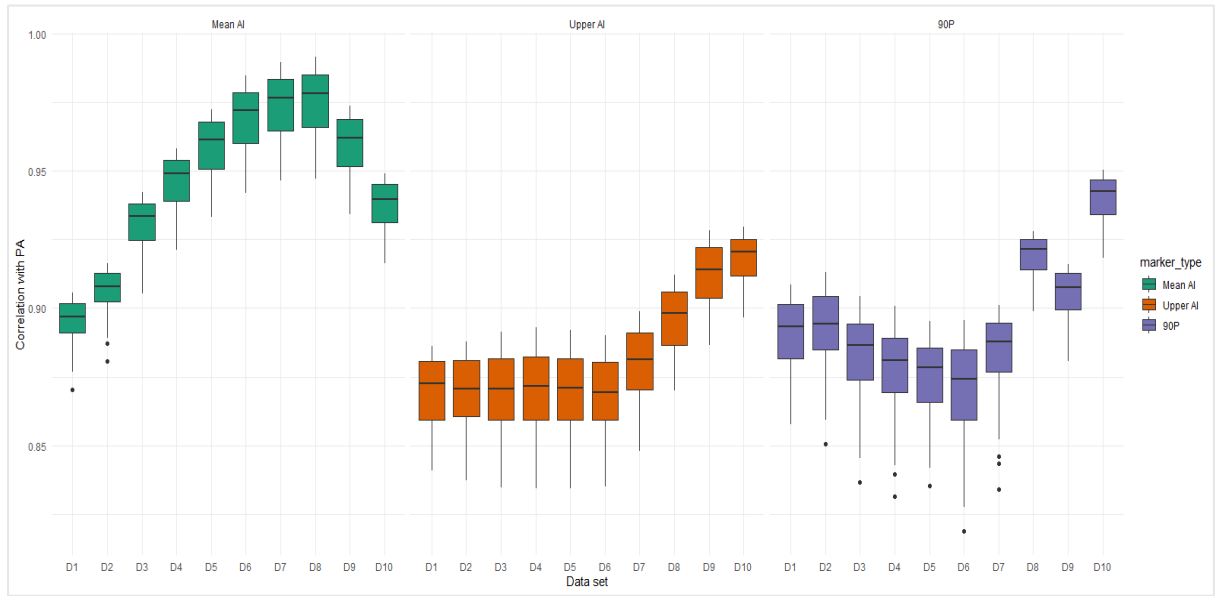


Figure 7: Longitudinal correlations of the three main marker types with %Active marker for D1-D10. The boxplots show the spread of the 61 markers based on 61 different AI-metrics

The correlations of the marker types with the %Active marker (3) are shown in Figure 7. Correlations ranged from 0.82 (95% CI: 0.67 – 0.97) to 0.99 (95% CI: 0.88 – 1.10).

Therefore, all AI-based markers correlated strongly with both AC-based markers. The correlation between AC-based markers A1 and %Active is slightly smaller with 0.78 (95% CI: 0.70 – 0.86). The Mean AI markers had the strongest association with the %Active marker. All markers had stronger correlations with %Active in longer epoch lengths. Correlations with the A1 marker, on the other hand, remained stable across all three marker types and data sets D1 to D10.

3.5 Final markers

Table 2 provides the Mean AI markers resulting from the final selection process together with the reason for their selection.

Marker type	ID	Reason for selection	Weights of modified AI
Mean AI	M1	Original AI, Rotational invariance	$w_1 = w_2 = w_3 = 1/3$
	M2	Highest mean correlation over all ten data sets	$w_1 = 0.7 ; w_2 = 0.3 ; w_3 = 0$
	M3	Highest correlation overall	$w_1 = w_2 = 0.5 ; w_3 = 0$
	M4	Lowest mean CoV over all ten data sets	$w_1 = 0.2 ; w_2 = 0.8 ; w_3 = 0$
	M5	Lowest CoV overall	$w_1 = w_2 = 0 ; w_3 = 1$

Table 2: Final selection of mean AI markers. ID column contains the IDs of the individual markers. In the Weights of modified AI column, w_1 is the weight for the x-axis, w_2 for the y-axis and w_3 for the z-axis.

Because the final markers of the three different marker types produced similar results, only the Mean AI markers will be discussed in depth. "Appendix E" contains detailed summaries of the Upper AI and 90P markers M6-M14.

<i>ID</i>	<i>Selection method</i>	<i>Data set ID</i>	<i>Epoch length</i>	<i>Correlation ALSFRS-R</i>	<i>95% CI ALSFRS-R</i>	<i>Correlation MOTOR</i>	<i>95% CI MOTOR</i>	<i>CoV</i>
<i>M1</i>	MaxCor	D10	7200	0.71	0.54 – 0.87	0.77	0.59 – 0.94	0.96
	MinCoV	D3	30	0.63	0.47 – 0.79	0.72	0.55 – 0.89	0.82
	Mean	D1-D10	5-7200	0.66	0.50 – 0.82	0.74	0.57 – 0.91	0.88
<i>M2</i>	MaxCor	D10	7200	0.73	0.56 – 0.88	0.79	0.62 – 0.96	0.96
	MinCoV	D4	60	0.67	0.52 – 0.82	0.76	0.60 – 0.92	0.84
	Mean	D1-D10	5-7200	0.68	0.53 – 0.83	0.76	0.60 – 0.92	0.89
<i>M3</i>	MaxCor	D10	7200	0.73	0.56 – 0.88	0.79	0.62 – 0.96	0.94
	MinCoV	D4	60	0.67	0.51 – 0.82	0.76	0.60 – 0.92	0.83
	Mean	D1-D10	5-7200	0.68	0.53 – 0.83	0.76	0.59 – 0.92	0.88
<i>M4</i>	MaxCor	D10	7200	0.73	0.56 – 0.88	0.79	0.62 – 0.96	0.92
	MinCoV	D4	60	0.66	0.50 – 0.81	0.75	0.59 – 0.92	0.83
	Mean	D1-D10	5-7200	0.67	0.52 – 0.83	0.76	0.59 – 0.92	0.87
<i>M5</i>	MaxCor	D10	7200	0.66	0.48 – 0.84	0.72	0.53 – 0.90	0.98
	MinCoV	D1	5	0.54	0.36 – 0.71	0.62	0.44 – 0.80	0.80
	Mean	D1-D10	5-7200	0.61	0.43 – 0.79	0.69	0.50 – 0.87	0.88

Table 3: Longitudinal correlations between disease progression and markers M1 -M5 (Mean AI). CoV of the markers is also included in the table for comparison. MaxCor in the "Selection method" column means that the row contains the values of the data set, in which the marker had the highest correlation with the ALSFRS-R total score. MinCoV means that the row contains the values of the data set in which the marker had the lowest CoV. Mean means that this row contains the mean values over all ten data sets. Column "Epoch length" reminds the reader which data set contains which epoch length.

Table 3 shows the longitudinal correlations between Mean AI markers M1-M5 and the ALSFRS-R total score and the motor domain (i.e., ALSFRS-R item 4–9), as well as the CoVs. All five markers had the highest correlations with ALSFRS-R in D10 (0.66 – 0.73). Correlations with the motor domain were stronger, with values ranging from 0.72 to 0.79. The CoVs were lower in Mean AI markers than ALSFRS-R across all data sets ; range 0.80-0.98 vs. 1.06. However, Mean AI markers showed more variability over time than AC-based markers; range 0.80-0.98 vs. 0.64–0.81. The minimum CoVs for the five Mean AI markers were found in D1, D3, and D4 so in data sets containing shorter epoch lengths.

All of the final markers showed moderate to strong correlations (28) with the clinical markers for disease severity. The motor domain was the primary driver of the correlations between the AI-markers and the ALSFRS-R for all three marker types. The p-values for all correlations with the ALSFRS-R and MOTOR scores were less than 0.001. Most markers correlated similarly with clinical markers of disease severity and had similar CoVs. Only markers M5, M9, and M14 with $w_3 = 1$ deviated notably and had a lower correlation with ALSFRS-R. "Appendix D" contains an extended list of final marker correlations with ALSFRS-R, MOTOR, BULBAR, and RESPI scores, as well as intercepts, slopes, and CoVs. Furthermore, "Appendix E" includes a visualization of the correlations of markers M1-M14 with AC-based markers A1 and %Active.

4 Discussion and Conclusion

This thesis project assessed to what extent markers based on the AI metric can be used to quantify disease progression in patients with amyotrophic lateral sclerosis and can therefore be used as an alternative to AC-based markers. The research discovered that AI-based markers were moderately to strongly associated with disease progression. Furthermore, most markers had less between-patient variability over time than clinical markers.

Using an open-source summary metric like AI instead of AC metrics to summarize raw accelerometer data into epochs makes results more comparable across studies (12) and easier to reproduce. It also provides a more transparent way to work with raw accelerometer data (12) and allows researchers to fine-tune the metric to better suit their research objectives. Furthermore, the markers based on this metric unaffected by device manufacturers and the Activity Count metrics they provide. This gives researchers more freedom when selecting which accelerometers to use in their studies.

The three marker types, Mean AI, Upper AI, and 90P, performed equally well in terms of their correlation with disease progression. The AI-based markers correlated less than the AC-based marker A1. They did, however, show stronger correlations than %Active. It is important to note that most of the proposed AI-based markers' 95% confidence intervals exceeded 0.80. As a result, AI-based markers may perform similarly to A1 in other studies. Moreover, AI-based markers may capture other signs of disease progression that the ALSFRS-R score does not, which could explain lower correlations.

The markers from the final selection process showed that markers based on AI metrics that put large weights on the accelerometer's x and y axes had higher correlations with the ALSFRS-R score than markers with a large weight on the z-axis. Therefore, it seems that movement along the z-axis is unimportant for quantifying disease progression in patients with ALS. Additionally, markers based on AI metrics with large weights on the y-axis had the least between-patient variability across all data sets which may indicate that vertical movement is especially important for quantifying disease progression. It should be noted that only the original AI metric, that is, the AI metric that assigns equal weights to all axes, is rotationally invariant. AI metrics with different axes weights should be avoided if researchers want this property to be present. Because the differences between markers in terms of correlation with the ALSFRS-R score and between-patient variability over time were not very large, using the rotationally invariant original AI metric may be more robust in free-living conditions (12).

Interestingly, all markers had stronger correlations with clinical markers of disease severity in longer epoch lengths compared to shorter epoch lengths. This means that examining the variances of the raw accelerometer data over longer epoch lengths may reveal more about ALS patients' disease severity. This property may benefit larger studies with many participants, as raw data can be compressed into much smaller data sets.

However, higher correlations came at the expense of increasing between-patient variability over time. Even though most AI-based markers had lower between-patient variability over time than the ALSFRS-R total score, the differences decreased with increasing epoch lengths and sometimes exceeded the CoV of the ALSFRS-R score. This is especially true for markers of type 90P. Overall, markers of type Mean AI had the least between-patient variability in the rate of decline. Compared to the AC-based markers (3) AI-based markers had a higher between-patient variability over time implying that they are less sensitive to detect differential disease progression (3). This may be because the marker types presented in this research are markers for a patient's physical activity, which can vary greatly between patients (3).

The marker types presented in this research correlated strongly with AC-based markers %Active, which reflects average daily activity, and A1, which reflects a patient's physical capability. This suggests that markers based on AI metrics could be viable alternatives to AC-based markers. Surprisingly, AI-based markers correlated better with the AC-based markers than the two AC-based markers with each other. This could mean that the AI markers for physical activity presented in this research might also include information about the physical capability of the patient.

Several limitations of this research and the AI metrics need to be addressed. First of all, raw accelerometer data must be processed to use the AI metric. Raw data files contain millions of observations, so processing can be time-consuming and may necessitate the use of good hardware depending on the size of the study. However, with ongoing technological advancements, this should not be an issue in the future. Second, the raw accelerometer data had many missing observations even during the day, that had to be imputed. This might have introduced some bias into the analysis. Because of the missingness, there were fewer detectable non-wear times in the unprocessed raw data, which meant that the systematic noise could only be calculated for eight of the ten accelerometer serial numbers used in the study. Furthermore, non-wear times for calculating the systematic noise had to be detected by a non-wear algorithm which could falsely classify wear time as non-wear time and resulting in an incorrect estimation of the systematic noise. Therefore, testing AI-based markers on data with clearly labeled wear and non-wear times, such as diaries kept by patients or caretakers, and less missingness, may be important. For future research, the accelerometer should ideally be placed on a desk before being used to obtain non-wear times of all serial numbers to calculate systematic noise (12). Finally, the ten data sets did not have the same number of wear time days because as much data as possible was kept in each data set. The differences were caused by different epoch lengths, which resulted in different monitoring times for each day, allowing for minor variations in wear time days. This may affect correlations and CoVs. However, because the differences between the data sets were so small, no significant deviations could be found when looking into this issue.

In conclusion, this thesis project demonstrated that activity markers based on AI metrics can be used to track disease progression and that they have the potential to be an alternative to markers based on Activity Counts. However, compared to some AC-based markers, the correlation of AI-based markers with disease progression was slightly lower, and the between-patient variability over time was higher. As a result, more research with other data sets is required to fully determine the feasibility of AI-based markers in practice.

References

1. Mayo Clinic: Amyotrophic lateral sclerosis (ALS) - Symptoms and causes. [Online].; 2022 [cited 2022 1 June]. Available from: <https://www.mayoclinic.org/diseases-conditions/amyotrophic-lateral-sclerosis/symptoms-causes/syc-20354022>.
2. van Eijk R, Beelen A, Kruitwagen ET, Murray D, Radakovic R, Hobson E, et al. A Road Map for Remote Digital Health Technology for Motor Neuron Disease. *Journal of medical Internet Research*. 2021; 23(9): e28766.
3. van Eijk R, Bakers J, Bunte T, de Fockert A, Eijkemans M, van den Berg L. Accelerometry for remote monitoring of physical activity in amyotrophic lateral sclerosis: a longitudinal cohort study. *Journal of Neurology*. 2019; 266(10): 2387-2395.
4. van Eijk R, Westeneng H, Nikolakopoulos S, Verhagen I, van Es M, Eijkemans M. Refining eligibility criteria for amyotrophic lateral sclerosis clinical trials. *Neurology*. 2019; 92(5): 451-460.
5. Arvidsson D, Fridolfsson J, Börjesson M. Measurement of physical activity in clinical practice using accelerometers. *J Intern Med*. 2019 August; 286(2): 127-153.
6. de Almeida Mendes M, da Silva ICM, Ramires VV, Reichert FF, Martins RC, Tomasi E. Calibration of raw accelerometer data to measure physical activity: A systematic review. *Gait & Posture*. 2018 March; 61: 98-110.
7. Di J, Demanuele C, Kettermann A, Karahanoglu FI, Cappelleri JC, Potter A, et al. Considerations to address missing data when deriving clinical trial endpoints from digital health technologies. *Contemporary Clinical Trials*. 2022; 113: 106661.
8. Feito Y, Hornbuckle LM, Reid LA, Crouter SE. Effect of ActiGraph's low frequency extension for estimating steps and physical activity intensity. *PLOS ONE*. 2017 November; 12(11): e0188242.
9. Crouter SE, Kuffel E, Haas JD, Frongillo EA, Basset DR. Refined two-regression model for the ActiGraph accelerometer. *Med Sci Sports Exerc*. 2010; 42: 1029-1037.
10. ActiGraph. What are counts? [Online].; 2022 [cited 2022 June 1]. Available from: <https://actigraphcorp.my.site.com/support/s/article/What-are-counts>.
11. ActiGraph. Raw and Summary (Epoch) Data Explained. [Online].; 2018 [cited 2022 June 22]. Available from: <https://actigraphcorp.my.site.com/support/s/article/Raw-and-Summary-Epoch-Data-Explained>.
12. Bai J, Xiao L, Evenson K, LaCroix A, Crainiceanu C, et al. An Activity Index for Raw Accelerometry Data and Its Comparison with Other Activity Metrics. *PLOS ONE*. 2016; 11(8): e0160644.
13. John D, Tang Q, Albinali F, Intille S. An Open-Source Monitor-Independent Movement Summary for Accelerometer Data Processing. *Journal for the Measurement of Physical Behaviour*. 2019; 2(4): 268-281.

14. Maczák B, Vadai G, Dér A, Szendi I, Gingl Z. Detailed analysis and comparison of different activity metrics. PLOS ONE. 2021; 16(12): e0261718.
15. Brønd JC, Arvidsson D. Sampling frequency affects the processing of Actigraph raw acceleration data to activity counts. J Appl Physiol (1985). 2016 February; 120(3): 362-369.
16. Bai J, He B, Shou H, Zipunnikov V, Glass T, Crainiceanu C. Normalization and extraction of interpretable metrics from raw accelerometry data. Biostatistics. 2013; 15(1): 102-116.
17. van Hees V, Gorzelniak L, Dean León E, Eder M, Pias M, Taherian S, et al. Separating Movement and Gravity Components in an Acceleration Signal and Implications for the Assessment of Human Daily Physical Activity. PLoS ONE. 2013; 8(4): e61691.10.
18. Cedarbaum JM, Stambler N, Malta E, Fuller C, Hilt D, Thurmond B, et al. The ALSFRS-R: a revised ALS functional rating scale that incorporates assessments of respiratory function. BDNF ALS Study Group (Phase III). Journal of the Neurological Sciences. 1999 October; 169(1-2): 13-21.
19. SURF.nl. Send large files securely and encrypted with SURFfilesender. [Online].; 2022 [cited 2022 June 2]. Available from: <https://www.surf.nl/en/surffilesender-send-large-files-securely-and-encrypted>.
20. GDPR.EU. [Online]. [cited 2022 June 7]. Available from: <https://gdpr.eu/>.
21. Tackney MS, Cook DG, Stahl D, Ismail K, Williamson E, Carpenter J. A framework for handling missing accelerometer outcome data in trials. Trials. 2021 June.
22. Syed S, Morseth B, Hopstock L, Horsch A. Evaluating the performance of raw and epoch non-wear algorithms using multiple accelerometers and electrocardiogram recordings. Scientific Reports. 2020; 10(1).
23. Syed S. CNN-Non-Wear_Time-Algorithm. [Online].; 2020 [cited 2022 May 15]. Available from: <https://github.com/shaheen-syed/CNN-Non-Wear-Time-Algorithm>.
24. Bai , Bai J. Activity Index Calculation using Raw 'Accelerometry' Data [R package ActivityIndex version 0.3.7]. [Online].; 2022 [cited 2022 June 2]. Available from: <https://cran.r-project.org/web/packages/ActivityIndex/index.html>.
25. van Hees V, Fang Z, Zhao J, Heywood J, Mirkes E, Sabia S, et al. GGIR: Raw Accelerometer Data Analysis. [Online].; 2022. Available from: <https://CRAN.R-project.org/package=GGIR>.
26. Caspersen CJ, Powell KE, Christenson GM. Physical activity, exercise, and physical fitness: definitions and distinctions for health-related research. Public Health Rep. 1985; 100(2): 126-131.
27. Bates D, Maechler M, Bolker B, Walker S. Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software. 2015; 67(1): 1-48.
28. Akoglu H. User's guide to correlation coefficients. Turk J Emerg Med. 2018 September; 18(3): 91-93.

Appendix A: R Packages

R Packages

Package name	Version	Reference
PhysicalActivity	0.2-4	Leena Choi, Cole Beck, Zhouwen Liu, Ryan Moore, Charles E. Matthews and Maciej S. Buchowski (2021). PhysicalActivity: Process Accelerometer Data for Physical Activity Measurement. R package version 0.2-4. https://CRAN.R-project.org/package=PhysicalActivity
RSQLite	2.2.13	Kirill Müller, Hadley Wickham, David A. James and Seth Falcon (2022). RSQLite: SQLite Interface for R. R package version 2.2.13. https://CRAN.R-project.org/package=RSQLite
ggplot2	3.3.5	H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.
tidyverse	1.3.1	Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686
read.gt3x	1.2.0	Tuomo Nieminen and John Muschelli (2022). read.gt3x: Parse 'ActiGraph' 'GT3X'/'GT3X+' 'Accelerometer' Data. R package version 1.2.0.
lubridate	1.8.0	Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL https://www.jstatsoft.org/v40/i03/ .
ActivityIndex	0.3.7	Bai J, Di C, Xiao L, Evenson KR, LaCroix AZ, Crainiceanu CM, Buchner DM (2016). “An activity index for raw accelerometry data and its comparison with other activity metrics.” <i>_PloS one_</i> , *11*(8), e0160644.
gtools	3.9.2	Gregory R. Warnes, Ben Bolker and Thomas Lumley (2021). gtools: Various R Programming Tools. R package version 3.9.2. https://CRAN.R-project.org/package=gtools
data.table	1.14.2	Matt Dowle and Arun Srinivasan (2021). data.table: Extension of `data.frame`. R package version 1.14.2. https://CRAN.R-project.org/package=data.table
GGIR	2.6-4	van Hees V, Fang Z, Zhao J, Heywood J, Mirkes E, Sabia S, Miguelles J(2022). <i>_GGIR: Raw Accelerometer Data Analysis_</i> . doi: 10.5281/zenodo.1051064 (URL: https://doi.org/10.5281/zenodo.1051064), R package version 2.6-4, <URL: https://CRAN.R-project.org/package=GGIR >.
matrixStats	0.61.0	Henrik Bengtsson (2021). matrixStats: Functions that Apply to Rows and Columns of Matrices (and to Vectors). R package version 0.61.0. https://CRAN.R-project.org/package=matrixStats
lme4	1.1-27.1	Douglas Bates, Martin Maechler, Ben Bolker, Steve Walker (2015). Fitting Linear Mixed-Effects Models Using lme4. Journal of Statistical Software, 67(1), 1-48. doi:10.18637/jss.v067.i01.
reshape2	1.4.4	Hadley Wickham (2007). Reshaping Data with the reshape Package. Journal of Statistical Software, 21(12), 1-20. URL http://www.jstatsoft.org/v21/i12/ .

Appendix B: AC-Data R-Code

Activity Count Data pre-processing and analysis

```
library(PhysicalActivity)
library(RSQLite)
library(ggplot2)
library(readr)
library(corr)
library(tidyverse)
library(nlme)
library(lubridate)
library(lme4)

B <- openxlsx::read.xlsx ("C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/ActigraphyProjectALS.xlsx", "BASELINE", detectDates = T)
B$SID <- paste ("ID", as.numeric (gsub (x = B$SID, "031-", replacement =
"")), sep = "")

#####
# Weartime data
#####

# Load in wear time data
W <- openxlsx::read.xlsx ( "C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/ActigraphyProjectALS.xlsx", sheet = "WEARTIME") # change to per-
sonal file location
W <- (W[, c ("ID", "START", "STOP", "VISIT", "tSTART", "tSTOP", "TO-
TAL", "BULBAR", "FINE", "GROSS", "RESPI", "MOTOR")])
W$START <- openxlsx::convertToDate(W$START)
W$STOP <- openxlsx::convertToDate(W$STOP)
W$tSTART <- format(as.POSIXct((W$tSTART) * 86400, origin = "1970-01-01", tz
= "UTC"), "%H:%M:%S")
W$tSTOP <- format(as.POSIXct((W$tSTOP) * 86400, origin = "1970-01-01", tz =
"UTC"), "%H:%M:%S")

#####
#####
## 2. Data generator
#####
#####

# # # # # # #
# SettingS #

## Low Frequency Extension ;
## Waist-worn ;
## 10-sec epoch ;

FILE <- dir ("C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/count_data") # change to per-personal AGD file location

rexp <- "^(\\w+)\\s?(.*)$"
IDs <- sub (rexp, "\\1", FILE)
VISITs <- gsub (pattern = "10sec.agd" , x = sub (rexp, "\\2", FILE), re-
placement = "")
VISITs <- ifelse (VISITs == "Baseline", "Month 0", VISITs)
VISITs <- as.numeric (gsub (pattern = "Month ", "", VISITs))
```



```

# Not in dataset:
W[!paste (W$ID, W$VISIT, sep = ":") %in% paste (IDs, VISITs, sep = ":"), ]
W <- W[paste (W$ID, W$VISIT, sep = ":") %in% paste (IDs, VISITs, sep =
":"), ]

# # Step 0: datafile [per ID for each visit]:
C <- apply (1:length (B$SID), function (i){

  # List per ID
  x <- list (dim (0))
  names (x) <- paste (B$SID[i])

  # List visit
  x[[B$SID[i]]] <- lapply (IDs[IDs == B$SID[i]], function (x){dim (0)})
  names (x[[B$SID[i]]]) <- paste ("M", VISITs[IDs == B$SID[i]], sep = "")
  x
})

for (i in 1:length (FILE)){

  # # Step 1: load AGD file in R
  D <- PhysicalActivity::readActigraph (paste ( "C:/Us-
ers/enied/OneDrive/Dokumente/ThesisProject_ALS/count_data/", FILE[i], sep =
""), convert-Time = TRUE) # change to personal AGD file location
  D$DATE <- as.Date (D$TimeStamp)

  # Make ID & VISIT
  ID <- sub (rexp,"\\1", FILE[i])
  VISIT <- gsub (pattern = "10sec.agd" , x = sub (rexp,"\\2", FILE[i]), re-
placement = "")
  VISIT <- ifelse (VISIT == "Baseline", "Month 0", VISIT)
  VISIT <- as.numeric (gsub (pattern = "Month ", "", VISIT))
  SELECT <- W[W$ID == ID & W$VISIT == VISIT, ]
  ALSFRS <- W[W$ID == ID & W$VISIT == VISIT, ]$TOTAL
  BULBAR <- W[W$ID == ID & W$VISIT == VISIT, ]$BULBAR
  FINE <- W[W$ID == ID & W$VISIT == VISIT, ]$FINE
  GROSS <- W[W$ID == ID & W$VISIT == VISIT, ]$GROSS
  RESPI <- W[W$ID == ID & W$VISIT == VISIT, ]$RESPI
  MOTOR <- W[W$ID == ID & W$VISIT == VISIT, ]$MOTOR

  # # Check 1: missing dates?
  if (is.na (SELECT$START) | is.na (SELECT$START)){

    print (paste ("Missing dates", ID, "- Month", VISIT))

    # # # Save NA in predefined list
    POS <- which (names (C) == ID)
    POSII <- which (names (C[[POS ]]) == paste ("M", VISIT, sep = ""))
    C[[POS]][[POSII]] <- NA

  } else {

    # # Check 2: file in W?
    if (nrow (SELECT) != 1){
      print ("Dates are not correct for:", ID, VISIT)
      break
    } else {

```

```

    if ((SELECT$START >= min (D$DATE)) & (SELECT$STOP <= max (D$DATE))) {
      print (paste ("Dates check OK for", ID, "- Month", VISIT))
    } else {
      print (paste ("Actigraph dates differ from W:", ID, VISIT))
      break
    }
  }
}

# # Step 2: Select datarows and collapse:
D <- D[((D$DATE >= SELECT$START) & (D$DATE <= SELECT$STOP)), ]
D$DATE <- D$TimeStamp
D$sDATE <- as.Date (D$TimeStamp)
D$TIME <- strftime (D$TimeStamp, format="%H:%M:%S", tz = "UTC")
D$SELECT <- ifelse ((D$sDATE != SELECT$START) & (D$sDATE != SE-
LECT$STOP), T,
                    ifelse ((D$sDATE == SELECT$START) & (D$TIME >= SE-
LECT$START), T,
                            ifelse ((D$sDATE == SELECT$STOP) & (D$TIME
<= SELECT$STOP), T, F)))
D <- D[D$SELECT, ]

# # Step 3: Data cleaning and reformatting:
D$DAY <- lubridate::wday (D$TimeStamp, label = T)
D <- D[, !colnames (D) %in% c ("TimeStamp")]
D$ID <- ID
D$VISIT <- VISIT
D$ALSFRS<-ALSFRS
D$FINE<-FINE
D$GROSS<-GROSS
D$BULBAR<-BULBAR
D$RESPI<-RESPI
D$MOTOR <- MOTOR
names (D)[names (D) %in% c ("axis1", "axis2", "axis3",
                           "steps", "inclineOff", "inclineStanding",
                           "inclineSitting", "inclineLying", "vm")] <-
  c ("A1", "A2", "A3", "STEPS", "OFF", "iSTAND", "iSIT", "iLIE", "VM")

D <- PhysicalActivity::wearingMarking (D, frame = 150, perMinuteCts =
6,
                                     TS = "DATE", cts = "VM",
                                     streamFrame = NULL, allowance-
Frame = 2, newcolname = "WEAR",
                                     getMinuteMarking = FALSE, day-
Start = "00:00:00", tz = "UTC") # Non-wear time alogritth : 150 min [Choi
et al 2010]
D <- D[, !colnames (D) %in% c ("weekday", "SELECT")]
names (D)[names (D) == "days"] <- "OBS"
D$WEAR <- as.numeric (D$WEAR == "w")

# subset dataframe into number of active epochs per day, transform to
active hours, and remove days with less than 8h of weartime
D <- merge (D, data.frame (QUOTE = (unlist (by (D[D$WEAR == 1, ],
D[D$WEAR == 1,]$OBS, nrow, simplify = F)) / (60*6)) >= 8,
OBS = names (unlist (by (D[D$WEAR == 1, ],
D[D$WEAR == 1,]$OBS, nrow, simplify = F))))), by = "OBS")
D <- D[D$QUOTE, ]
D <- D[, !colnames (D) %in% c ("HR")]
D <- D[order (D$DATE), ]

# # # Save data in predefined list
POS <- which (names (C) == ID)

```

```

    POSII <- which (names (C[[POS ]]) == paste ("M", VISIT, sep = ""))
    C[[POS]][[POSII]] <- D

  }

  cat (i, "\r")

}

#####
#####
## Generating analysis set
#####
#####

IDS <- names (sapply (C, function (d){names (d)}))
rep (IDS, sapply (sapply (C, function (d){names (d)}), function (d){length
(d)}))
SUM <- data.frame (ID = rep (IDS, sapply (sapply (C, function (d){names
(d)}), function (d){length (d)})),
                  M = do.call ("c", sapply (C, function (d){names (d)})))
SUM$VISIT <- gsub (SUM$M, pattern = "M", replacement = "")

DATES <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]])) == paste ("M", SUM[i, ]$VISIT, sep =
""))

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # Binary physical activity:
  as.numeric (unlist (by (as.Date (D1[D1$WEAR == 1, ]$DATE),
                           D1[D1$WEAR == 1, ]$OBS,
                           min, simplify = F)))
})
DATES <- as.Date.numeric (unlist (DATES), origin = "1970-01-01")

ID <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # Binary physical activity:
  as.character (unlist (by (D1[D1$WEAR == 1, ]$ID,
                           D1[D1$WEAR == 1, ]$OBS,
                           unique, simplify = F)))
})
ID <- unlist (ID)

VISIT <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:

```

```

POS <- which (names (C) == SUM[i, ]$ID)
POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

# Data:
D1 <- C[[POS]][[POSII]]

# # Binary physical activity:
as.numeric (unlist (by (D1[D1$WEAR == 1, ]$VISIT,
                        D1[D1$WEAR == 1, ]$OBS,
                        unique, simplify = F)))
})
VISIT <- unlist (VISIT)

ALSFRS <- lapply (1:nrow (SUM), function (i){

# Coordinate data:
POS <- which (names (C) == SUM[i, ]$ID)
POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

# Data:
D1 <- C[[POS]][[POSII]]

# # Binary physical activity:
as.numeric (unlist (by (D1$ALSFRS,
                        D1$OBS,
                        min, simplify = F)))
})
ALSFRS<-unlist(ALSFRS)

BULBAR <- lapply (1:nrow (SUM), function (i){

# Coordinate data:
POS <- which (names (C) == SUM[i, ]$ID)
POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

# Data:
D1 <- C[[POS]][[POSII]]

# # Binary physical activity:
as.numeric (unlist (by (D1$BULBAR,
                        D1$OBS,
                        min, simplify = F)))
})
BULBAR<-unlist(BULBAR)

FINE <- lapply (1:nrow (SUM), function (i){

# Coordinate data:
POS <- which (names (C) == SUM[i, ]$ID)
POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

# Data:
D1 <- C[[POS]][[POSII]]

# # Binary physical activity:
as.numeric (unlist (by (D1$FINE,
                        D1$OBS,

```

```

min, simplify = F)))
})
FINE<-unlist(FINE)

GROSS <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]])) == paste ("M", SUM[i, ]$VISIT, sep =
  "")

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # Binary physical activity:
  as.numeric (unlist (by (D1$GROSS,
                          D1$OBS,
                          min, simplify = F)))
})

GROSS<-unlist(GROSS)

RESPI <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]])) == paste ("M", SUM[i, ]$VISIT, sep =
  "")

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # Binary physical activity:
  as.numeric (unlist (by (D1$RESPI,
                          D1$OBS,
                          min, simplify = F)))
})

RESPI<-unlist(RESPI)

MOTOR <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]])) == paste ("M", SUM[i, ]$VISIT, sep =
  "")

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # Binary physical activity:
  as.numeric (unlist (by (D1$MOTOR,
                          D1$OBS,
                          min, simplify = F)))
})

MOTOR<-unlist(MOTOR)

# # # Calculate HOURS:
HRS <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)

```

```

  POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
  ""))

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # HRS recorder per day:
  as.numeric (unlist (by (D1[D1$WEAR == 1, ]$VM,
                        D1[D1$WEAR == 1, ]$OBS,
                        function (d){
                          ((length (d) * 10)/60)/60 # Calculate to hours
                        }, simplify = F)))

  })
HRS <- unlist (HRS)
sum(HRS)
mean(HRS)

# # # Calculate %active
PA <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
  ""))

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # HRS recorder per day:
  as.numeric (unlist (by (D1[D1$WEAR == 1, ]$VM>=(100/6),
                        D1[D1$WEAR == 1, ]$OBS,
                        function (d){
                          prop.table (table (d)) [2]
                        }, simplify = F)))

  })
PA <- unlist (PA)

# Sustained activity bounds per day (VM >= 100) Slysz et al. 2021
SAB <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
  ""))

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # HRS recorder per day:
  as.numeric (unlist (by (D1[D1$WEAR == 1, ]$VM >= (100/6),
                        D1[D1$WEAR == 1, ]$OBS,
                        function (d){
                          sum (rle (d)$lengths >= (5*6)) # count number
of time sequences during which VM >= 100
                        }, simplify = F)))

  })
SAB <- unlist (SAB)

```

```

# AUC PA histogram Winkler et al. 2005
# Total activity score represents the total count for all the epochs between sleep start and sleep end
# https://www.salusa.se/Filer/Produktinfo/Aktivitet/TheActiwatchUserManualV7.2.pdf
AUC <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep = ""))

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # HRS recorder per day:
  as.numeric (unlist (by (D1[D1$WEAR == 1, ]$VM,
                        D1[D1$WEAR == 1, ]$OBS,
                        function (d){
                          sum(d)
                        }, simplify = F)))

})
AUC <- unlist (AUC)

# # # Calculate Variability in Y-axis
VAR <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep = ""))

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # HRS recorder per day:
  as.numeric (unlist (by (D1[D1$WEAR == 1, ]$A1,
                        D1[D1$WEAR == 1, ]$OBS,
                        function (d){
                          sd (log (d + 1))
                        }, simplify = F)))

})
VAR <- unlist (VAR)

VARII <- lapply (1:nrow (SUM), function (i){

  # Coordinate data:
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep = ""))

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # HRS recorder per day:
  as.numeric (unlist (by (D1[D1$WEAR == 1, ]$VM,
                        D1[D1$WEAR == 1, ]$OBS,
                        function (d){

```

```

        sd (log (d + 1))
      }, simplify = F))
    })
  VARIID <- unlist (VARIID)

  # # # Variability in VM
  VARIID <- lapply (1:nrow (SUM), function (i){

    # Coordinate data:
    POS <- which (names (C) == SUM[i, ]$ID)
    POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
    ""))

    # Data:
    D1 <- C[[POS]][[POSII]]

    # # HRS recorder per day:
    as.numeric (unlist (by (D1[D1$WEAR == 1, ]$VM,
      D1[D1$WEAR == 1, ]$OBS,
      function (d){
        {log (mean (d+1))} * {sd (log (d+1))}
      }, simplify = F))
    })
  VARIID <- unlist (VARIID)

  # # # Variability in VM
  VARIV <- lapply (1:nrow (SUM), function (i){

    # Coordinate data:
    POS <- which (names (C) == SUM[i, ]$ID)
    POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
    ""))

    # Data:
    D1 <- C[[POS]][[POSII]]

    # # HRS recorder per day:
    as.numeric (unlist (by (D1[D1$WEAR == 1, ]$A2,
      D1[D1$WEAR == 1, ]$OBS,
      function (d){
        sd (log (d + 1))
      }, simplify = F))
    })
  VARIV <- unlist (VARIV)

  # # # Variability in VM
  VARV <- lapply (1:nrow (SUM), function (i){

    # Coordinate data:
    POS <- which (names (C) == SUM[i, ]$ID)
    POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
    ""))

    # Data:
    D1 <- C[[POS]][[POSII]]

```



```

# # HRS recorder per day:
as.numeric (unlist (by (D1[D1$WEAR == 1, ]$A3,
                        D1[D1$WEAR == 1, ]$OBS,
                        function (d){
                          sd (log (d + 1))
                        }, simplify = F)))

}))
VARV <- unlist (VARV)

#####
#####
# # # Longitudinal dataframe
#####
#####

SUMIT <- data.frame (ID = ID,
                     VISIT = VISIT,
                     DATE = DATES,
                     HRS = HRS,
                     PA = PA*100,
                     SAB = SAB,
                     AUC = AUC,
                     A1 = VAR, # Y-axis
                     VM_sd = VARII,
                     VM_mul = VARIII,
                     A2 = VARIV,
                     A3 = VARV,
                     ALSFRS = ALSFRS,
                     BULBAR = BULBAR,
                     FINE = FINE,
                     GROSS = GROSS,
                     RESPI = RESPI,
                     MOTOR = MOTOR
                     )

#####
#####
## Calculating number of
## months since baseline visit
#####
#####

SUMIT$TIME<-rep(0,times=nrow(SUMIT))
for (i in 1:42){

  D1<-SUMIT[SUMIT$ID==IDS[i], ]
  baseline<-D1[1,]$DATE

  for (j in 1:nrow(D1)){
    SUMIT[SUMIT$ID==IDS[i], ][j,]$TIME<-interval(baseline, D1[j,]$DATE) %/%
    days(1) / (365.25/12)
  }
}

#####
#####

```

```

## Calculate Longitudinal
## rates of decline
#####

markers<- c('PA', 'SAB', 'AUC', 'A1', 'VM_sd', 'VM_mul', 'A2', 'A3', 'ALSFRS',
'MOTOR', 'BULBAR', 'RESPI')
df_final<-data.frame(matrix(nrow=length(markers),ncol=7))
colnames(df_final)<- c("marker","intercept","slope","CI_lower","CI_upper",
"pvalue","COV")

k<-1

for(i in markers){

  D1 <- SUMIT[, c(i,"TIME","ID")]
  D1 <- as.data.frame(D1)

  fml <- as.formula( paste( i, "~", paste(c("TIME","(TIME|ID)"), col-
lapse="+") ) )
  m<-lmer(fml,data=D1, control = lmerControl(optimizer ="Nelder_Mead"))
  ci<-as.numeric (confint (profile (m, which = 6)))
  sd_slope<-attr(VarCorr(m)$ID,"stddev")[2]
  slope<-fixef(m)[2]

  coefs <- data.frame(coef(summary(m)))
  coefs$sp.z <- 2 * (1 - pnorm(abs(coefs$st.value)))
  p_value <- coefs$sp.z[2]

  # assign fit to list by name
  df_final$marker[k]<-i
  df_final$intercept[k]<-round(fixef(m)[1],2)
  df_final$slope[k]<-round(fixef(m)[2],2)
  df_final$CI_lower[k]<-round(ci[1],2)
  df_final$CI_upper[k]<-round(ci[2],2)
  df_final$pvalue[k] <- round(p_value,3)
  df_final$COV[k]<-round(abs(sd_slope/slope),2)

  k<-k+1
}

#####
## Calculate Longitudinal
## correlations with
## disease progression
#####

markers<- c('PA', 'SAB', 'AUC', 'A1', 'VM_sd', 'VM_mul', 'A2', 'A3')
df_final<-data.frame(matrix(nrow=length(markers),ncol=13))
colnames(df_final)<- c("marker","correlation_ALSFRS","CI_lower_ALS-
FRS","CI_upper_ALSFRS","correlation_MOTOR","CI_lower_MOTOR","CI_upper_MO-
TOR","correlation_BULBAR","CI_lower_BULBAR","CI_upper_BULBAR","correla-
tion_RESPI","CI_lower_RESPI","CI_upper_RESPI")

k<-1

for(i in markers){

```

```

D1 <- SUMIT[, c(i, "ALSFRS", "ID")]
D1[, c(i, "ALSFRS")] <- scale(D1[, c(i, "ALSFRS")])
D1 <- as.data.frame(D1)

fml <- as.formula( paste( i, "~", paste(c("ALSFRS", "(ALSFRS|ID)"), col-
lapse="+") ) )
m<-lmer(fml, data=D1)
ci<-as.numeric (confint (profile (m, which = 6)))

df_final$marker[k]<-i
df_final$correlation_ALSFRS[k]<-round(fixef(m) [2], 2)
df_final$CI_lower_ALSFRS[k]<-round(ci [1], 2)
df_final$CI_upper_ALSFRS[k]<-round(ci [2], 2)

D1 <- SUMIT[, c(i, "MOTOR", "ID")]
D1[, c(i, "MOTOR")] <- scale(D1[, c(i, "MOTOR")])
D1 <- as.data.frame(D1)

fml <- as.formula( paste( i, "~", paste(c("MOTOR", "(MOTOR|ID)"), col-
lapse="+") ) )
m<-lmer(fml, data=D1)
ci<-as.numeric (confint (profile (m, which = 6)))

df_final$correlation_MOTOR[k]<-round(fixef(m) [2], 2)
df_final$CI_lower_MOTOR[k]<-round(ci [1], 2)
df_final$CI_upper_MOTOR[k]<-round(ci [2], 2)

D1 <- SUMIT[, c(i, "BULBAR", "ID")]
D1[, c(i, "BULBAR")] <- scale(D1[, c(i, "BULBAR")])
D1 <- as.data.frame(D1)

fml <- as.formula( paste( i, "~", paste(c("BULBAR", "(BULBAR|ID)"), col-
lapse="+") ) )
m<-lmer(fml, data=D1)
ci<-as.numeric (confint (profile (m, which = 6)))

df_final$correlation_BULBAR[k]<-round(fixef(m) [2], 2)
df_final$CI_lower_BULBAR[k]<-round(ci [1], 2)
df_final$CI_upper_BULBAR[k]<-round(ci [2], 2)

D1 <- SUMIT[, c(i, "RESPI", "ID")]
D1[, c(i, "RESPI")] <- scale(D1[, c(i, "RESPI")])
D1 <- as.data.frame(D1)

fml <- as.formula( paste( i, "~", paste(c("RESPI", "(RESPI|ID)"), col-
lapse="+") ) )
m<-lmer(fml, data=D1)
ci<-as.numeric (confint (profile (m, which = 6)))

df_final$correlation_RESPI[k]<-round(fixef(m) [2], 2)
df_final$CI_lower_RESPI[k]<-round(ci [1], 2)
df_final$CI_upper_RESPI[k]<-round(ci [2], 2)

k<-k+1
}

```

Appendix C: AI-Data R-Code

Missingness

```
library(tidyverse)
library(read.gt3x)
library(gtools)

#load gt3x data files
FILE<- mixedsort(dir("C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/raw_data"))

B <-openxlsx::read.xlsx ("C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/ActigraphyProjectALS.xlsx", "BASELINE", detectDates = T)
B$SID <- paste ("ID", as.numeric (gsub (x = B$SID, "031-", replacement =
"")), sep = "")

#####
# Weartime data
#####

# Load in wear time data
W <- openxlsx::read.xlsx ( "C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/ActigraphyProjectALS.xlsx", sheet = "WEARTIME") # Add Excel file
location
W <- (W[, c ("ID", "START", "STOP", "VISIT", "tSTART", "tSTOP", "TO-
TAL", "BULBAR", "FINE", "GROSS", "RESPI", "MOTOR")])
W$START <- openxlsx::convertToDate(W$START)
W$STOP <- openxlsx::convertToDate(W$STOP)
W$tSTART <- format(as.POSIXct((W$tSTART) * 86400, origin = "1970-01-01", tz
= "UTC"), "%H:%M:%S")
W$tSTOP <- format(as.POSIXct((W$tSTOP) * 86400, origin = "1970-01-01", tz =
"UTC"), "%H:%M:%S")

Fi <- 1
rexp <- "^(\\w+)\\s?(.*)$"
IDs <- sub (rexp,"\\1", FILE)
VISITs <- gsub (pattern = ".gt3x" , x = sub (rexp,"\\2", FILE), replacement
= "")
VISITs <- ifelse (VISITs == "Baseline", "Month 0", VISITs)
VISITs <- as.numeric (gsub (pattern = "Month ", "", VISITs))

W[!paste (W$ID, W$VISIT, sep = ":") %in% paste (IDs, VISITs, sep = ":"), ]
W <- W[paste (W$ID, W$VISIT, sep = ":") %in% paste (IDs, VISITs, sep =
":"), ]

#####
# Missingness
#####

missing_old<-data.frame()

#iterate through all 207 raw GT3X files
for (i in 1:length(FILE)){
  gc()

  # Step 1: load gt3x file in R
  raw<-read.gt3x(paste("C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/raw_data/", FILE[i], sep=""))
```

```

ID <- sub (rexp,"\\1", FILE[i])
VISIT <- gsub (pattern = ".gt3x" , x = sub (rexp,"\\2", FILE[i]), re-
placement = "")
VISIT <- ifelse (VISIT == "Baseline", "Month 0", VISIT)
VISIT <- as.numeric (gsub (pattern = "Month ", "", VISIT))
SELECT <- W[W$ID == ID & W$VISIT == VISIT, ]

#Step2: Get missing observations
missing<-attributes(raw)[setdiff(names(attributes(raw)), c("dim",
"dimnames", "time_index"))]$missingness
colnames(missing)[1] <- "timestamp"
missing$date<-as.Date(missing$timestamp)

#Step 3: Select data rows and collapse:
missing <- missing[((missing$date >= SELECT$START) & (missing$date <= SE-
LECT$STOP)), ]
missing$date <- missing$timestamp
missing$date <- as.Date (missing$timestamp)
missing$time <- strftime (missing$timestamp, format="%H:%M:%S", tz =
"UTC")
missing$SELECT <- ifelse ((missing$date != SELECT$START) & (miss-
ing$date != SELECT$STOP), T,
                        ifelse ((missing$date == SELECT$START) & (miss-
ing$time >= SELECT$tSTART), T,
                                ifelse ((missing$date == SELECT$STOP) &
(missing$time <= SELECT$tSTOP), T, F)))
missing <- missing[missing$SELECT, ]

#Step 4: Save missing observations in data frame
if(nrow(missing)>0){
  missing$ID<-ID
  missing$VISIT<-VISIT
  merged_missing<-rbind(missing_old,missing)
  missing_old<-merged_missing
}

print(paste("FILE", i, ": done"))
}

#subset data frame to day and night
merged_night<-merged_missing[(as.POSIXct(merged_missing$time,format =
"%H:%M:%S")>as.POSIXct("22:00:00",format = "%H:%M:%S")) |
(as.POSIXct(merged_missing$time,format =
"%H:%M:%S")<as.POSIXct("10:00:00",format = "%H:%M:%S")),]
merged_day<-merged_missing[(as.POSIXct(merged_missing$time,format =
"%H:%M:%S")>=as.POSIXct("10:00:00",format = "%H:%M:%S")) &
(as.POSIXct(merged_missing$time,format =
"%H:%M:%S")<=as.POSIXct("22:00:00",format = "%H:%M:%S")),]

#calculate mean length of missingness in a row for day and night
mean(merged_day$n_missing)
mean(merged_night$n_missing)
sd(merged_day$n_missing)
sd(merged_night$n_missing)

#calculate total missingness for day and night
sum(merged_day$n_missing)
sum(merged_night$n_missing)

```

Van Hees non-wear algorithm

```
hees_2013_non_wear_time<-function(data, hz = 30, min_non_wear_time_window =
135, window_overlap=15, std_mg_threshold = 7.0, std_min_num_axes = 1,
value_range_mg_threshold = 1.0, value_range_min_num_axes = 1, nwt_encoding
= 0, wt_encoding = 1){
  data_timestamped<-data
  data<-data %>% select(2,3,4)
  num_samples_per_min <- hz * 60

  min_non_wear_time_window<-min_non_wear_time_window* num_samples_per_min
  window_overlap<-window_overlap* num_samples_per_min

  std_mg_threshold<-std_mg_threshold/1000
  value_range_mg_threshold <- value_range_mg_threshold/1000

  non_wear_vector<-rep(wt_encoding, times=nrow(data))
  for (i in seq(1, nrow(data), by=window_overlap)){
    start<-i
    end<-i + min_non_wear_time_window
    subset_data <-data %>% slice(start:end)

    n<-nrow(subset_data)

    # check if the data sequence has been exhausted, meaning that there are
    no full windows left in the data sequence (this happens at the end of the
    sequence)
    if (nrow(subset_data) < min_non_wear_time_window){
      break
    }

    std<-apply(subset_data,2,sd)

    if(sum(std < std_mg_threshold, na.rm=TRUE)>=std_min_num_axes){
      non_wear_vector[start:end] <- nwt_encoding
    }

    value_range<-apply(subset_data,2,diff_min_max)

    if(sum(value_range < value_range_mg_threshold,
na.rm=TRUE)>=value_range_min_num_axes){
      non_wear_vector[start:end]<-nwt_encoding
    }

  }

  if(length(non_wear_vector)>nrow(data_timestamped)){
    non_wear_vector<-head(non_wear_vector,-(length(non_wear_vector)-
nrow(data_timestamped)))
  }

  data_timestamped$wear<-non_wear_vector
  return(data_timestamped)
}
```

Systematic Noise Calculation

```
library(ggplot2)
library(tidyverse)
library(read.gt3x)
library(lubridate)
library(ActivityIndex)
library(gtools)
library(data.table)
library(GGIR)
library(matrixStats)
library(lme4)
library(reshape2)

# get raw accelerometer data
FILE<- mixedsort(dir("C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/raw_data")) #add personal gt3x file location

#get information form additional excel sheet
B <-openxlsx::read.xlsx ("C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/ActigraphyProjectALS.xlsx", "BASELINE", detectDates = T)
B$SID <- paste ("ID", as.numeric (gsub (x = B$SID, "031-", replacement =
"")), sep = "")

# Load in wear time data
W <- openxlsx::read.xlsx ( "C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/ActigraphyProjectALS.xlsx", sheet = "WEARTIME") # Add Excel file
location
W <- (W[, c ("ID", "START", "STOP", "VISIT", "tSTART", "tSTOP", "TO-
TAL", "BULBAR", "FINE", "GROSS", "RESPI", "MOTOR")])
W$START <- openxlsx::convertToDate(W$START)
W$STOP <- openxlsx::convertToDate(W$STOP)
W$tSTART <- format(as.POSIXct((W$tSTART) * 86400, origin = "1970-01-01", tz
= "UTC"), "%H:%M:%S")
W$tSTOP <- format(as.POSIXct((W$tSTOP) * 86400, origin = "1970-01-01", tz =
"UTC"), "%H:%M:%S")

Fi <- 1
rexp <- "^(\\w+)\\s?(.*)$"
IDs <- sub (rexp, "\\1", FILE)
VISITs <- gsub (pattern = ".gt3x" , x = sub (rexp, "\\2", FILE), replacement
= "")
VISITs <- ifelse (VISITs == "Baseline", "Month 0", VISITs)
VISITs <- as.numeric (gsub (pattern = "Month ", "", VISITs))

W[!paste (W$ID, W$VISIT, sep = ":") %in% paste (IDs, VISITs, sep = ":"), ]
W <- W[paste (W$ID, W$VISIT, sep = ":") %in% paste (IDs, VISITs, sep =
":"), ]

# extract name of serial numbers for each file/measurement to get unique
serial numbers used in the study

df_sigma<-data.table(matrix(nrow=length(FILE),ncol=3))
colnames(df_sigma)<- c("serialnumber","index","file")
for (i in 1:length(FILE)){
  gc()
  raw<-read.gt3x(paste("C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/raw_data/", FILE[i], sep=""))
```

```

    df_sigma$serialnumber[i]<-attributes(raw)[setdiff(names(attributes(raw)),
c("dim", "dimnames", "time_index"))]$header[[1]]
    df_sigma$index[i]<-i
    df_sigma$file[i]<-FILE[i]
  }

#count how often each serial number was used during the study
df_sigma<-df_sigma[order(serialnumber)]
FILE<-df_sigma$file
serialnumbers<-as.data.table(table(df_sigma$serialnumber))$V1
number_used<-as.data.table(table(df_sigma$serialnumber))$N

aggregated_number_used<-vector()
k<-0
for(i in 1:length(number_used)){
  aggregated_number_used[i]<-number_used[i]+k
  k<-aggregated_number_used[i]
}

# function for calculation of systematic noise
sigma_all_axis = function(x, hertz = 30)
{
  # stopifnot(is.data.table(x))
  x = as.data.table(x)
  X = Y = Z = NULL
  rm(list = c("X", "Y", "Z"))
  n = nrow(x) %/% hertz*hertz
  result = mean(x[1:n,
                    sqrt((rowVars(matrix(X, ncol = hertz, byrow=TRUE))+
                        rowVars(matrix(Y, ncol = hertz, byrow=TRUE))+
                        rowVars(matrix(Z, ncol = hertz, byrow=TRUE)))/3)]
  )
  names(result) = c("SD")
  return(result)
}

mean_serial_number_all<-vector()
sd_all<-vector()
sample_sigma_all_list<-list()
nonwear_found<-0
# iterating through all files based on serial numbers
for (j in 1:length(serialnumbers)){
  gc()
  k<-1
  sample_sigma_all<-vector()

  for (i in max(1,aggregated_number_used[j-1]+1):(aggregated_number_used[j])){

    ## Step 1: load gt3x file in R
    raw<-read.gt3x(paste("C:/Users/enied/OneDrive/Dokumente/ThesisProject_ALS/raw_data/", FILE[i], sep=""))
    raw<-as.data.frame(raw)

    ID <- sub (rexp,"\\1", FILE[i])
    VISIT <- gsub (pattern = ".gt3x" , x = sub (rexp,"\\2", FILE[i]), replacement = "")
  }
}

```



```

VISIT <- ifelse (VISIT == "Baseline", "Month 0", VISIT)
VISIT <- as.numeric (gsub (pattern = "Month ", "", VISIT))

# calculating non-wear time
raw<- hees_2013_non_wear_time(raw)

raw_nonwear<-raw %>% filter(wear == 0)
raw_nonwear$Index<-raw_nonwear$time
raw_nonwear<-raw_nonwear[, c('Index', 'X', 'Y', 'Z')]

if((nrow(raw_nonwear)%30)>0){
  raw_nonwear<-head(raw_nonwear, -(nrow(raw_nonwear)%30))
}

#calculating systematic noise based on non-wear time
if(nrow(raw_nonwear)>0){
  sample_sigma_all[k] <- sigma_all_axis(raw_nonwear)
  nonwear_found<-nonwear_found+1
}

else{
  sample_sigma_all[k]<-NA
}

print (paste ("Sigmas calculated for", ID, "- Month", VISIT))

k=k+1
}

sample_sigma_all_list[[j]]<-sample_sigma_all
mean_serial_number_all[j]<-mean(sample_sigma_all,na.rm=TRUE)
sd_all[j]<-sd(sample_sigma_all,na.rm=TRUE)

print (paste ("mean calculated for serialnumber", j))
}

#calculate standard deviation of systematic noises to check if they can be
combined
standard_deviation_sigma_all<-sd(mean_serial_number_all,na.rm=TRUE)

#calculate mean of systematic noises of the different serial numbers
final_sigma_all<-mean(mean_serial_number_all,na.rm=TRUE)

```

Epoch-level AI data

```
library(ggplot2)
library(tidyverse)
library(read.gt3x)
library(lubridate)
library(ActivityIndex)
library(gtools)
library(data.table)
library(GGIR)
library(matrixStats)
library(lme4)
library(reshape2)

# calculating weights for modified AI metric
x <- seq(0, 1, by=0.1)
y <- seq(0, 1, by=0.1)
z <- seq(0, 1, by=0.1)
df_weights <- as.data.frame(expand.grid(x = x, y = y, z=z))
df_weights$sum <- rowSums(df_weights)
df_weights <- df_weights[df_weights$sum == 1, ]
equal_weights <- c(1/3, 1/3, 1/3, 1)
df_weights <- rbind(df_weights, equal_weights)

#define modified AI function
computeActivityIndex_all = function(x, weights, sigma0=final_sigma_all, epoch=NULL, hertz=30)
{
  Index = NULL
  rm("Index")

  x = as.data.table(x)
  n = nrow(x) %/% hertz * hertz
  result = array(0, c(n %/% hertz, 2))
  result[, 2] = x[1:n, (weights[1] * (rowSds(matrix(x$X, ncol=hertz, byrow=TRUE))^2 - sigma0^2)) +
    (weights[2] * (rowSds(matrix(x$Y, ncol=hertz, byrow=TRUE))^2 - sigma0^2)) +
    (weights[3] * (rowSds(matrix(x$Z, ncol=hertz, byrow=TRUE))^2 - sigma0^2))]
  result[which(result[, 2] < 0), 2] = 0
  result[, 2] = sqrt(result[, 2])
  result = as.data.frame(result, stringsAsFactors=FALSE)
  # is Index a column name or some other thing?
  result[, 1] = x[(1:(n %/% hertz) - 1) * hertz + 1, Index] # fetch "Index"
  if (epoch > 1)
  {
    L_AI = length(result[, 2])
    result0 = as.data.frame(array(0, c(L_AI %/% epoch, 2)), stringsAsFactors=FALSE)
    result0[, 2] = as.numeric(rowSums(matrix(result[1:(L_AI - L_AI %/% epoch), 2], ncol=epoch, byrow=TRUE)))
    result0[, 1] = result[(1:(L_AI %/% epoch) - 1) * epoch + 1, 1]
    result = result0
  }
  colnames(result) = c("RecordNo", "AI")
  class(result) = c("ActivityIndex", class(result))
  return(result)
}
```

```

# defining datafile [per ID for each visit]:
C <- apply (1:length (B$SID), function (i){

  # List per ID
  x <- list (dim (0))
  names (x) <- paste (B$SID[i])

  # List visit
  x[[B$SID[i]]] <- lapply (IDs[IDs == B$SID[i]], function (x){dim (0)})
  names (x[[B$SID[i]]]) <- paste ("M", VISITs[IDs == B$SID[i]], sep = "")
  x
})

```

epoch <-10 #choosing desired epoch length,in the data was summarized into 10 different epoch lengths resulting in 10 different data sets D1-D10

```

#iterate through all gt3x files
for (i in 1:length(FILE)){
  gc()

  # # Step 1: load gt3x file in R
  raw<-read.gt3x(paste("C:/Users/enied/OneDrive/Dokumente/ThesisPro-
ject_ALS/raw_data/", FILE[i], sep=""))
  raw<-as.data.frame(raw)
  colnames(raw)[1] <- "timestamp"
  raw$date<-as.Date(raw$timestamp)
  # Make ID & VISIT
  ID <- sub (rexp,"\\1", FILE[i])
  VISIT <- gsub (pattern = ".gt3x" , x = sub (rexp,"\\2", FILE[i]), re-
placement = "")
  VISIT <- ifelse (VISIT == "Baseline", "Month 0", VISIT)
  VISIT <- as.numeric (gsub (pattern = "Month ", "", VISIT))
  SELECT <- W[W$ID == ID & W$VISIT == VISIT, ]
  ALSFRS <- W[W$ID == ID & W$VISIT == VISIT, ]$TOTAL
  BULBAR <- W[W$ID == ID & W$VISIT == VISIT, ]$BULBAR
  FINE <- W[W$ID == ID & W$VISIT == VISIT, ]$FINE
  GROSS <- W[W$ID == ID & W$VISIT == VISIT, ]$GROSS
  RESPI <- W[W$ID == ID & W$VISIT == VISIT, ]$RESPI
  MOTOR <- W[W$ID == ID & W$VISIT == VISIT, ]$MOTOR

  # Check if there are missing dates
  if (is.na (SELECT$START) | is.na (SELECT$START)){

    print (paste ("Missing dates", ID, "- Month", VISIT))

    # # # Save NA in predefined list
    POS <- which (names (C) == ID)
    POSII <- which (names (C[[POS ]]) == paste ("M", VISIT, sep = ""))
    C[[POS]][[POSII]] <- NA

  } else {

    # # Check 2: file in W?
    if (nrow (SELECT) != 1){
      print ("Dates are not correct for:", ID, VISIT)
      break
    } else {

```

```

    if ((SELECT$START >= min (raw$date)) & (SELECT$STOP <= max
(raw$date))) {
      print (paste ("Dates check OK for", ID, "- Month", VISIT))
    } else {
      print (paste ("Actigraph dates differ from W:", ID, VISIT))
      break
    }
  }
}

# Step 2: Select datarows and collapse:
raw <- raw[((raw$date >= SELECT$START) & (raw$date <= SELECT$STOP)), ]
raw$date <- raw$timestamp
raw$sdate <- as.Date (raw$timestamp)
raw$time <- strptime (raw$timestamp, format="%H:%M:%S", tz = "UTC")
raw$SELECT <- ifelse ((raw$sdate != SELECT$START) & (raw$sdate != SE-
LECT$STOP), T,
                     ifelse ((raw$sdate == SELECT$START) & (raw$time
>= SELECT$tSTART), T,
                             ifelse ((raw$sdate == SELECT$STOP) &
(raw$time <= SELECT$tSTOP), T, F)))
raw <- raw[raw$SELECT, ]

# Step 3: Data cleaning and reformating:

#impute missing values
raw<-g.imputeTimegaps(raw, xyzCol=c("X","Y","Z"), timeCol = "timestamp"
,sf=30,k=2/30)

#calculate wear time using van Hees non-wear algortihm
raw<- hees_2013_non_wear_time(raw)
raw <- raw[order(raw$date), ]
raw<-raw %>% filter(wear == 1)
raw$Index<-raw$timestamp

final_wear<-raw[, c('Index', 'X', 'Y','Z')]
if((nrow(final_wear)%30>0){
  final_wear<-head(final_wear, -(nrow(final_wear)%30))
}

#calculate modified AIs, AI is calculated for each set of weights re-
sulting in 61 different AIs that are saved in different columns
activity_index<-data.frame(matrix(nrow=nrow(fi-
nal_wear)/(epoch*30),ncol=nrow(df_weights)+1))
t<-vector()
t[1]<-"RecordNo"
for (n in 1:nrow(df_weights)){
  t[n+1]<-paste("AI_weight",n,sep = '')
}
colnames(activity_index)<-t

weights<-as.numeric(df_weights[1,])

activity_index$RecordNo <-computeActivityIndex_all(fi-
nal_wear,weights=weights,sigma0 = final_sigma_all, epoch = epoch,hertz =
30)$RecordNo
activity_index[, 2]<- computeActivityIndex_all(fi-
nal_wear,weights=weights,sigma0 = final_sigma_all, epoch = epoch,hertz =
30)$AI

```

```

for (j in 2:(nrow(df_weights))){

  weights<-as.numeric(df_weights[j,])
  activity_index[, j+1]<- computeActivityIndex_all(fi-
nal_wear,weights=weights,sigma0 = final_sigma_all, epoch = epoch ,hertz =
30)$AI
  }

  activity_index$date <- activity_index$RecordNo
  activity_index$sdate <- as.Date (activity_index$RecordNo)
  activity_index$time <- strftime (activity_index$RecordNo, for-
mat="%H:%M:%S", tz = "UTC")
  activity_index$ID <- ID
  activity_index$VISIT <- VISIT
  activity_index$ALSFRS <- ALSFRS
  activity_index$BULBAR <- BULBAR
  activity_index$FINE <- FINE
  activity_index$GROSS <- GROSS
  activity_index$RESPI <-RESPI
  activity_index$MOTOR<-MOTOR

  m<-as.data.frame(table(activity_index$date))
  day_int<-vector()
  for (j in 1:nrow(m)){
    day_int<-append(day_int,rep(j, times=m$Freq[j]))
  }

  activity_index$OBS<-day_int

  #subset dataframe into number of active epochs per day, transform to
  active hours, and remove days with less than 8h of weartime
  activity_index <- merge (activity_index, data.frame (QUOTE = (unlist
(by (activity_index, activity_index$OBS, nrow, simplify =
F))/(60*(60/epoch))) >= 8,
OBS = names (un-
list (by (activity_index, activity_index$OBS, nrow, simplify = F))))), by =
"OBS")

  activity_index <- activity_index[activity_index$QUOTE, ]

  POS <- which (names (C) == ID)
  POSII <- which (names (C[[POS]]) == paste ("M", VISIT, sep = ""))
  C[[POS]][[POSII]] <- activity_index
  print (paste ("Calculation of Activity index", ID, "- Month", VISIT ,
"successful"))

}

}

```

Defining AI-based markers and generating analysis set

```
library(tidyverse)
library(read.gt3x)
library(lubridate)
library(ActivityIndex)
library(gtools)
library(data.table)
library(GGIR)
library(matrixStats)
library(lme4)
library(reshape2)

C<-readRDS(file.choose()) #choose one the 10 data sets D1-D10

epoch<-7200 #specify which epoch length the data set contains

IDS <- names (sapply(C, function (d){names (d)}))
rep (IDS, sapply (sapply (C, function (d){names (d)}), function (d){length
(d)}))
SUM <- data.frame (ID = rep (IDS, sapply (sapply (C, function (d){names
(d)}), function (d){length (d)})),
                  M = do.call ("c", sapply (C, function (d){names (d)})))
SUM$VISIT <- gsub (SUM$M, pattern = "M", replacement = "")

DATES <- lapply (1:nrow (SUM), function (i){
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

  D1 <- C[[POS]][[POSII]]

  as.numeric (unlist (by (as.Date (D1$date),
                           D1$OBS,
                           min, simplify = F)))
})
DATES <- as.Date.numeric (unlist (DATES), origin = "1970-01-01")

ALSFRS <- lapply (1:nrow (SUM), function (i){
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

  D1 <- C[[POS]][[POSII]]

  as.numeric (unlist (by (D1$ALSFRS,
                           D1$OBS,
                           min, simplify = F)))
})
ALSFRS<-unlist(ALSFRS)

BULBAR <- lapply (1:nrow (SUM), function (i){
  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

  D1 <- C[[POS]][[POSII]]
```

```

    as.numeric (unlist (by (D1$BULBAR,
                           D1$OBS,
                           min, simplify = F)))
  })

BULBAR<-unlist(BULBAR)

FINE <- lapply (1:nrow (SUM), function (i){

  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]]) == paste ("M", SUM[i, ]$VISIT, sep =
  ""))

  # Data:
  D1 <- C[[POS]][[POSII]]

  # # Binary physical activity:
  as.numeric (unlist (by (D1$FINE,
                           D1$OBS,
                           min, simplify = F)))
  })
FINE<-unlist(FINE)

GROSS <- lapply (1:nrow (SUM), function (i){

  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]]) == paste ("M", SUM[i, ]$VISIT, sep =
  ""))

  D1 <- C[[POS]][[POSII]]

  as.numeric (unlist (by (D1$GROSS,
                           D1$OBS,
                           min, simplify = F)))
  })

GROSS<-unlist(GROSS)

RESPI <- lapply (1:nrow (SUM), function (i){

  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]]) == paste ("M", SUM[i, ]$VISIT, sep =
  ""))

  D1 <- C[[POS]][[POSII]]

  as.numeric (unlist (by (D1$RESPI,
                           D1$OBS,
                           min, simplify = F)))
  })
RESPI<-unlist(RESPI)

MOTOR <- lapply (1:nrow (SUM), function (i){

  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS]]) == paste ("M", SUM[i, ]$VISIT, sep =
  ""))

  D1 <- C[[POS]][[POSII]]

```

```

as.numeric (unlist (by (D1$MOTOR,
                        D1$OBS,
                        min, simplify = F)))
}))
MOTOR<-unlist(MOTOR)

ID <- lapply (1:nrow (SUM), function (i){

  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

  D1 <- C[[POS]][[POSII]]

  as.character (unlist (by (D1$ID,
                           D1$OBS,
                           unique, simplify = F)))
}))
ID <- unlist (ID)

VISIT <- lapply (1:nrow (SUM), function (i){

  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

  D1 <- C[[POS]][[POSII]]

  as.numeric (unlist (by (D1$VISIT,
                          D1$OBS,
                          unique, simplify = F)))
}))
VISIT <- unlist (VISIT)

HRS <- lapply (1:nrow (SUM), function (i){

  POS <- which (names (C) == SUM[i, ]$ID)
  POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
""))

  D1 <- C[[POS]][[POSII]]

  as.numeric (unlist (by (D1$RecordNo ,
                          D1$OBS,
                          function (d){
                            ((length (d) * epoch)/60)/60 }, simplify = F)))
}))
HRS <- unlist (HRS)

sum(HRS) #calculate total wear time
mean(HRS) #calculate average daily wear time

#####
#####
# # # Longitudinal dataframe
#####
#####

```



```

#create empty dataframe to fill
SUMIT<-data.frame(matrix(nrow=length(DATES),ncol=10+(61*3)))

# create column names for all 61 markers of the three marker types
colnames_meanAI<-vector()
colnames_P90AI<-vector()
colnames_upperAI<-vector()
for (i in 1:61){
  colnames_meanAI[i]<-paste('meanAI_',i,sep='')
  colnames_P90AI[i]<-paste('P90AI_',i,sep='')
  colnames_upperAI[i]<-paste('upperAI_',i,sep='')
}

colnames(SUMIT)<-c('ID','VISIT','DATE','HRS','ALSFRS','BULB-
AR','FINE','GROSS','RESPI','MO-
TOR',colnames_meanAI,colnames_P90AI,colnames_upperAI)

# add additional information to longitudinal data frame
SUMIT$ID <- ID
SUMIT$VISIT <- VISIT
SUMIT$DATE <- DATES
SUMIT$HRS <- HRS
SUMIT$ALSFRS <- ALSFRS
SUMIT$BULBAR <- BULBAR
SUMIT$FINE <- FINE
SUMIT$GROSS <- GROSS
SUMIT$RESPI <- RESPI
SUMIT$MOTOR <- MOTOR

# functions for the calculation of the average of quantiles, functions
taken from https://stackoverflow.com/questions/19598544/compute-the-average-for-quantiles
# user: Michal Polowczyk

averageQuantile<- function(vec, value, value2) {
  chunk = getChunkOfVector(vec, value, value2)
  if(length(chunk) >0) {
    return(mean(chunk))
  }
  return(0.0)
}

getChunkOfVector<- function(vector, value, value2) {
  len = length(vector)
  result<-vector()
  vector<-sort(vector)
  k<-1
  for(i in vector){
    if(k/len > value & k/len <= value2) {
      result = append(result, i)
    }
    k<-k+1
  }
  return(result)
}

k<-11

```

```

# fill longitudinal dataframe with markers, one marker for each modified AI
for (j in 3:63){

  # calculate Mean AI
  meanAI <- lapply (1:nrow (SUM), function (i){

    POS <- which (names (C) == SUM[i, ]$ID)
    POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
    ""))

    D1 <- C[[POS]][[POSII]]

    as.numeric (unlist (by (D1[,j],
                          D1$OBS,
                          function (d){
                            log(mean(d+1))
                          }, simplify = F)))

  })
  SUMIT[,k]<-unlist(meanAI)

#Calculate 90P
  Q90AI <- lapply (1:nrow (SUM), function (i){

    # Coordinate data:
    POS <- which (names (C) == SUM[i, ]$ID)
    POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
    ""))

    # Data:
    D1 <- C[[POS]][[POSII]]

    # # HRS recorder per day:
    as.numeric (unlist (by (D1[,j],
                          D1$OBS,
                          function (d){
                            log(quantile(d+1,probs=c(0.90)))
                          }, simplify = F)))

  })

  SUMIT[,k+(1*61)]<-unlist(Q90AI)

#Calculate Upper AI
  meanQ75to1AI <- lapply (1:nrow (SUM), function (i){

    # Coordinate data:
    POS <- which (names (C) == SUM[i, ]$ID)
    POSII <- which (names (C[[POS ]]) == paste ("M", SUM[i, ]$VISIT, sep =
    ""))

    # Data:
    D1 <- C[[POS]][[POSII]]

    # # HRS recorder per day:
    as.numeric (unlist (by (D1[,j],
                          D1$OBS,
                          function (d){
                            log(averageQuantile(d+1,0.75,1))
                          }, simplify = F)))

  })

```

```
SUMIT[,k+(2*61)]<-unlist(meanQ75tolAI)
k<-k+1
}
```

Longitudinal rates of decline in activity markers or disease progression

```
library(ggplot2)
library(tidyverse)
library(read.gt3x)
library(lubridate)
library(ActivityIndex)
library(gtools)
library(data.table)
library(GGIR)
library(matrixStats)
library(lme4)
library(reshape2)

SUMIT<-readRDS(choose.files()) #choose one the 10 longitudinal dataframes
(i.e., choose which epoch length to look at)

#add number of month since baseline visit column to the longitudinal data-
frame
SUMIT$TIME<-rep(0,times=nrow(SUMIT))
for (i in 1:42){
  # Data:
  D1<-SUMIT[SUMIT$ID==IDS[i], ]
  baseline<-D1[1,]$DATE

  for (j in 1:nrow(D1)){
    SUMIT[SUMIT$ID==IDS[i], ][j,]$TIME <- interval(baseline, D1[j,]$DATE)
  }
  %% days(1) / (365.25/12)
}

}

#calculating longitudinal rates of decline
markers<- c('ALSFRS','BULBAR','FINE','GROSS','RESPI','MO-
TOR',colnames_meanAI,colnames_P90AI,colnames_upperAI)
df_final<-data.frame(matrix(nrow=length(markers),ncol=7))
colnames(df_final)<- c("marker","intercept","slope","CI_lower","CI_up-
per","pvalue","COV")

k<-1
for(i in markers){ #loop through all clinical and AI markers

  # create temporary data matrix and model formula
  D1 <- SUMIT[, c(i,"TIME","ID")]
  D1 <- as.data.frame(D1)

  fml <- as.formula( paste( i, "~", paste(c("TIME","(TIME|ID)"), col-
lapse="+") ) ) #use linear mixed model
m<-lmer(fml,data=D1, control = lmerControl(optimizer ="Nelder_Mead"))
ci<-as.numeric( confint( profile( m, which = 6))) #confidence intervals
sd_slope<-attr(VarCorr(m)$ID,"stddev")[2] #standard deviation of the
slope
slope<-fixef(m)[2] #fixed effect for time, i.e. slope

#calculate p-values using normal approximation
coefs <- data.frame(coef(summary(m)))
coefs$P.z <- 2 * (1 - pnorm(abs(coefs$t.value)))
```

```

p_value <- coefs$p.z[2]

df_final$marker[k]<-i
df_final$intercept[k]<-round(fixef(m)[1],4)
df_final$slope[k]<-round(fixef(m)[2],4)
df_final$CI_lower[k]<-round(ci[1],4)
df_final$CI_upper[k]<-round(ci[2],4)
df_final$pvalue[k] <- round(p_value,4)
df_final$COV[k]<-round(abs(sd_slope/slope),4)

k<-k+1
}

```

Longitudinal correlations between AI-based markers and clinical markers of disease severity

```
library(ggplot2)
library(tidyverse)
library(read.gt3x)
library(lubridate)
library(ActivityIndex)
library(gtools)
library(data.table)
library(GGIR)
library(matrixStats)
library(lme4)
library(reshape2)

SUMIT<-readRDS(choose.files()) ##choose one the 10 longitudinal dataframes
(i.e., choose which epoch length to look at)

markers<-c(colnames_meanAI,colnames_P90AI,colnames_upperAI)
df_final<-data.table(matrix(nrow=length(markers),ncol=17))
colnames(df_final)<- c("marker","correlation_ALSFERS","CI_lower_ALS-
FRS","CI_upper_ALSFERS","Pvalue_ALSFERS","correlation_MOTOR","CI_lower_MO-
TOR","CI_upper_MOTOR","Pvalue_MOTOR","correla-
tion_BULBAR","CI_lower_BULBAR","CI_upper_BULBAR","Pvalue_BULBAR","correla-
tion_RESPI","CI_lower_RESPI","CI_upper_RESPI","Pvalue_RESPI")

k<-1
# loop over all AI 183 markers and calculate longitudinal correlations with
clinical markers of disease severity
for(i in markers){

  #calculate longitudinal correlation with ALSFRS-R score
  D1 <- SUMIT[, c(i,"ALSFRS","ID")]
  D1[,c(i,"ALSFRS")]<-scale(D1[,c(i,"ALSFRS")]) #scaling columns
  D1 <- as.data.frame(D1)

  fml <- as.formula( paste( i, "~", paste(c("ALSFRS","(ALSFRS|ID)"), col-
lapse="+" ) ) )
  m<-lmer(fml,data=D1,control = lmerControl(optimizer ="Nelder_Mead")) #use
mixed model
  ci<-as.numeric (confint (profile (m, which = 6))) #confidence interval

  #calculate p-values using normal approximation
  coefs <- data.frame(coef(summary(m)))
  coefs$p.z <- 2 * (1 - pnorm(abs(coefs$t.value)))
  p_value <- coefs$p.z[2]

  df_final$marker[k]<-i
  df_final$correlation_ALSFERS[k]<-round(fixef(m)[2],4) #fixed effect for
slope, i.e. longitudinal correlation
  df_final$CI_lower_ALSFERS[k]<-round(ci[1],4)
  df_final$CI_upper_ALSFERS[k]<-round(ci[2],4)
  df_final$Pvalue_ALSFERS[k]<-round(p_value,4)

  #calculate longitudinal correlation with MOTOR score
  D1 <- SUMIT[, c(i,"MOTOR","ID")]
  D1[,c(i,"MOTOR")]<-scale(D1[,c(i,"MOTOR")])
  D1 <- as.data.frame(D1)
```

```

fml <- as.formula( paste( i, "~", paste(c("MOTOR", "(MOTOR|ID)"), col-
lapse="+") ) )
m<-lmer(fml,data=D1,control = lmerControl(optimizer ="Nelder_Mead"))
ci<-as.numeric (confint (profile (m, which = 6)))
coefs <- data.frame(coef(summary(m)))
coefs$p.z <- 2 * (1 - pnorm(abs(coefs$t.value)))
p_value <- coefs$p.z[2]

df_final$correlation_MOTOR[k]<-round(fixef(m) [2],4)
df_final$CI_lower_MOTOR[k]<-round(ci[1],4)
df_final$CI_upper_MOTOR[k]<-round(ci[2],4)
df_final$pvalue_MOTOR[k]<-round(p_value,4)

#calculate longitudinal correlation with BULBAR score
D1 <- SUMIT[, c(i,"BULBAR","ID")]
D1[,c(i,"BULBAR")]<-scale(D1[,c(i,"BULBAR")])
D1 <- as.data.frame(D1)

fml <- as.formula( paste( i, "~", paste(c("BULBAR", "(BULBAR|ID)"), col-
lapse="+") ) )
m<-lmer(fml,data=D1,control = lmerControl(optimizer ="Nelder_Mead"))
ci<-as.numeric (confint (profile (m, which = 6)))
coefs <- data.frame(coef(summary(m)))
coefs$p.z <- 2 * (1 - pnorm(abs(coefs$t.value)))
p_value <- coefs$p.z[2]

df_final$correlation_BULBAR[k]<-round(fixef(m) [2],4)
df_final$CI_lower_BULBAR[k]<-round(ci[1],4)
df_final$CI_upper_BULBAR[k]<-round(ci[2],4)
df_final$pvalue_BULBAR[k]<-round(p_value,4)

#calculate longitudinal correlation with RESPI score
D1 <- SUMIT[, c(i,"RESPI","ID")]
D1[,c(i,"RESPI")]<-scale(D1[,c(i,"RESPI")])
D1 <- as.data.frame(D1)

fml <- as.formula( paste( i, "~", paste(c("RESPI", "(RESPI|ID)"), col-
lapse="+") ) )
m<-lmer(fml,data=D1,control = lmerControl(optimizer ="Nelder_Mead"))
ci<-as.numeric (confint (profile (m, which = 6)))
coefs <- data.frame(coef(summary(m)))
coefs$p.z <- 2 * (1 - pnorm(abs(coefs$t.value)))
p_value <- coefs$p.z[2]

df_final$correlation_RESPI[k]<-round(fixef(m) [2],4)
df_final$CI_lower_RESPI[k]<-round(ci[1],4)
df_final$CI_upper_RESPI[k]<-round(ci[2],4)
df_final$pvalue_RESPI[k]<-round(p_value,4)

k<-k+1
}

```

Longitudinal correlation between AI-based markers and AC-based markers

```
library(ggplot2)
library(tidyverse)
library(read.gt3x)
library(lubridate)
library(ActivityIndex)
library(gtools)
library(data.table)
library(GGIR)
library(matrixStats)
library(lme4)
library(reshape2)

SUMIT_AI_7200sec<-readRDS(choose.files()) #choose one the 10 longitudinal
AI data frames (i.e., choose which epoch length to look at)
SUMIT_count<-readRDS(choose.files()) #read in longitudinal data frame of
count data

SUMIT_all<-merge(SUMIT_AI_7200sec,SUMIT_count,by=c("DATE","ID","VISIT"))
#merge AI and count data frame

colnames_meanAI<-vector()
colnames_P90AI<-vector()
colnames_upperAI<-vector()
for (i in 1:61){
  colnames_meanAI[i]<-paste('meanAI_',i,sep='')
  colnames_P90AI[i]<-paste('P90AI_',i,sep='')
  colnames_upperAI[i]<-paste('upperAI_',i,sep='')
}

markers<-c(colnames_meanAI,colnames_P90AI,colnames_upperAI)
df_final<-data.table(matrix(nrow=length(markers),ncol=11))
colnames(df_final)<- c("marker","correlation_AI","CI_lower_AI","CI_upper_AI",
"Pvalue_AI","CoV_AI","correlation_PA","CI_lower_PA","CI_upper_PA",
"Pvalue_PA","CoV_PA")

k<-1
# loop over all AI markers to calculate longitudinal correlations with
count-based markers
for(i in markers){

  #calculate longitudinal correlation with AI marker
  D1 <- SUMIT_all[, c(i,"AI","ID")]
  D1[,c(i,"AI")]<-scale(D1[,c(i,"AI")]) #scales columns
  D1 <- as.data.frame(D1)

  fml <- as.formula( paste( i, "~", paste(c("AI","(AI|ID)"), collapse="+")
) )
  m<-lmer(fml,data=D1,control = lmerControl(optimizer ="Nelder_Mead")) #use
mixed model
  sd_slope<-attr(VarCorr(m)$ID,"stddev")[2]
  slope<-fixef(m)[2]
  ci<-as.numeric (confint (profile (m, which = 6))) #confidence intervals

  #calculate p-values using normal approximation
  coefs <- data.frame(coef(summary(m)))
  coefs$p.z <- 2 * (1 - pnorm(abs(coefs$t.value)))
  p_value <- coefs$p.z[2]
```



```

df_final$marker[k]<-i
df_final$correlation_A1[k]<-round(fixef(m)[2],4)
df_final$CI_lower_A1[k]<-round(ci[1],4)
df_final$CI_upper_A1[k]<-round(ci[2],4)
df_final$Pvalue_A1[k]<-round(p_value,4)
df_final$CoV_A1[k]<-round(abs(sd_slope/slope),4)

#calculate longitudinal correlation with %active
D1 <- SUMIT_all[, c(i,"PA","ID")]
D1[,c(i,"PA")]<-scale(D1[,c(i,"PA")])
D1 <- as.data.frame(D1)

fml <- as.formula( paste( i, "~", paste(c("PA","(PA|ID)"), collapse="+")
) )
m<-lmer(fml,data=D1,control = lmerControl(optimizer ="Nelder_Mead"))
sd_slope<-attr(VarCorr(m)$ID,"stddev")[2]
slope<-fixef(m)[2]
ci<-as.numeric (confint (profile (m, which = 6)))
coefs <- data.frame(coef(summary(m)))
coefs$p.z <- 2 * (1 - pnorm(abs(coefs$t.value)))
p_value <- coefs$p.z[2]

# assign fit to list by name
df_final$correlation_PA[k]<-round(fixef(m)[2],4)
df_final$CI_lower_PA[k]<-round(ci[1],4)
df_final$CI_upper_PA[k]<-round(ci[2],4)
df_final$Pvalue_PA[k]<-round(p_value,4)
df_final$CoV_PA[k]<-round(abs(sd_slope/slope),4)

k<-k+1

}

```

Final selection of markers

```
library(ggplot2)
library(tidyverse)
library(read.gt3x)
library(lubridate)
library(ActivityIndex)
library(gtools)
library(data.table)
library(GGIR)
library(matrixStats)
library(lme4)
library(reshape2)

cor_10sec<-readRDS(file.choose()) # read in the files that contain the cor-
relations with the clinical markers of disease severity (10 in total for
epoch length 10-sec to 7200-sec)

#add epoch length column to dataframes
cor_5sec$epoch<-rep("5sec", times=nrow(cor_5sec))
cor_10sec$epoch<-rep("10sec", times=nrow(cor_10sec))
cor_30sec$epoch<-rep("30sec", times=nrow(cor_30sec))
cor_60sec$epoch<-rep("60sec", times=nrow(cor_60sec))
cor_120sec$epoch<-rep("120sec", times=nrow(cor_120sec))
cor_300sec$epoch<-rep("300sec", times=nrow(cor_300sec))
cor_600sec$epoch<-rep("600sec", times=nrow(cor_600sec))
cor_1800sec$epoch<-rep("1800sec", times=nrow(cor_1800sec))
cor_3600sec$epoch<-rep("3600sec", times=nrow(cor_3600sec))
cor_7200sec$epoch<-rep("7200sec", times=nrow(cor_7200sec))

#combine the data sets into one dataframe
cor_merged <- do.call("rbind", list(cor_5sec,cor_10sec, cor_30sec,
cor_60sec , cor_120sec , cor_300sec , cor_600sec , cor_1800sec,
cor_3600sec,cor_7200sec ))
time_10sec<-readRDS(file.choose()) # read in the files that contain the
longitudinal rates of decline and CoV for the clinical markers of disease
severity and AI markers (10 in total for epoch length 10-sec to 7200-sec)

# add epoch length column to the dataframes
time_5sec$epoch<-rep("5sec", times=nrow(time_5sec))
time_10sec$epoch<-rep("10sec", times=nrow(time_10sec))
time_30sec$epoch<-rep("30sec", times=nrow(time_30sec))
time_60sec$epoch<-rep("60sec", times=nrow(time_60sec))
time_120sec$epoch<-rep("120sec", times=nrow(time_120sec))
time_300sec$epoch<-rep("300sec", times=nrow(time_300sec))
time_600sec$epoch<-rep("600sec", times=nrow(time_600sec))
time_1800sec$epoch<-rep("1800sec", times=nrow(time_1800sec))
time_3600sec$epoch<-rep("3600sec", times=nrow(time_3600sec))
time_7200sec$epoch<-rep("7200sec", times=nrow(time_7200sec))

# merge dataframes into one
time_merged <- do.call("rbind", list(time_5sec,time_10sec, time_30sec,
time_60sec , time_120sec , time_300sec , time_600sec , time_1800sec ,
time_3600sec, time_7200sec))

#create a new data frame for the final selection of markers
final_selection<-data.frame(matrix(nrow=15,ncol=2))
colnames(final_selection)<-c('selection_method','marker')

# M3: Mean AI marker with highest correlation overall
```

```

final_selection[1,1]<-'overall_max_mean'
diff_epoch_meanAI<-cor_merged[ grepl( "meanAI" , cor_merged$marker ), ]

final_selection[1,2]<-diff_epoch_meanAI[which.max(diff_epoch_meanAI$correlation_ALSFRS),][,1]

# M2: Mean AI marker with highest mean correlation over all 10 epoch lengths
final_selection[2,1]<-'max_all_epochs_mean'
diff_epoch_meanAI<-cor_merged[ grepl( "meanAI" , cor_merged$marker ), ]

diff_epoch_meanAI<-aggregate(correlation_ALSFRS ~ marker, data=
diff_epoch_meanAI, FUN=mean)
final_selection[2,2]<-diff_epoch_meanAI[which.max(diff_epoch_meanAI$correlation_ALSFRS),][,1]

# M8: Upper AI marker with highest correlation overall
final_selection[3,1]<-'overall_max_upper'
diff_epoch_upperAI<-cor_merged[ grepl( "upperAI" , cor_merged$marker ), ]

final_selection[3,2]<-diff_epoch_upperAI[which.max(diff_epoch_upperAI$correlation_ALSFRS),][,1]

# M7: Upper AI marker with highest mean correlation over all 10 epoch lengths
final_selection[4,1]<-'max_all_epochs_upper'
diff_epoch_upperAI<-cor_merged[ grepl( "upperAI" , cor_merged$marker ), ]

diff_epoch_upperAI<-aggregate(correlation_ALSFRS ~ marker, data=
diff_epoch_upperAI, FUN=mean)
final_selection[4,2]<-diff_epoch_upperAI[which.max(diff_epoch_upperAI$correlation_ALSFRS),][,1]

# M12: 90P marker with highest correlation overall
final_selection[5,1]<-'overall_max_Q90'
diff_epoch_Q90AI<-cor_merged[ grepl( "P90AI" , cor_merged$marker ), ]

final_selection[5,2]<-diff_epoch_Q90AI[which.max(diff_epoch_Q90AI$correlation_ALSFRS),][,1]

# M11: 90P marker with highest mean correlation over all 10 epoch lengths
final_selection[6,1]<-'max_all_epochs_Q90'
diff_epoch_Q90AI<-cor_merged[ grepl( "P90AI" , cor_merged$marker ), ]

diff_epoch_Q90AI<-aggregate(correlation_ALSFRS ~ marker, data=
diff_epoch_Q90AI, FUN=mean)
final_selection[6,2]<-diff_epoch_Q90AI[which.max(diff_epoch_Q90AI$correlation_ALSFRS),][,1]
# M5: Mean AI marker with overall lowest CoV
final_selection[7,1]<-'overall_min_mean'
diff_epoch_meanAI_time<-time_merged[ grepl( "meanAI" , time_merged$marker ), ]

final_selection[7,2]<-
diff_epoch_meanAI_time[which.min(diff_epoch_meanAI_time$COV),][,1]

# M4: Mean AI marker with lowest mean CoV over all 10 epoch lengths
final_selection[8,1]<-'min_all_epochs_mean'
diff_epoch_meanAI_time<-time_merged[ grepl( "meanAI" , time_merged$marker ), ]

```

```

diff_epoch_meanAI_time<-aggregate(COV ~ marker, data=
diff_epoch_meanAI_time, FUN=mean)
final_selection[8,2]<-
diff_epoch_meanAI_time[which.min(diff_epoch_meanAI_time$COV),][,1]

# M9: Upper AI marker with lowest CoV overall
final_selection[9,1]<-'overall_min_upper'
diff_epoch_upperAI_time<-time_merged[ grepl( "upperAI" , time_merged$marker
), ]

final_selection[9,2]<-diff_epoch_upperAI_time[which.min(diff_epoch_up-
perAI_time$COV),][,1]

# M9: Upper AI marker with lowest mean CoV over all 10 epoch lengths
final_selection[10,1]<-'min_all_epochs_upper'
diff_epoch_upperAI_time<-time_merged[ grepl( "upperAI" , time_merged$marker
), ]

diff_epoch_upperAI_time<-aggregate(COV ~ marker, data= diff_epoch_up-
perAI_time, FUN=mean)
final_selection[10,2]<-diff_epoch_upperAI_time[which.min(diff_epoch_up-
perAI_time$COV),][,1]

# M14: 90P marker with lowest CoV overall
final_selection[11,1]<-'overall_min_Q90'
diff_epoch_Q90AI_time<-time_merged[ grepl( "P90AI" , time_merged$marker ),
]

final_selection[11,2]<-
diff_epoch_Q90AI_time[which.min(diff_epoch_Q90AI_time$COV),][,1]

# M13: 90P marker with lowest mean CoV over all 10 epoch lengths
final_selection[12,1]<-'min_all_epochs_Q90'
diff_epoch_Q90AI_time<-time_merged[ grepl( "P90AI" , time_merged$marker ),
]

diff_epoch_Q90AI_time<-aggregate(COV ~ marker, data= diff_epoch_Q90AI_time,
FUN=mean)
final_selection[12,2]<-
diff_epoch_Q90AI_time[which.min(diff_epoch_Q90AI_time$COV),][,1]

final_selection[13:15,1]<-'equal_weights'

# M1: Mean AI marker based on original AI
final_selection[13,2]<-'meanAI_61'

# M6: Upper AI marker based on original AI
final_selection[14,2]<-'upperAI_61'

#M10: 90P marker based on original AI
final_selection[15,2]<-'P90AI_61'

```

Appendix D: Supplementary Data

Raw accelerometer files for analysis

ID1 Baseline.gt3x	ID9 Baseline.gt3x	ID18 Month 8.gt3x	ID28 Month 6.gt3x
ID1 Month 2.gt3x	ID9 Month 2.gt3x	D18 Month 10.gt3x	ID28 Month 8.gt3x
ID1 Month 4.gt3x	ID9 Month 4.gt3x	D18 Month 12.gt3x	D28 Month 10.gt3x
ID1 Month 6.gt3x	ID9 Month 6.gt3x	D19 Baseline.gt3x	D28 Month 12.gt3x
ID1 Month 8.gt3x	ID9 Month 8.gt3x	ID19 Month 4.gt3x	D29 Baseline.gt3x
ID1 Month 10.gt3x	ID9 Month 10.gt3x	ID19 Month 6.gt3x	ID29 Month 2.gt3x
ID1 Month 12.gt3x	ID9 Month 12.gt3x	ID19 Month 8.gt3x	ID29 Month 4.gt3x
ID2 Baseline.gt3x	D10 Baseline.gt3x	D20 Baseline.gt3x	D30 Baseline.gt3x
ID2 Month 2.gt3x	ID10 Month 2.gt3x	ID20 Month 2.gt3x	ID30 Month 4.gt3x
ID2 Month 4.gt3x	ID10 Month 4.gt3x	ID20 Month 4.gt3x	ID30 Month 6.gt3x
ID2 Month 6.gt3x	ID10 Month 6.gt3x	ID20 Month 6.gt3x	ID30 Month 8.gt3x
ID2 Month 8.gt3x	ID10 Month 8.gt3x	ID20 Month 8.gt3x	D30 Month 10.gt3x
ID2 Month 10.gt3x	D10 Month 10.gt3x	D20 Month 10.gt3x	D30 Month 12.gt3x
ID2 Month 12.gt3x	D10 Month 12.gt3x	D20 Month 12.gt3x	D31 Baseline.gt3x
ID3 Baseline.gt3x	D11 Baseline.gt3x	D21 Baseline.gt3x	ID31 Month 2.gt3x
ID3 Month 2.gt3x	ID11 Month 2.gt3x	ID21 Month 2.gt3x	ID31 Month 6.gt3x
ID3 Month 4.gt3x	ID11 Month 4.gt3x	ID21 Month 4.gt3x	ID31 Month 8.gt3x
ID3 Month 6.gt3x	ID11 Month 6.gt3x	ID21 Month 6.gt3x	D31 Month 10.gt3x
ID3 Month 8.gt3x	ID11 Month 8.gt3x	ID21 Month 8.gt3x	D31 Month 12.gt3x
ID3 Month 10.gt3x	D12 Baseline.gt3x	D22 Baseline.gt3x	D32 Baseline.gt3x
ID3 Month 12.gt3x	ID12 Month 2.gt3x	ID22 Month 2.gt3x	D33 Baseline.gt3x
ID4 Baseline.gt3x	ID12 Month 4.gt3x	ID22 Month 4.gt3x	ID33 Month 2.gt3x
ID4 Month 2.gt3x	ID12 Month 6.gt3x	ID22 Month 8.gt3x	ID33 Month 4.gt3x
ID4 Month 4.gt3x	ID12 Month 8.gt3x	D22 Month 10.gt3x	ID33 Month 6.gt3x
ID4 Month 6.gt3x	D12 Month 10.gt3x	D22 Month 12.gt3x	ID33 Month 8.gt3x
ID4 Month 8.gt3x	D12 Month 12.gt3x	D23 Baseline.gt3x	D33 Month 10.gt3x
ID4 Month 10.gt3x	D13 Baseline.gt3x	ID23 Month 2.gt3x	D33 Month 12.gt3x
ID4 Month 12.gt3x	ID13 Month 2.gt3x	ID23 Month 4.gt3x	D34 Baseline.gt3x
ID5 Baseline.gt3x	ID13 Month 4.gt3x	ID23 Month 6.gt3x	ID34 Month 2.gt3x
ID5 Month 2.gt3x	D14 Baseline.gt3x	ID23 Month 8.gt3x	ID34 Month 4.gt3x
ID5 Month 4.gt3x	ID14 Month 2.gt3x	D23 Month 10.gt3x	D35 Baseline.gt3x
ID5 Month 6.gt3x	ID14 Month 6.gt3x	D24 Baseline.gt3x	ID35 Month 2.gt3x
ID5 Month 8.gt3x	ID14 Month 8.gt3x	D25 Baseline.gt3x	D36 Baseline.gt3x
ID5 Month 10.gt3x	D15 Baseline.gt3x	ID25 Month 2.gt3x	ID36 Month 2.gt3x
ID5 Month 12.gt3x	ID15 Month 2.gt3x	ID25 Month 4.gt3x	ID36 Month 6.gt3x
ID6 Baseline.gt3x	ID15 Month 4.gt3x	D26 Baseline.gt3x	ID36 Month 8.gt3x
ID6 Month 4.gt3x	ID15 Month 6.gt3x	ID26 Month 2.gt3x	D37 Baseline.gt3x
ID6 Month 6.gt3x	ID15 Month 8.gt3x	ID26 Month 4.gt3x	ID37 Month 2.gt3x
ID6 Month 8.gt3x	D15 Month 10.gt3x	ID26 Month 6.gt3x	ID37 Month 4.gt3x
ID6 Month 10.gt3x	D16 Baseline.gt3x	ID26 Month 8.gt3x	D38 Baseline.gt3x
ID6 Month 12.gt3x	ID16 Month 2.gt3x	D26 Month 10.gt3x	D39 Baseline.gt3x
ID7 Baseline.gt3x	D17 Baseline.gt3x	D26 Month 12.gt3x	ID39 Month 2.gt3x
ID7 Month 2.gt3x	ID17 Month 2.gt3x	D27 Baseline.gt3x	D40 Baseline.gt3x
ID7 Month 4.gt3x	ID17 Month 4.gt3x	ID27 Month 2.gt3x	ID40 Month 2.gt3x
ID7 Month 6.gt3x	ID17 Month 6.gt3x	ID27 Month 4.gt3x	ID40 Month 4.gt3x
ID7 Month 8.gt3x	ID17 Month 8.gt3x	ID27 Month 6.gt3x	ID40 Month 6.gt3x
ID7 Month 10.gt3x	D17 Month 10.gt3x	ID27 Month 8.gt3x	D41 Baseline.gt3x
ID7 Month 12.gt3x	D17 Month 12.gt3x	D27 Month 10.gt3x	ID41 Month 2.gt3x
ID8 Baseline.gt3x	D18 Baseline.gt3x	D27 Month 12.gt3x	ID41 Month 4.gt3x
ID8 Month 2.gt3x	ID18 Month 2.gt3x	D28 Baseline.gt3x	D42 Baseline.gt3x
ID8 Month 4.gt3x	ID18 Month 4.gt3x	ID28 Month 2.gt3x	ID42 Month 2.gt3x
ID8 Month 6.gt3x	ID18 Month 6.gt3x	ID28 Month 4.gt3x	

Overview over systematic noise

Serial number	Systematic standard deviation
TAS1E44150308	0.003120528
TAS1E44150310	NA
TAS1E44150315	0.003121133
TAS1E44150318	0.002975902
TAS1E44150319	0.003132822
TAS1E44150320	NA
TAS1E44150325	0.002844982
TAS1E44150326	0.003097446
TAS1E44150327	0.003089388
TAS1E44150328	0.003195838
Mean (SD):	0.003072387 (0.0001105347)

AI Contribution Weights

The table shows the 61 different axes weights used in the modified AI metrics.

Index	X-Axis	Y-Axis	Z-Axis
1	1	0	0
2	0,9	0,1	0
3	0,8	0,2	0
4	0,7	0,3	0
5	0,6	0,4	0
6	0,5	0,5	0
7	0,4	0,6	0
8	0,3	0,7	0
9	0,2	0,8	0
10	0,1	0,9	0
11	0	1	0
12	0,9	0	0,1
13	0,8	0,1	0,1
14	0,7	0,2	0,1
15	0,5	0,4	0,1
16	0,4	0,5	0,1
17	0,2	0,7	0,1
18	0,1	0,8	0,1
19	0	0,9	0,1
20	0,8	0	0,2
21	0,7	0,1	0,2
22	0,6	0,2	0,2
23	0,5	0,3	0,2
24	0,4	0,4	0,2
25	0,3	0,5	0,2
26	0,2	0,6	0,2
27	0,1	0,7	0,2
28	0	0,8	0,2
29	0,7	0	0,3
30	0,5	0,2	0,3
31	0,4	0,3	0,3
32	0,3	0,4	0,3
33	0,2	0,5	0,3
34	0	0,7	0,3
35	0,6	0	0,4
36	0,5	0,1	0,4
37	0,4	0,2	0,4
38	0,3	0,3	0,4
39	0,2	0,4	0,4
40	0,1	0,5	0,4
41	0	0,6	0,4
42	0,5	0	0,5
43	0,4	0,1	0,5
44	0,3	0,2	0,5
45	0,2	0,3	0,5
46	0,1	0,4	0,5
47	0	0,5	0,5
48	0,4	0	0,6
49	0,2	0,2	0,6
50	0	0,4	0,6
51	0,3	0	0,7
52	0,2	0,1	0,7
53	0,1	0,2	0,7

54	0	0,3	0,7
55	0,2	0	0,8
56	0,1	0,1	0,8
57	0	0,2	0,8
58	0,1	0	0,9
59	0	0,1	0,9
60	0	0	1
61	0,333333	0,333333	0,333333

Full list of longitudinal rates of decline, baseline values and CoV

The table shows the longitudinal rates of decline, the baseline values, and the coefficient of variation of the final markers.

ID	Epoch length	intercept	slope	CI lower	CI upper	p-value	CoV
M1	10sec	0,173	-0,004	-0,005	-0,003	0	0,827
M1	120sec	1,144	-0,021	-0,028	-0,014	0	0,844
M1	1800sec	3,471	-0,037	-0,051	-0,024	0	0,94
M1	300sec	1,834	-0,029	-0,039	-0,019	0	0,876
M1	30sec	0,441	-0,01	-0,013	-0,006	0	0,823
M1	3600sec	4,15	-0,038	-0,052	-0,024	0	0,974
M1	5sec	0,091	-0,002	-0,003	-0,001	0	0,829
M1	600sec	2,438	-0,033	-0,045	-0,022	0	0,905
M1	60sec	0,737	-0,015	-0,02	-0,01	0	0,828
M1	7200sec	4,84	-0,04	-0,054	-0,026	0	0,956
M10	10sec	0,445	-0,01	-0,014	-0,007	0	0,833
M10	120sec	1,923	-0,029	-0,04	-0,018	0	0,987
M10	1800sec	4,195	-0,036	-0,05	-0,022	0	0,98
M10	300sec	2,68	-0,032	-0,045	-0,019	0	1,091
M10	30sec	0,951	-0,019	-0,025	-0,012	0	0,844
M10	3600sec	4,739	-0,035	-0,049	-0,021	0	1,036
M10	5sec	0,254	-0,006	-0,008	-0,004	0	0,859
M10	600sec	3,264	-0,032	-0,046	-0,019	0	1,048
M10	60sec	1,402	-0,025	-0,034	-0,016	0	0,901
M10	7200sec	5,309	-0,039	-0,055	-0,024	0	1,01
M11	10sec	0,45	-0,01	-0,014	-0,007	0	0,89
M11	120sec	1,933	-0,03	-0,043	-0,018	0	1,072
M11	1800sec	4,194	-0,037	-0,052	-0,023	0	1,023
M11	300sec	2,686	-0,033	-0,047	-0,019	0	1,156
M11	30sec	0,958	-0,019	-0,026	-0,013	0	0,9
M11	3600sec	4,735	-0,036	-0,05	-0,021	0	1,07
M11	5sec	0,257	-0,006	-0,009	-0,004	0	0,92
M11	600sec	3,268	-0,033	-0,048	-0,019	0	1,119
M11	60sec	1,41	-0,025	-0,035	-0,016	0	0,978
M11	7200sec	5,307	-0,04	-0,056	-0,024	0	1,045
M12	10sec	0,448	-0,01	-0,014	-0,007	0	0,915
M12	120sec	1,928	-0,03	-0,043	-0,018	0	1,113
M12	1800sec	4,184	-0,037	-0,052	-0,022	0	1,047
M12	300sec	2,678	-0,032	-0,047	-0,018	0	1,183
M12	30sec	0,955	-0,019	-0,026	-0,012	0	0,931
M12	3600sec	4,725	-0,035	-0,05	-0,021	0	1,088
M12	5sec	0,256	-0,006	-0,009	-0,004	0	0,944
M12	600sec	3,259	-0,033	-0,048	-0,019	0	1,169
M12	60sec	1,405	-0,025	-0,035	-0,015	0	1,016
M12	7200sec	5,297	-0,04	-0,056	-0,024	0	1,064
M13	10sec	0,429	-0,011	-0,014	-0,007	0	0,841
M13	120sec	1,889	-0,031	-0,042	-0,019	0	0,954
M13	1800sec	4,172	-0,038	-0,052	-0,024	0	0,952
M13	300sec	2,646	-0,034	-0,048	-0,021	0	1,028
M13	30sec	0,925	-0,02	-0,026	-0,013	0	0,81
M13	3600sec	4,72	-0,038	-0,053	-0,023	0	1,001
M13	5sec	0,243	-0,006	-0,009	-0,004	0	0,882
M13	600sec	3,235	-0,035	-0,049	-0,022	0	0,989
M13	60sec	1,372	-0,026	-0,035	-0,017	0	0,869
M13	7200sec	5,288	-0,041	-0,057	-0,025	0	0,986
M14	10sec	0,418	-0,009	-0,012	-0,006	0	0,81
M14	120sec	1,85	-0,027	-0,037	-0,017	0	0,954
M14	1800sec	4,1	-0,033	-0,046	-0,02	0	0,999
M14	300sec	2,598	-0,03	-0,042	-0,018	0	1,036
M14	30sec	0,901	-0,017	-0,023	-0,011	0	0,844
M14	3600sec	4,641	-0,032	-0,046	-0,019	0	1,05
M14	5sec	0,237	-0,006	-0,007	-0,004	0	0,803
M14	600sec	3,176	-0,03	-0,042	-0,018	0	1,051
M14	60sec	1,339	-0,022	-0,031	-0,014	0	0,909
M14	7200sec	5,211	-0,037	-0,052	-0,023	0	1,013
M2	10sec	0,175	-0,004	-0,006	-0,003	0	0,849
M2	120sec	1,152	-0,022	-0,03	-0,015	0	0,849
M2	1800sec	3,481	-0,039	-0,053	-0,025	0	0,943

M2	300sec	1,842	-0,03	-0,04	-0,02	0	0,879
M2	30sec	0,446	-0,01	-0,014	-0,007	0	0,838
M2	3600sec	4,159	-0,04	-0,054	-0,025	0	0,973
M2	5sec	0,092	-0,002	-0,003	-0,002	0	0,856
M2	600sec	2,447	-0,034	-0,047	-0,023	0	0,91
M2	60sec	0,743	-0,016	-0,021	-0,011	0	0,838
M2	7200sec	4,849	-0,042	-0,057	-0,027	0	0,958
M3	10sec	0,175	-0,004	-0,006	-0,003	0	0,844
M3	120sec	1,149	-0,022	-0,03	-0,015	0	0,84
M3	1800sec	3,476	-0,039	-0,053	-0,025	0	0,924
M3	300sec	1,838	-0,03	-0,04	-0,02	0	0,866
M3	30sec	0,445	-0,01	-0,014	-0,007	0	0,832
M3	3600sec	4,154	-0,04	-0,054	-0,025	0	0,954
M3	5sec	0,092	-0,002	-0,003	-0,002	0	0,85
M3	600sec	2,442	-0,034	-0,047	-0,023	0	0,893
M3	60sec	0,741	-0,016	-0,021	-0,011	0	0,83
M3	7200sec	4,843	-0,042	-0,057	-0,027	0	0,938
M4	10sec	0,172	-0,004	-0,006	-0,003	0	0,846
M4	120sec	1,133	-0,022	-0,029	-0,015	0	0,836
M4	1800sec	3,45	-0,039	-0,053	-0,025	0	0,911
M4	300sec	1,817	-0,03	-0,04	-0,02	0	0,859
M4	30sec	0,437	-0,01	-0,014	-0,007	0	0,833
M4	3600sec	4,128	-0,04	-0,054	-0,026	0	0,941
M4	5sec	0,09	-0,002	-0,003	-0,002	0	0,852
M4	600sec	2,419	-0,034	-0,046	-0,023	0	0,882
M4	60sec	0,73	-0,016	-0,021	-0,011	0	0,83
M4	7200sec	4,817	-0,042	-0,057	-0,027	0	0,922
M5	10sec	0,157	-0,003	-0,005	-0,002	0	0,797
M5	120sec	1,078	-0,019	-0,026	-0,013	0	0,843
M5	1800sec	3,379	-0,034	-0,048	-0,022	0	0,961
M5	300sec	1,753	-0,026	-0,036	-0,017	0	0,884
M5	30sec	0,406	-0,008	-0,011	-0,006	0	0,805
M5	3600sec	4,056	-0,035	-0,049	-0,022	0	1
M5	5sec	0,082	-0,002	-0,002	-0,001	0	0,796
M5	600sec	2,351	-0,03	-0,042	-0,02	0	0,915
M5	60sec	0,686	-0,013	-0,018	-0,009	0	0,819
M5	7200sec	4,747	-0,037	-0,051	-0,023	0	0,978
M6	10sec	0,472	-0,01	-0,013	-0,006	0	0,865
M6	120sec	1,968	-0,028	-0,038	-0,018	0	0,928
M6	1800sec	4,226	-0,035	-0,048	-0,022	0	0,996
M6	300sec	2,725	-0,032	-0,043	-0,02	0	0,965
M6	30sec	0,991	-0,018	-0,024	-0,012	0	0,87
M6	3600sec	4,782	-0,036	-0,05	-0,022	0	1,025
M6	5sec	0,272	-0,006	-0,008	-0,004	0	0,874
M6	600sec	3,312	-0,033	-0,046	-0,021	0	0,987
M6	60sec	1,442	-0,023	-0,031	-0,015	0	0,892
M6	7200sec	5,324	-0,039	-0,054	-0,024	0	1,012
M7	10sec	0,478	-0,01	-0,014	-0,007	0	0,89
M7	120sec	1,978	-0,028	-0,039	-0,018	0	0,953
M7	1800sec	4,235	-0,036	-0,05	-0,022	0	1,016
M7	300sec	2,735	-0,032	-0,045	-0,02	0	0,988
M7	30sec	0,999	-0,018	-0,025	-0,012	0	0,892
M7	3600sec	4,79	-0,037	-0,052	-0,022	0	1,038
M7	5sec	0,276	-0,006	-0,008	-0,004	0	0,901
M7	600sec	3,322	-0,034	-0,047	-0,021	0	1,009
M7	60sec	1,452	-0,024	-0,032	-0,016	0	0,916
M7	7200sec	5,331	-0,04	-0,056	-0,024	0	1,025
M8	10sec	0,479	-0,01	-0,014	-0,007	0	0,886
M8	120sec	1,98	-0,029	-0,039	-0,018	0	0,943
M8	1800sec	4,237	-0,036	-0,05	-0,022	0	1,007
M8	300sec	2,736	-0,033	-0,045	-0,021	0	0,977
M8	30sec	1	-0,019	-0,025	-0,012	0	0,885
M8	3600sec	4,792	-0,037	-0,052	-0,022	0	1,029
M8	5sec	0,277	-0,006	-0,008	-0,004	0	0,897
M8	600sec	3,324	-0,034	-0,048	-0,021	0	0,998
M8	60sec	1,454	-0,024	-0,033	-0,016	0	0,907
M8	7200sec	5,332	-0,04	-0,056	-0,024	0	1,015
M9	10sec	0,434	-0,008	-0,011	-0,005	0	0,829
M9	120sec	1,879	-0,025	-0,034	-0,016	0	0,913
M9	1800sec	4,123	-0,032	-0,044	-0,02	0	0,994
M9	300sec	2,628	-0,029	-0,04	-0,018	0	0,956

M9	30sec	0,925	-0,016	-0,021	-0,011	0	0,846
M9	3600sec	4,68	-0,033	-0,046	-0,02	0	1,027
M9	5sec	0,248	-0,005	-0,007	-0,003	0	0,831
M9	600sec	3,212	-0,03	-0,042	-0,019	0	0,979
M9	60sec	1,364	-0,021	-0,028	-0,014	0	0,875
M9	7200sec	5,226	-0,036	-0,051	-0,022	0	1,015

Full list of longitudinal correlations of final markers with clinical markers of disease severity

The table shows the longitudinal correlations of the final markers with the ALSFRS-R, MOTOR, BULBAR and RESPI scores. The p-values for the longitudinal correlations of the final markers with the ALSFRS-R total score and BULBAR were not included in the table, since they were smaller than 0.001 for all markers.

ID	Epoch length	COR ALSFRS	95% CI Low	95% CI Up	COR MOTOR	95% CI Low	95% CI Up	COR BULBAR	95% CI Low	95% CI Up	P-value	COR RESPI	95% CI Low	95% CI Up	P-value
M1	5sec	0,593	0,434	0,752	0,686	0,519	0,855	0,489	0,209	0,765	0	0,371	0,081	0,667	0,01
M1	10sec	0,604	0,444	0,762	0,697	0,529	0,867	0,497	0,212	0,777	0	0,379	0,086	0,68	0,01
M1	30sec	0,63	0,47	0,788	0,724	0,555	0,893	0,514	0,218	0,806	0	0,404	0,101	0,715	0,008
M1	60sec	0,65	0,49	0,808	0,742	0,574	0,911	0,53	0,222	0,833	0	0,424	0,113	0,745	0,006
M1	120sec	0,668	0,507	0,825	0,755	0,588	0,923	0,547	0,226	0,863	0	0,447	0,126	0,777	0,005
M1	300sec	0,684	0,522	0,841	0,763	0,595	0,93	0,569	0,23	0,903	0	0,474	0,141	0,817	0,004
M1	600sec	0,69	0,527	0,847	0,762	0,594	0,93	0,581	0,229	0,928	0,001	0,488	0,148	0,839	0,004
M1	1800sec	0,695	0,529	0,854	0,76	0,591	0,928	0,597	0,229	0,96	0,001	0,504	0,154	0,865	0,004
M1	3600sec	0,695	0,528	0,853	0,756	0,586	0,925	0,609	0,235	0,979	0,001	0,522	0,167	0,889	0,003
M1	7200sec	0,709	0,54	0,871	0,768	0,594	0,942	0,619	0,242	0,992	0,001	0,527	0,163	0,903	0,004
M10	5sec	0,618	0,463	0,78	0,718	0,568	0,874	0,468	0,196	0,738	0	0,336	0,013	0,672	0,038
M10	10sec	0,638	0,487	0,796	0,739	0,596	0,886	0,482	0,202	0,758	0	0,345	0,015	0,689	0,038
M10	30sec	0,666	0,519	0,817	0,768	0,633	0,907	0,508	0,21	0,801	0	0,369	0,025	0,731	0,034
M10	60sec	0,683	0,534	0,833	0,771	0,634	0,91	0,541	0,225	0,854	0	0,398	0,046	0,77	0,026
M10	120sec	0,69	0,536	0,842	0,76	0,612	0,91	0,566	0,232	0,899	0	0,42	0,058	0,803	0,022
M10	300sec	0,702	0,543	0,857	0,753	0,604	0,9	0,591	0,244	0,939	0	0,475	0,105	0,867	0,011
M10	600sec	0,699	0,546	0,85	0,744	0,599	0,888	0,588	0,253	0,926	0	0,448	0,101	0,818	0,011
M10	1800sec	0,699	0,538	0,863	0,748	0,595	0,896	0,584	0,251	0,918	0	0,475	0,13	0,844	0,007
M10	3600sec	0,685	0,532	0,836	0,726	0,578	0,87	0,615	0,261	0,976	0	0,499	0,16	0,857	0,004
M10	7200sec	0,696	0,532	0,855	0,726	0,571	0,879	0,584	0,226	0,942	0,001	0,569	0,21	0,95	0,002
M11	5sec	0,645	0,486	0,813	0,747	0,586	0,911	0,509	0,242	0,776	0	0,349	0,017	0,694	0,037
M11	10sec	0,66	0,505	0,823	0,764	0,613	0,919	0,521	0,239	0,801	0	0,357	0,019	0,71	0,036
M11	30sec	0,681	0,53	0,835	0,787	0,644	0,933	0,539	0,235	0,84	0	0,376	0,026	0,743	0,034
M11	60sec	0,688	0,532	0,845	0,787	0,638	0,937	0,55	0,222	0,873	0,001	0,404	0,048	0,779	0,025
M11	120sec	0,701	0,543	0,86	0,781	0,623	0,942	0,59	0,252	0,928	0	0,431	0,068	0,815	0,02
M11	300sec	0,707	0,545	0,866	0,767	0,609	0,922	0,606	0,26	0,951	0	0,474	0,108	0,862	0,011
M11	600sec	0,713	0,556	0,868	0,768	0,615	0,921	0,608	0,28	0,938	0	0,468	0,118	0,843	0,009
M11	1800sec	0,718	0,551	0,887	0,773	0,616	0,926	0,62	0,29	0,951	0	0,488	0,144	0,856	0,005
M11	3600sec	0,698	0,543	0,85	0,748	0,6	0,893	0,628	0,284	0,976	0	0,513	0,165	0,885	0,004
M11	7200sec	0,705	0,544	0,862	0,738	0,584	0,889	0,596	0,251	0,938	0	0,578	0,212	0,968	0,002
M12	5sec	0,647	0,483	0,819	0,749	0,585	0,917	0,517	0,244	0,788	0	0,351	0,018	0,699	0,037
M12	10sec	0,663	0,502	0,83	0,767	0,611	0,927	0,53	0,241	0,815	0	0,36	0,02	0,715	0,036
M12	30sec	0,678	0,522	0,838	0,787	0,637	0,939	0,546	0,235	0,853	0	0,379	0,029	0,748	0,033
M12	60sec	0,686	0,525	0,848	0,786	0,631	0,943	0,554	0,217	0,888	0,001	0,407	0,051	0,783	0,024
M12	120sec	0,696	0,532	0,86	0,78	0,615	0,947	0,594	0,247	0,94	0	0,433	0,069	0,818	0,019
M12	300sec	0,703	0,538	0,865	0,765	0,604	0,924	0,608	0,259	0,956	0	0,475	0,111	0,861	0,01
M12	600sec	0,712	0,551	0,871	0,769	0,611	0,927	0,613	0,282	0,946	0	0,472	0,123	0,847	0,008
M12	1800sec	0,719	0,548	0,892	0,777	0,616	0,934	0,629	0,294	0,965	0	0,495	0,149	0,864	0,005
M12	3600sec	0,695	0,538	0,85	0,749	0,599	0,896	0,626	0,283	0,973	0	0,515	0,165	0,888	0,004
M12	7200sec	0,702	0,54	0,862	0,737	0,582	0,889	0,595	0,25	0,936	0	0,581	0,212	0,976	0,002
M13	5sec	0,589	0,436	0,75	0,675	0,53	0,825	0,438	0,205	0,671	0	0,322	0,014	0,644	0,037
M13	10sec	0,618	0,468	0,776	0,708	0,57	0,851	0,458	0,209	0,703	0	0,336	0,017	0,672	0,036
M13	30sec	0,66	0,52	0,805	0,759	0,636	0,886	0,484	0,21	0,75	0	0,364	0,024	0,724	0,034
M13	60sec	0,689	0,547	0,832	0,778	0,652	0,906	0,525	0,23	0,816	0	0,4	0,048	0,773	0,025
M13	120sec	0,697	0,551	0,843	0,768	0,632	0,904	0,557	0,24	0,872	0	0,426	0,065	0,809	0,02
M13	300sec	0,712	0,558	0,863	0,765	0,623	0,905	0,59	0,246	0,933	0	0,475	0,099	0,876	0,013
M13	600sec	0,705	0,553	0,857	0,75	0,607	0,893	0,584	0,259	0,91	0	0,46	0,106	0,841	0,011
M13	1800sec	0,699	0,54	0,859	0,75	0,6	0,895	0,558	0,24	0,876	0	0,479	0,127	0,861	0,008
M13	3600sec	0,686	0,531	0,839	0,731	0,581	0,879	0,594	0,26	0,933	0	0,504	0,161	0,871	0,004
M13	7200sec	0,695	0,53	0,856	0,734	0,576	0,89	0,583	0,237	0,927	0	0,565	0,204	0,951	0,002
M14	5sec	0,6	0,439	0,765	0,691	0,544	0,846	0,456	0,149	0,761	0,002	0,336	0,026	0,654	0,029
M14	10sec	0,622	0,461	0,786	0,712	0,565	0,866	0,475	0,16	0,787	0,002	0,351	0,032	0,68	0,027
M14	30sec	0,653	0,491	0,815	0,738	0,593	0,888	0,51	0,186	0,833	0,001	0,376	0,043	0,722	0,024
M14	60sec	0,665	0,5	0,828	0,739	0,589	0,892	0,539	0,205	0,872	0,001	0,406	0,063	0,764	0,019
M14	120sec	0,669	0,5	0,835	0,724	0,565	0,886	0,559	0,209	0,907	0,001	0,424	0,074	0,789	0,016
M14	300sec	0,679	0,509	0,844	0,716	0,562	0,867	0,59	0,229	0,949	0,001	0,477	0,122	0,848	0,008

M14	600sec	0,673	0,509	0,835	0,709	0,556	0,859	0,606	0,243	0,973	0,001	0,464	0,131	0,815	0,006
M14	1800sec	0,673	0,502	0,844	0,71	0,556	0,857	0,58	0,209	0,953	0,001	0,468	0,13	0,824	0,006
M14	3600sec	0,651	0,487	0,81	0,688	0,536	0,833	0,611	0,223	1,007	0,001	0,495	0,161	0,842	0,003
M14	7200sec	0,665	0,49	0,835	0,692	0,529	0,851	0,577	0,189	0,966	0,002	0,559	0,211	0,922	0,001
M2	5sec	0,613	0,458	0,768	0,704	0,542	0,867	0,505	0,236	0,77	0	0,381	0,081	0,69	0,011
M2	10sec	0,623	0,468	0,779	0,715	0,553	0,879	0,513	0,238	0,783	0	0,389	0,086	0,702	0,01
M2	30sec	0,649	0,494	0,804	0,742	0,58	0,905	0,531	0,241	0,814	0	0,414	0,102	0,737	0,008
M2	60sec	0,669	0,514	0,823	0,76	0,598	0,922	0,546	0,244	0,841	0	0,435	0,116	0,767	0,007
M2	120sec	0,688	0,532	0,84	0,774	0,612	0,936	0,561	0,245	0,871	0	0,458	0,13	0,8	0,005
M2	300sec	0,704	0,547	0,856	0,782	0,619	0,944	0,58	0,246	0,908	0	0,487	0,147	0,84	0,004
M2	600sec	0,71	0,551	0,863	0,783	0,618	0,946	0,592	0,245	0,933	0	0,502	0,156	0,862	0,004
M2	1800sec	0,715	0,554	0,87	0,781	0,614	0,946	0,607	0,244	0,963	0,001	0,519	0,165	0,888	0,004
M2	3600sec	0,715	0,553	0,868	0,778	0,61	0,944	0,619	0,251	0,98	0,001	0,536	0,177	0,909	0,003
M2	7200sec	0,727	0,562	0,884	0,789	0,616	0,96	0,627	0,258	0,99	0	0,54	0,173	0,922	0,004
M3	5sec	0,61	0,454	0,766	0,702	0,538	0,867	0,502	0,232	0,767	0	0,379	0,083	0,684	0,011
M3	10sec	0,62	0,464	0,776	0,713	0,549	0,879	0,509	0,234	0,779	0	0,388	0,088	0,697	0,01
M3	30sec	0,646	0,491	0,801	0,74	0,577	0,904	0,528	0,24	0,81	0	0,413	0,103	0,733	0,008
M3	60sec	0,667	0,512	0,82	0,759	0,596	0,922	0,543	0,243	0,838	0	0,434	0,116	0,764	0,007
M3	120sec	0,685	0,531	0,837	0,773	0,611	0,935	0,56	0,246	0,868	0	0,457	0,13	0,798	0,005
M3	300sec	0,702	0,547	0,853	0,782	0,619	0,943	0,58	0,248	0,906	0	0,486	0,147	0,839	0,004
M3	600sec	0,709	0,552	0,861	0,782	0,618	0,944	0,593	0,248	0,932	0	0,501	0,155	0,861	0,004
M3	1800sec	0,715	0,555	0,868	0,781	0,615	0,944	0,608	0,248	0,963	0	0,518	0,163	0,887	0,004
M3	3600sec	0,715	0,555	0,868	0,778	0,611	0,942	0,621	0,255	0,981	0	0,535	0,175	0,909	0,003
M3	7200sec	0,728	0,564	0,883	0,789	0,617	0,958	0,629	0,262	0,991	0	0,538	0,17	0,922	0,004
M4	5sec	0,602	0,444	0,76	0,695	0,528	0,865	0,493	0,225	0,756	0	0,374	0,084	0,672	0,01
M4	10sec	0,612	0,454	0,77	0,707	0,539	0,877	0,5	0,228	0,769	0	0,383	0,088	0,686	0,009
M4	30sec	0,638	0,482	0,794	0,734	0,567	0,902	0,52	0,234	0,8	0	0,409	0,103	0,725	0,008
M4	60sec	0,659	0,503	0,813	0,753	0,588	0,92	0,537	0,24	0,828	0	0,431	0,115	0,758	0,006
M4	120sec	0,679	0,523	0,831	0,768	0,605	0,932	0,555	0,245	0,859	0	0,455	0,128	0,794	0,005
M4	300sec	0,697	0,541	0,848	0,778	0,615	0,94	0,578	0,249	0,9	0	0,484	0,145	0,838	0,005
M4	600sec	0,704	0,547	0,856	0,779	0,616	0,941	0,592	0,25	0,929	0	0,5	0,153	0,861	0,004
M4	1800sec	0,711	0,551	0,864	0,778	0,613	0,942	0,608	0,25	0,961	0	0,517	0,16	0,889	0,004
M4	3600sec	0,712	0,551	0,865	0,776	0,609	0,939	0,623	0,258	0,981	0	0,535	0,172	0,912	0,003
M4	7200sec	0,726	0,562	0,882	0,787	0,617	0,956	0,632	0,265	0,993	0	0,539	0,167	0,927	0,004
M5	5sec	0,538	0,363	0,708	0,622	0,444	0,801	0,456	0,156	0,753	0,001	0,346	0,079	0,611	0,008
M5	10sec	0,549	0,372	0,72	0,634	0,454	0,813	0,464	0,159	0,764	0,001	0,353	0,083	0,624	0,007
M5	30sec	0,576	0,398	0,748	0,662	0,482	0,842	0,482	0,169	0,792	0,001	0,378	0,096	0,66	0,006
M5	60sec	0,598	0,419	0,771	0,682	0,502	0,863	0,499	0,177	0,818	0,001	0,398	0,107	0,69	0,005
M5	120sec	0,618	0,438	0,792	0,699	0,519	0,88	0,519	0,185	0,85	0,001	0,419	0,118	0,723	0,005
M5	300sec	0,636	0,454	0,81	0,71	0,53	0,89	0,545	0,193	0,893	0,001	0,447	0,131	0,766	0,004
M5	600sec	0,642	0,46	0,816	0,712	0,532	0,892	0,557	0,193	0,918	0,002	0,46	0,136	0,789	0,004
M5	1800sec	0,647	0,464	0,823	0,71	0,528	0,889	0,578	0,194	0,957	0,002	0,478	0,141	0,821	0,004
M5	3600sec	0,646	0,459	0,823	0,706	0,523	0,887	0,589	0,196	0,978	0,002	0,5	0,153	0,851	0,004
M5	7200sec	0,665	0,478	0,843	0,719	0,533	0,905	0,601	0,203	0,995	0,002	0,506	0,151	0,867	0,004
M6	5sec	0,653	0,497	0,812	0,749	0,585	0,916	0,52	0,228	0,806	0	0,396	0,074	0,732	0,015
M6	10sec	0,669	0,514	0,827	0,764	0,603	0,929	0,531	0,231	0,824	0	0,412	0,083	0,757	0,013
M6	30sec	0,693	0,54	0,847	0,783	0,627	0,942	0,549	0,235	0,856	0	0,446	0,104	0,806	0,01
M6	60sec	0,704	0,551	0,856	0,788	0,634	0,945	0,564	0,239	0,882	0	0,466	0,116	0,837	0,009
M6	120sec	0,711	0,556	0,865	0,788	0,632	0,945	0,577	0,242	0,905	0	0,487	0,129	0,867	0,007
M6	300sec	0,714	0,556	0,869	0,778	0,621	0,934	0,594	0,254	0,929	0	0,512	0,146	0,9	0,006
M6	600sec	0,711	0,552	0,866	0,766	0,609	0,922	0,599	0,258	0,936	0	0,523	0,159	0,909	0,005
M6	1800sec	0,698	0,541	0,852	0,744	0,588	0,897	0,598	0,263	0,932	0	0,529	0,174	0,907	0,003
M6	3600sec	0,685	0,53	0,834	0,726	0,577	0,87	0,603	0,262	0,944	0	0,551	0,197	0,926	0,002
M6	7200sec	0,683	0,521	0,84	0,716	0,562	0,869	0,565	0,211	0,918	0,001	0,57	0,213	0,95	0,002
M7	5sec	0,66	0,507	0,818	0,755	0,595	0,918	0,53	0,25	0,802	0	0,399	0,073	0,741	0,016
M7	10sec	0,675	0,523	0,832	0,769	0,612	0,929	0,539	0,251	0,821	0	0,414	0,082	0,764	0,014
M7	30sec	0,697	0,546	0,85	0,789	0,636	0,944	0,555	0,25	0,853	0	0,446	0,101	0,81	0,011
M7	60sec	0,707	0,555	0,858	0,795	0,642	0,948	0,567	0,251	0,876	0	0,466	0,114	0,84	0,009
M7	120sec	0,714	0,56	0,867	0,796	0,641	0,95	0,578	0,252	0,897	0	0,487	0,127	0,87	0,008
M7	300sec	0,718	0,56	0,873	0,787	0,631	0,942	0,595	0,265	0,919	0	0,513	0,146	0,904	0,006
M7	600sec	0,716	0,558	0,871	0,776	0,62	0,931	0,601	0,272	0,925	0	0,524	0,159	0,914	0,005
M7	1800sec	0,706	0,55	0,86	0,756	0,601	0,907	0,604	0,282	0,924	0	0,536	0,176	0,922	0,003
M7	3600sec	0,695	0,542	0,842	0,74	0,593	0,882	0,608	0,281	0,933	0	0,556	0,196	0,94	0,002
M7	7200sec	0,693	0,535	0,848	0,73	0,578	0,879	0,578	0,237	0,917	0	0,574	0,214	0,96	0,002
M8	5sec	0,659	0,507	0,817	0,755	0,593	0,919	0,528	0,249	0,8	0	0,398	0,074	0,738	0,015
M8	10sec	0,674	0,522	0,83	0,769	0,611	0,93	0,538	0,25	0,818	0	0,413	0,082	0,762	0,014
M8	30sec	0,697	0,547	0,849	0,789	0,636	0,944	0,554	0,25	0,851	0	0,445	0,102	0,809	0,011
M8	60sec	0,707	0,557	0,858	0,795	0,643	0,948	0,567	0,252	0,874	0	0,466	0,114	0,839	0,009
M8	120sec	0,715	0,561	0,867	0,796	0,642	0,95	0,578	0,253	0,896	0	0,487	0,128	0,869	0,008
M8	300sec	0,718	0,561	0,872	0,787	0,631	0,941	0,595	0,265	0,918	0	0,512	0,146	0,902	0,006
M8	600sec	0,716	0,559	0,87	0,776	0,62	0,93	0,601	0,272	0,925	0	0,523	0,159	0,912	0,005

M8	1800sec	0,706	0,549	0,859	0,755	0,6	0,906	0,602	0,28	0,922	0	0,535	0,175	0,919	0,003
M8	3600sec	0,693	0,541	0,841	0,739	0,592	0,881	0,606	0,279	0,931	0	0,554	0,195	0,936	0,002
M8	7200sec	0,692	0,533	0,847	0,729	0,577	0,878	0,576	0,235	0,915	0	0,572	0,213	0,956	0,002
M9	5sec	0,615	0,447	0,782	0,695	0,532	0,863	0,497	0,174	0,814	0,001	0,376	0,07	0,691	0,013
M9	10sec	0,634	0,468	0,801	0,714	0,552	0,88	0,511	0,183	0,833	0,001	0,394	0,081	0,718	0,012
M9	30sec	0,664	0,5	0,827	0,739	0,581	0,901	0,536	0,199	0,867	0,001	0,433	0,105	0,773	0,008
M9	60sec	0,679	0,514	0,84	0,747	0,59	0,907	0,556	0,21	0,897	0,001	0,457	0,119	0,809	0,007
M9	120sec	0,688	0,523	0,85	0,748	0,59	0,908	0,574	0,217	0,926	0,001	0,48	0,133	0,843	0,006
M9	300sec	0,692	0,525	0,853	0,739	0,58	0,899	0,594	0,23	0,954	0,001	0,504	0,149	0,878	0,005
M9	600sec	0,687	0,522	0,848	0,728	0,57	0,886	0,6	0,233	0,964	0,001	0,515	0,161	0,887	0,004
M9	1800sec	0,67	0,506	0,829	0,706	0,548	0,859	0,599	0,232	0,965	0,001	0,515	0,173	0,875	0,003
M9	3600sec	0,657	0,493	0,814	0,69	0,536	0,838	0,606	0,229	0,983	0,001	0,544	0,201	0,903	0,002
M9	7200sec	0,652	0,479	0,82	0,681	0,519	0,84	0,554	0,17	0,939	0,003	0,559	0,213	0,921	0,001

Appendix E: Supplementary Tables, Figures and Equations

Additivity of AI metric

$$AI_i(t; N \cdot H) = \sum_{s=0}^N AI_i(t + sH; H)$$

Equation 6: *Additivity of AI metric*

In Equation 6 $N \in \mathbb{Z}$ is the factor with which the epoch length gets multiplied. $AI_i(t; N \cdot H)$ is only defined at $\{t: t/(N \cdot H) \in \mathbb{Z}\}$.

Boxplots of baseline values and slopes of the three main marker types and clinical markers for D1-D10

1. Baseline values:

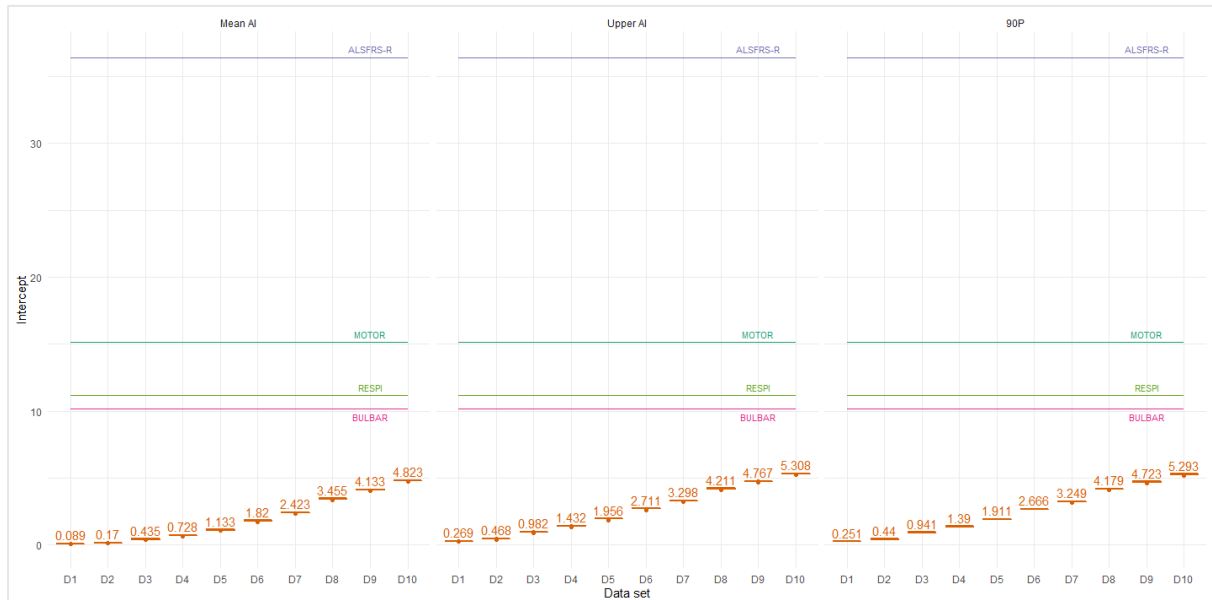


Figure 8: Boxplot of baseline values for three main marker types for D1-D10(orange). The boxplots indicate the spread between the markers based on 61 different AI-metrics. Above the boxplots, the mean intercepts over all 61 markers are displayed. The clinical markers of disease severity, ALSFRS-R (blue), MOTOR (green), BULBAR (pink) and RESPI (green) are also shown for comparison.

2. Slope

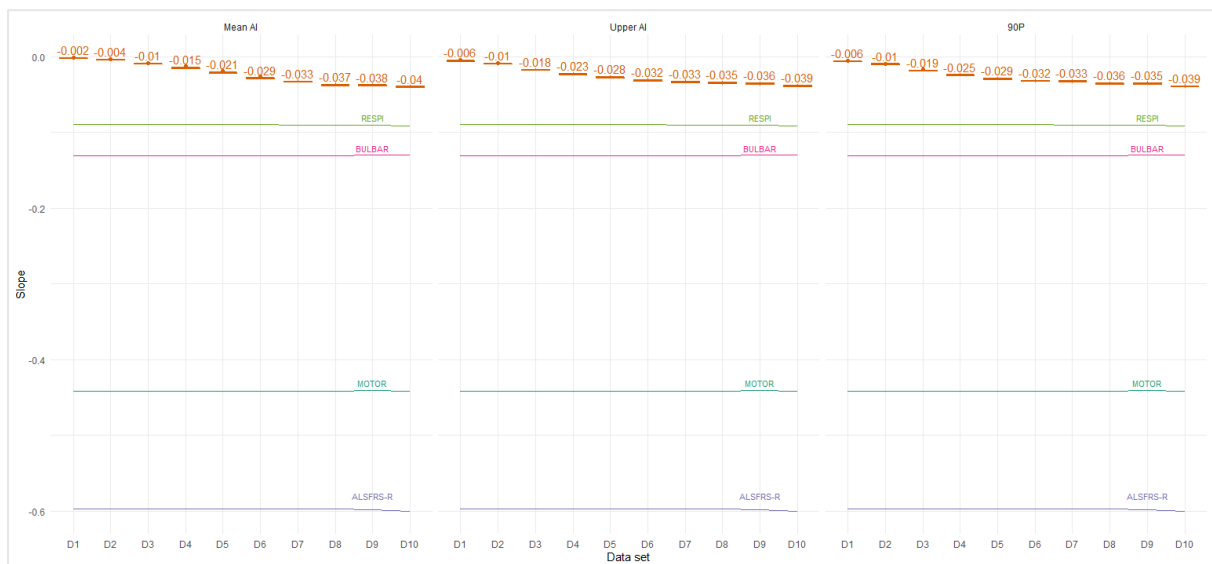


Figure 9: Boxplot of slopes for three main marker types for D1-D10(orange). The boxplots indicate the spread between the markers based on 61 different AI-metrics. The orange numbers show the mean slopes over all 61 AI metrics. The clinical markers of disease severity, ALSFRS-R (blue), MOTOR (green), BULBAR (pink) and RESPI (green) are also shown for comparison.

Boxplot of correlations of the three main marker types with A1 marker for D1-D10

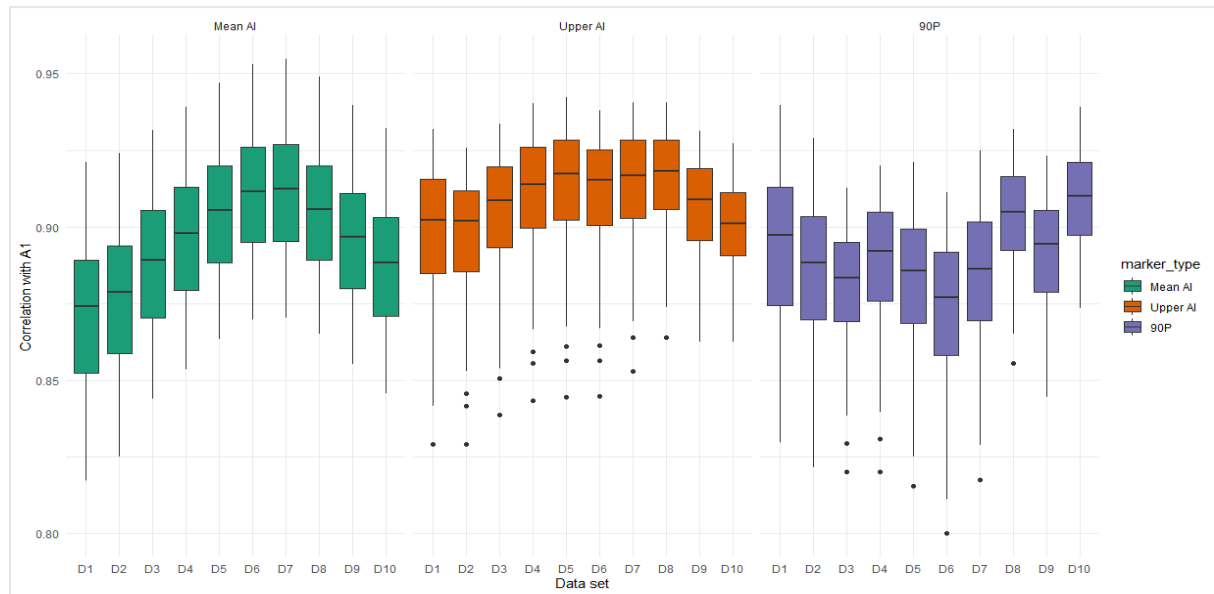


Figure 10: Longitudinal correlations of the three main marker types with A1 marker for D1-D10. The boxplots indicate the spread between the markers based on 61 different AI-metrics.

Tables containing final selection of Upper AI and 90P markers M6-M14 and their correlations with disease progression and CoV

1. Final selection of Upper AI and 90P markers

Marker type	ID	Reason for selection	Weights of modified AI
Upper AI	M6	Original AI, Rotational invariance	$w_1 = w_2 = w_3 = 1/3$
	M7	Highest mean correlation over all ten data sets	$w_1 = 0.7 ; w_2 = 0.3 ; w_3 = 0$
	M8	Highest correlation overall	$w_1 = 0.6 ; w_2 = 0.4 ; w_3 = 0$
	M9	Lowest mean CoV over all ten data sets	$w_1 = w_2 = 0 ; w_3 = 1$
	M9	Lowest CoV overall	$w_1 = w_2 = 0 ; w_3 = 1$
90P	M10	Original AI, Rotational invariance	$w_1 = w_2 = w_3 = 1/3$
	M11	Highest mean correlation over all ten data sets	$w_1 = 0.9 ; w_2 = 0.1 ; w_3 = 0$
	M12	Highest correlation overall	$w_1 = 1 ; w_2 = 0 ; w_3 = 0$
	M13	Lowest mean CoV over all ten data sets	$w_1 = 0.1 ; w_2 = 0.9 ; w_3 = 0$
	M14	Lowest CoV overall	$w_1 = w_2 = 0 ; w_3 = 1$

Table 4: Final selection of Upper AI and 90P markers. ID column contains the IDs of the individual markers. In the Weights of modified AI column, w_1 is the weight for the x-axis, w_2 for the y-axis and w_3 for the z-axis.

2. Longitudinal correlations with disease progression and CoV for Markers M6 -M9 (Upper AI)

ID	Selection method	Data set ID	Epoch length	Correlation ALSFRS-R	95% CI ALSFRS-R	Correlation MOTOR	95% CI MOTOR	CoV
M6	MaxCor	D6	300	0.71	0.57 – 0.87	0.78	0.62 – 0.93	0.96
	MinCoV	D2	10	0.67	0.51 – 0.83	0.76	0.60 – 0.93	0.87
	Mean	D1-D10	5-7200	0.69	0.54 – 0.85	0.76	0.60 – 0.92	0.94
M7	MaxCor	D6	300	0.72	0.56 – 0.87	0.79	0.63 – 0.94	0.99
	MinCoV	D2	10	0.68	0.52 – 0.83	0.77	0.61 – 0.93	0.89
	Mean	D1-D10	5-7200	0.70	0.54 – 0.85	0.77	0.61 – 0.92	0.96
M8	MaxCor	D6	300	0.72	0.56 – 0.87	0.79	0.63 – 0.94	0.98
	MinCoV	D3	30	0.70	0.55 – 0.85	0.79	0.64 – 0.94	0.89
	Mean	D1-D10	5-7200	0.70	0.54 – 0.85	0.77	0.61 – 0.95	0.95
M9	MaxCor	D6	300	0.69	0.53 – 0.85	0.74	0.58 – 0.90	0.96
	MinCoV	D2	10	0.63	0.47 – 0.80	0.71	0.55 – 0.88	0.83
	Mean	D1-D10	5-7200	0.66	0.50 – 0.83	0.72	0.56 – 0.88	0.93

Table 5: Longitudinal correlations with disease progression and CoV for markers M6 -M9 (UpperAI). MaxCor in the “Selection method” column means that the row contains the values of the data set, in which the marker had the highest correlation with the ALSFRS-R total score. MinCoV means that the row contains the values of the data set in which the marker had the lowest CoV. Mean means that this row contains the mean values over all ten data sets. Column Epoch length reminds the reader which data set contains which epoch length.

3. Longitudinal correlations with disease progression and CoV for Markers M10 -M14 (90P):

<i>ID</i>	<i>Selection method</i>	<i>Data set ID</i>	<i>Epoch length</i>	<i>Correlation ALSFRS-R</i>	<i>95% CI ALSFRS-R</i>	<i>Correlation MOTOR</i>	<i>95% CI MOTOR</i>	<i>CoV</i>
<i>M10</i>	MaxCor	D6	300	0.70	0.54 – 0.86	0.75	0.60 – 0.90	1.09
	MinCoV	D2	10	0.64	0.49 – 0.80	0.74	0.60 – 0.89	0.83
	Mean	D1-D10	5-7200	0.68	0.52 – 0.83	0.75	0.60 – 0.89	0.95
<i>M11</i>	MaxCor	D8	1800	0.72	0.55 – 0.89	0.77	0.62 – 0.93	1.02
	MinCoV	D2	10	0.66	0.50 – 0.82	0.76	0.61 – 0.92	0.89
	Mean	D1-D10	5-7200	0.69	0.53 – 0.85	0.77	0.61 – 0.92	1.02
<i>M12</i>	MaxCor	D8	1800	0.72	0.55 – 0.89	0.78	0.62 – 0.93	1.05
	MinCoV	D2	10	0.66	0.50 – 0.83	0.77	0.61 – 0.93	0.92
	Mean	D1-D10	5-7200	0.69	0.53 – 0.85	0.77	0.61 – 0.92	1.05
<i>M13</i>	MaxCor	D6	300	0.71	0.56 – 0.86	0.77	0.62 – 0.90	1.03
	MinCoV	D3	30	0.66	0.52 – 0.81	0.76	0.64 – 0.89	0.81
	Mean	D1-D10	5-7200	0.67	0.52 – 0.83	0.74	0.60 – 0.88	0.93
<i>M14</i>	MaxCor	D6	300	0.68	0.51 – 0.84	0.72	0.56 – 0.88	1.04
	MinCoV	D1	5	0.60	0.44 – 0.77	0.69	0.54 – 0.85	0.80
	Mean	D1-D10	5-7200	0.65	0.49 – 0.82	0.71	0.56 – 0.86	0.95

Table 6: Longitudinal correlations with disease progression and CoV for markers M10 -M14 (90P). MaxCor in the “Selection method” column means that the row contains the values of the data set, in which the marker had the highest correlation with the ALSFRS-R total score. MinCoV means that the row contains the values of the data set in which the marker had the lowest CoV. Mean means that this row contains the mean values over all ten data sets. Column Epoch length reminds the reader which data set contains which epoch length.

Boxplot of correlations of markers M1-M14 with AC-based markers

Figure 8 shows the correlations of markers M1-M14 with the count-based marker A1 and %Active by van Eijk et al. (3). On average, markers M5, M11, and M12 correlate less with A1, whereas markers M4 and M13 have a stronger correlation. A1 is a marker based on the variation in the vertical axis. Therefore, markers based on modified AI metrics with a large weight for the y-axis might correlate more strongly.

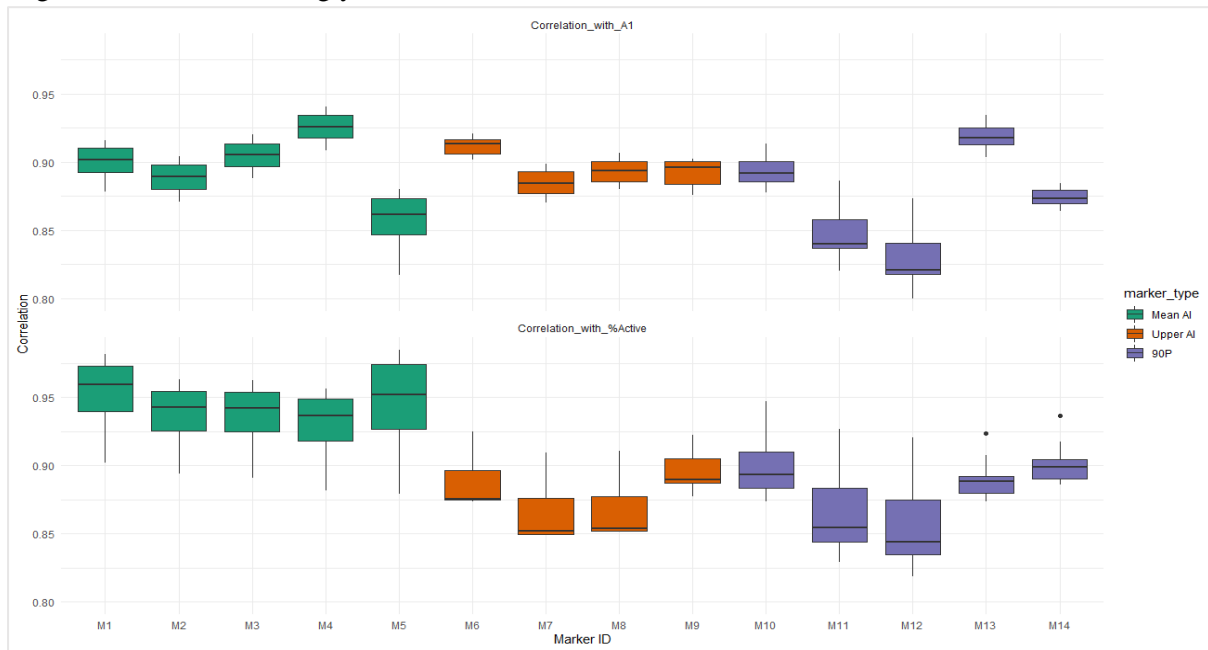


Figure 11: Correlations of markers M1-M14 with AC-markers A1 and %Active. The boxplots indicate the spread of the correlations between the ten data sets D1-D10.