

---

## GENEActiv Technical Note: AWD Conversion

### 1. DOCUMENT

#### 1.1 Introduction

The GENEActiv (2023) software provides a function to create AWD files from raw GENEActiv data recorded at any frequency.

This output does not attempt to reverse engineer the sensor and propriety algorithms of other devices, but rather approximate those outputs in a consistent and open manner.

The specification of the file format, the operation of the algorithm, and the development verification and validation processes are described in this technical note.

#### 1.2 Inspecting AWD files

The files are formatted with a 7-line header with subject and measurement period information, followed by comma-delimited values for each epoch. There can be data columns for just activity, activity and light levels or a wider range for more complicated devices. The best match for GENEActiv raw data conversion is activity and light levels.

The activity levels are output as counts based on the operation of a piezo-electric accelerometer and black box algorithm. The light levels are output as lux.

General observations:

- Epochs are of length 15, 30 and 60 seconds (coded 1, 2 & 4 as multiples of 15s).
- While active counts increase somewhat linearly with epoch length, sleep and inactive counts remain similar and a single threshold of 40 counts is commonly applied to all epoch lengths.
- Count scores appear to have a resolution of slightly greater than 1 count, although differences in signal noise may account for some of the variation. Quantisation effects are also apparent at a roughly 10-count resolution.
- Once epochs with counts less than 40 are removed from 15s, the distribution is closely exponential until 1000 counts.
- The percentage of epochs with counts 1-40 is roughly 15% of epochs with zero counts – similar to typical estimates of sleep inefficiency.

---

### 1.3 Estimation approach

Zero count epoch are commonly recognised as periods of no movement. In raw data accelerometry the standard deviation of the acceleration signal is the best, simple measure of movement. A single threshold is applied to the standard deviation of the vector magnitude of acceleration (combining x, y & z axes) with epochs below this threshold set to zero counts. The zero count rate per day is set to be equivalent to that output by devices that output AWD files directly.

All other epochs have a count score that is scaled based on the mean acceleration for the epoch to be as equivalent as possible to devices that output AWD files directly. This scaling is linear once the effects of lower acceleration range and noise floors have been compensated for. The scaling factor is a multiple of epoch length, so longer epochs have higher count scores.

The light output is based on the mean for the epoch and the 7-line header is populated with data from the raw data bin file header (see Appendix 1).

### 1.4 Activity count conversion process

- Limit ranges of the accelerometer axes output to  $\pm 2g$
- Calculate the vector magnitude of all the data points
- Split the vector magnitude output into segments for each epoch
- Find the arithmetic mean vector magnitude for each epoch
- Find the standard deviation of vector magnitude for each epoch
- Remove the noise floor of the accelerometer by subtracting 6.5mg and taking the absolute value
- Set all epochs with a standard deviation below 6.5mg to zero counts
- Multiply the epoch values by 5,000 to replicate counts values
- Scale the count values by the epoch length by multiplying by epoch length in seconds and dividing by 60
- Round to 0 dp

### 1.5 Light level conversion process

- Split the raw light output into segments for each epoch
- Find the arithmetic mean light level for each epoch
- Round to 1 dp

---

## 1.6 Verification

The conversion processes for the header, activity count and light meter were independently coded in R (see Appendix 2) and C# (the GENEActiv software application version 0.9.9). Outputs were compared from a range of input raw data bin files to ensure they were consistent.

Both code bases were also checked by inspection and their outputs were checked against the specification.

## 1.7 Validation

Data were collected simultaneously from GENEActiv and devices that output AWD files directly. The raw GENEActiv files were converted by the GENEActiv software.

The outputs of both systems were compared in widely used sleep analysis software that accepts AWD imports. The converted GENEActiv files were accepted by the sleep analysis software which was able to apply its standard rest / sleep interval algorithms without manual intervention.

The zero count rates for the data from both systems were compared and found to be within a few percent of each other. A linear regression comparing the count values of time-matched non-zero epochs demonstrated a roughly linear relationship with a slope close to 1.

## 2. APPENDICES

### 2.1 File format specification

Line	Description	Example
1	SubjectCode	jl
2	Start Date in yyyy-mm-dd ISO format	2021-60-13
3	Start Time in hh:mm (24 hour) format	10:43
4	Epoch code "1" = 15s, "2" = 30s, "4" = 60s	4
5	Subject age in years	48
6	Device Serial number with "P" prefix	P030092
7	Subject sex "M" or "F" or ""	M
8	activity_counts(0dp), mean_lux(1dp)	318 , 385.3
9	<i>As 8 until end of data</i>	

---

## 2.2 R code embodiment

```
# Load in the library needed
library(GENEAread)

bintoAWD<-function(binfile, epoch,output){

  # Read core binfile
  procfile = read.bin(binfile)
  filename = toString(as.character(output))
  filename = paste(filename, "AWD", sep = ".")

  #### Header ####

  # Subject code
  ID = as.character(procfile$header[10,])

  # Start date
  date = (procfile$header$value[4])
  date = parse.time(as.character(date), format = "POSIX") ## timezone not collected from file so may need adjusting
  startDate = strptime(date, "%Y-%m-%d")
  writeD = c(as.character(procfile$header[10,]),as.character(startDate))

  # Start time
  startTime = strptime(date, "%H:%M")
  writeD = append(writeD,as.character(startTime))

  # Epoch length
  writeD = append(writeD,as.character(epoch/15))

  # Subject age based on their birth date
  time = parse.time(as.character(Sys.time()), format = "POSIX")
  Age = floor(as.numeric(difftime(time, date, units = "days")/365.25))

  writeD = append(writeD,as.character(Age))

  # Actigraph ID
  ID = procfile$header$value[1]
  writeD = append(writeD,paste0("P",as.character(ID)))

  # Gender
  Gender = procfile$header[12,]
  if (Gender == "male"){
    Gender="M"}
  if (Gender=="female"){
    Gender = "F"}

  # Output header to a AWD file
  writeD = append(writeD,as.character(Gender))
  write(writeD,file = filename)

  #### Activity counts ####

  # Extract the axis and limits them to +/-2g
  x = pmax(pmin(procfile$data.out[,2],2),-2)
  y = pmax(pmin(procfile$data.out[,3],2),-2)
  z = pmax(pmin(procfile$data.out[,4],2),-2)

  # Work out the vector magnitude of all the data points
  df <-abs(sqrt(x^2+y^2+z^2)-1)

  # Find the number of data points in each segment based on the frequency and the epoch size
  n<-((as.numeric(procfile$header[2,]))*(epoch))
```

---

```
# Break vector magnitude into segments and fins the mean & SD of the vector magnitudes
dfb<-aggregate(as.matrix(df),list(rep(1:(nrow(as.matrix(df)))/%n+1),each=n,len=nrow(as.matrix(df))),mean)[-1];
dfc<-aggregate(as.matrix(df),list(rep(1:(nrow(as.matrix(df)))/%n+1),each=n,len=nrow(as.matrix(df))),sd)[-1];

combined = data.frame(
  dfb$V1,
  dfc$V1
)

# Remove noise floor of mean acceleration
combined$dfb.V1 = abs(combined$dfb.V1 - 0.0065)

# Set epochs with a SD below 6.5 mg to zero counts
combined$dfc.V1 = ifelse(combined$dfc.V1<0.0065,0,combined$dfc.V1)

# Scale to replicate counts range and include epoch length
combined$dfb.V1 = combined$dfb.V1*5000*(epoch/60)

# Format number as integer
df2 <- round(combined$dfb.V1,0)

### Light level ###

# Set up raw light level data
df<-procfile$data.out[,5]

# Break light data into segments and find the mean
df3<-aggregate(df,list(rep(1:(nrow(as.matrix(df)))/%n+1),each=n,len=nrow(as.matrix(df))),mean)[-1];
df3<-round(df3$x,1)

# Combine activity count and light level outputs
df = data.frame(
  df2,
  df3
)

# Appends activity count and light level information to the file
write.table(df, file = filename, row.names=FALSE,col.names=FALSE, append=TRUE, sep = " , ")
}

# Test
binfile = "jl_left_wrist_030092_2021-06-18 14-29-25_Calibrated.bin"
output = "jl_left_wrist_030092_2021-06-18 14-29-25_Calibrated_28Mar23_60s"
epoch = 60

bintoAWD(binfile, epoch, output)
```

**END OF DOCUMENT**