universalaccel (UA) Manual

Part 1: Generating Metrics • Part 2: Analysis & Interpretation

Contents

Abbreviations and Terms

- **UA** (*universalaccel*): the package ecosystem (functions + reference tables + output contracts).
- **ENMO** (*Euclidean Norm Minus One*): acceleration magnitude above 1g after removing the static component (autocalibration applied).
- **MAD** (*Mean Amplitude Deviation*): mean deviation of acceleration magnitude within an epoch.
- **AI** (*Activity Index*): variability-based activity metric computed from raw acceleration.
- **MIMS-unit** (*Monitor-Independent Movement Summary unit*): standardized activity summary intended to be comparable across devices.
- **ROCAM** (*Rate of Change Acceleration Movement*): change-based activity metric derived from successive acceleration movement samples.
- **AC** (*ActiGraph counts*): traditional "counts".
- **NHANES** (*National Health and Nutrition Examination Survey*): U.S. survey used here as the reference population for zone boundaries and percentile tables.
- **Anchor** (*default metric scale for conditioning*): the metric context used to define zones and percentiles; other metrics can be **equated** onto this anchor via **equipercentile** mapping.
- **Epoch** (*aggregation window*): the time length used to summarize raw samples into rows (e.g., 5 seconds or 60 seconds).
- **AA** (*Average acceleration*): the daily time-weighted mean of the intensity distribution (computed from bin midpoints × minutes in bin).
- **IG** (*Intensity gradient*): the slope from regressing log(time in bin) on log(intensity midpoint); more negative means a steeper drop-off at higher intensities.

Part 1 — Generating Metrics (how to produce harmonized files)

1. What UA is (universalaccel)

universalaccel (UA) is an R pipeline for computing harmonized accelerometer summary metrics across multiple devices (Tested on: ActiGraph, Axivity, GENEActiv). The goal is to make raw accelerometer data comparable by producing standardized, epoch-level outputs with consistent naming, time handling, and reproducible settings.

UA tested files and order of operations:

- ActiGraph (GT3X), .gt3x raw data is passed through `?agcounts::agcalibrate()` first and metrics are computed.
- Axivity (AX3), .resampled.CSV is autocalibrated in OmGui, so metrics are directly computed.
- GeneActiv, .bin raw data is passed through `?GENEAread::recalibrate()` first and metrics are computed.
- Note that this workflow does not eliminate any file based on the threshold of calibration error.

2. Inputs and expectations (file formats, columns, epochs)

What you provide

Depending on device type, you provide: - Raw device files (ActiGraph `.gt3x`, Axivity `resampled`.csv, `GENEActiv`.bin‘)

Key expectations

- A consistent alignment ofmetric at the selected epoch length (`epoch_sec`; commonly 5 seconds or 60 seconds).
- Correct sample rate and dynamic range for raw devices as configured (or when required by some summary metrics).
- Consistent timezone handling (typically this is handled by device software but it is configurable here).

3. Quick Start: generate metrics with `?accel_summaries()` (epoch-level outputs)

```r
library(universalaccel)

# Locate sample data bundled with the package (ActiGraph example)
actigraph_path <- system.file("extdata/actigraph", package = "universalaccel")
output_dir <- tempdir()

# Run metric generation.
# The set up can hold other arguments see more:?universalaccel::accel_summaries()
accel_summaries(
  device        = "actigraph", #define device type
  data_folder   = actigraph_path, # data folder for batch processing
  output_folder = output_dir, # output folder
  epochs        = 60,
  sample_rate   = 100,
```

```r
  dynamic_range = c(-8, 8),
  metrics       = c("MIMS","ENMO","MAD","COUNTS","AI","ROCAM"),
  apply_nonwear = TRUE
)
#> Loading chunk: 1
#> ================================================
#> ================================
```

4. Output files and variable list (epoch-level metrics)

After `accel_summaries()`, you will typically see one or more CSV files written to `output_folder`.

```r
# List generated CSVs
files <- list.files(output_dir, pattern = "UA_actigraph", full.names = TRUE)
# Processed files
basename(files)
#> [1] "UA_actigraph_epoch60s_2025-12-26.csv"

filecheck <- function(pattern, index = 1) {
  f <- list.files(output_dir, pattern = pattern, full.names = TRUE)
  if (!length(f)) stop("No files found.")

  if (index > length(f)) stop("Index out of range.")

  readr::read_csv(f[index], n_max = 3)
}

filecheck("UA_actigraph", index = 1)
#> # A tibble: 3 x 15
#>   time                MIMS_UNIT MIMS_UNIT_X MIMS_UNIT_Y MIMS_UNIT_Z ID      ENMO
#>   <dttm>                  <dbl>       <dbl>       <dbl>       <dbl> <chr>  <dbl>
#> 1 2023-06-13 08:34:00      25.8        8.82        7.12        9.83 examp~  107.
#> 2 2023-06-13 08:35:00      23.8       11.8         5.35        6.67 examp~  161.
#> 3 2023-06-13 08:36:00      21.5        9.30        5.75        6.42 examp~  144.
#> # i 8 more variables: MAD <dbl>, Axis1 <dbl>, Axis2 <dbl>, Axis3 <dbl>,
#> #   Vector.Magnitude <dbl>, AI <dbl>, ROCAM <dbl>, choi_nonwear <lgl>
```

Typical epoch-level variables (what they mean)

Exact columns depend on device and selected metrics, but commonly include:

- `time` (*timestamp*; start of the epoch)
- `ID` or `id` (*participant ID*; derived from file name or metadata)
- `epoch_sec` (*epoch length in seconds*)
- `ENMO` (*Euclidean Norm Minus One*)
- `MAD` (*Mean Amplitude Deviation*)
- `AI` (*Activity Index*)
- `MIMS_UNIT` (*Monitor-Independent Movement Summary unit*)
- `ROCAM` (*Rate of Change Acceleration Movement*)
- `Axis1`, `Axis2`, `Axis3` or `X`, `Y`, `Z` (*axis components*)
- `Vector.Magnitude` (*ActiGraph/Activity Counts*)

- `choi_nonwear` (*nonwear flag: epoch-aware*)

```r
# In your UA project root:
renv::init()       # once
renv::snapshot()   # whenever dependencies change
```

Part 2 — Analysis & Interpretation (precomputed files → distributions → zones → percentiles)

6. What "analysis-ready" means in UA

What you provide

Running this part necessitates pre-computed data of summary metric(s). Therefore, you provide:

- A pre-computed/processed table already aligned to an epoch and containing metric columns with age and sex columns.
- Note that the Second part is run on a file whose nonwear time was already removed (choi method was used when generating NHANES comparisons). Ideally one would pass UA part 1 outputs to part 2.

An "analysis-ready" UA file has:

- consistent time and epoch (`epoch_sec`)
- consistent metric naming (e.g., `enmo`, `mad`, `mims_unit`, `ac`)
- enough metadata to interpret results (age/sex when relevant; location label)

In UA, Part 2 analysis commonly produces:

- **daily distributions** (bins; minutes across intensity ranges)
- **zone summaries** (movement zones from NHANES references)
- **AA** (Average acceleration) and **IG** (Intensity gradient)
- **NHANES positioning** (percentiles and anchor-conditioned lookups)

7. Daily summaries and binned distributions (bins, zones)

Two common analysis constructs:

- **AA (Average acceleration):** the time-weighted mean intensity over the day.
- **IG (Intensity gradient):** the slope from regressing log(time-in-bin) on log(intensity-bin-midpoint).

Conceptually:

- Bins summarize *how much time* is accumulated in each intensity range.
- Zones group bins into interpretable "movement zones" based on normative boundaries.

8. NHANES references (National Health and Nutrition Examination Survey)

NHANES references provide:

- **normative zone boundaries** (per metric, epoch, NHANES age group, and sex when available)
- **normative percentiles** for AA (Average acceleration) and IG (Intensity gradient), by NHANES age ranges (e.g., Age 3–19 and Age 20–80)

UA uses NHANES references to answer:

- "Where does this person-day rank relative to a U.S. reference?"
- "What would other metrics look like at the same **anchor** percentile?" (anchor-conditioned lookup)

9. Quick Start (Part 2): run analysis on the bundled precomputed demo file

UA includes a small but "analysis-capable" precomputed demo file under:

`inst/extdata/Precomputed/UA_FLASH_sample.csv`

This demo is intentionally larger than the tiny raw samples (so AA, IG, and weekly rollups are meaningful).

```r
library(universalaccel)
library(readr)

# Precomputed demo file shipped with the package
pre_base <- "UA_FLASH_sample"
pre_csv  <- system.file(
  "extdata/Precomputed",
  paste0(pre_base, ".csv"),
  package = "universalaccel"
)

stopifnot(nzchar(pre_csv))  # ensures the file exists in the installed package

# Where to write Part 2 outputs
out_root <- tempdir()

# Run Part 2 analysis (writes "<pre_base>_UA_outputs" under out_root)
analysis <- ua_analyze_precomputed(
  in_path     = pre_csv,
  out_dir     = out_root,
  location    = "ndw",
  make_weekly = TRUE,
  overwrite   = TRUE
)

# Inspect files written (short names only to avoid overflow)
part2_dir <- file.path(out_root, paste0(pre_base, "_UA_outputs"))
basename(list.files(part2_dir, full.names = TRUE))
#> [1] "UA_Output1_BINS_epoch60s_2025-12-26.csv"
#> [2] "UA_Output2_SUMMARY_epoch60s_2025-12-26.csv"
#> [3] "UA_Output3_PERCENTILES_epoch60s_2025-12-26.csv"
#> [4] "UA_Output4_WEEKLY_epoch60s_2025-12-26.csv"
#> [5] "UA_Output5_LOGISTICS_epoch60s_2025-12-26.txt"

# Safely preview outputs
preview <- function(pattern) {
  f <- list.files(part2_dir, pattern = pattern, full.names = TRUE)
  if (length(f)) read_csv(
    f[1],
    n_max = 5,
    col_types = cols(
      sex      = col_character(),
      sex_name = col_character()
    )
  )
}
```

```r
preview("UA_Output1_BINS")
#> # A tibble: 5 x 21
#>   id      metric    bin_i   day     n time_bin_min lower upper midpoint width
#>   <chr>   <chr>     <dbl> <dbl> <dbl>        <dbl> <dbl> <dbl>    <dbl> <dbl>
#> 1 F3BA010 mims_unit     1     1   913          913   0    10.0     5.02 10.0
#> 2 F3BA010 mims_unit     2     1   263          263  10.0  22.7    16.4  12.7
#> 3 F3BA010 mims_unit     3     1    67           67  22.7  26.3    24.5   3.56
#> 4 F3BA010 mims_unit     4     1    40           40  26.3  28.9    27.6   2.60
#> 5 F3BA010 mims_unit     5     1    28           28  28.9  31.4    30.1   2.55
#> # i 11 more variables: log_midpoint <dbl>, log_time_bin <dbl>, epoch_sec <dbl>,
#> #   location <chr>, cm_time_min <dbl>, sex <chr>, age <dbl>, category <chr>,
#> #   category_key <chr>, anchor <chr>, intensity_zone_raw <chr>
preview("UA_Output2_SUMMARY")
#> # A tibble: 5 x 30
#>   id      day epoch_sec location metric anchor sex     age category category_key
#>   <chr> <dbl>     <dbl> <chr>    <chr>  <chr>  <chr> <dbl> <chr>    <chr>
#> 1 F3BA~     1        60 ndw      ac     ac     F        18 Age-18-~ Age-18-64_S~
#> 2 F3BA~     1        60 ndw      ac     ac     F        18 Age-18-~ Age-18-64_S~
#> 3 F3BA~     1        60 ndw      ac     ac     F        18 Age-18-~ Age-18-64_S~
#> 4 F3BA~     1        60 ndw      ac     ac     F        18 Age-18-~ Age-18-64_S~
#> 5 F3BA~     1        60 ndw      ac     ac     F        18 Age-18-~ Age-18-64_S~
#> # i 20 more variables: intensity_zone <chr>, minutes_in_zone <dbl>,
#> #   observed_intensity <dbl>, nhanes_anchor_zone_lower <dbl>,
#> #   nhanes_anchor_zone_upper <dbl>, nhanes_anchor_mean_nhanesw <dbl>,
#> #   nhanes_anchor_se_nhanesw <dbl>, nhanes_anchor_total_min_ref <dbl>,
#> #   equated_ac_mean_nhanesw <lgl>, equated_ai_mean_nhanesw <dbl>,
#> #   equated_enmo_mean_nhanesw <dbl>, equated_mad_mean_nhanesw <dbl>,
#> #   equated_mims_unit_mean_nhanesw <dbl>, equated_rocam_mean_nhanesw <dbl>, ...
preview("UA_Output3_PERCENTILES")
#> # A tibble: 5 x 19
#>   id         day epoch_sec location sex_name   age age_range anchor stat
#>   <chr>    <dbl>     <dbl> <chr>    <chr>    <dbl> <chr>     <chr>  <chr>
#> 1 F3BA010      1        60 ndw      F           18 Age 3-19  ac     Average acce~
#> 2 F3BA010      1        60 ndw      F           18 Age 3-19  ac     Intensity gr~
#> 3 F3BA010      1        60 ndw      F           18 Age 3-19  enmo   Average acce~
#> 4 F3BA010      1        60 ndw      F           18 Age 3-19  enmo   Intensity gr~
#> 5 F3BA010      1        60 ndw      F           18 Age 3-19  mad    Average acce~
#> # i 10 more variables: anchor_value_obs <dbl>, anchor_nearest_centile <dbl>,
#> #   nhanes_ac_value_at_anchor_centile <dbl>, nhanes_ac_centile_used <dbl>,
#> #   nhanes_enmo_value_at_anchor_centile <dbl>, nhanes_enmo_centile_used <dbl>,
#> #   nhanes_mad_value_at_anchor_centile <dbl>, nhanes_mad_centile_used <dbl>,
#> #   nhanes_mims_unit_value_at_anchor_centile <dbl>,
#> #   nhanes_mims_unit_centile_used <dbl>
preview("UA_Output4_WEEKLY")
#> # A tibble: 5 x 12
#>   id      metric    epoch_sec location sex     age category   week n_days
#>   <chr>   <chr>         <dbl> <chr>    <chr> <dbl> <chr>     <dbl>  <dbl>
#> 1 F3BA010 mims_unit        60 ndw      F        18 Age-18-64     1      1
#> 2 F3BA014 mims_unit        60 ndw      F        18 Age-18-64     1      1
#> 3 F3BA010 ai               60 ndw      F        18 Age-18-64     1      1
#> 4 F3BA014 ai               60 ndw      F        18 Age-18-64     1      1
#> 5 F3BA010 mad              60 ndw      F        18 Age-18-64     1      1
#> # i 3 more variables: total_min_day_mean <dbl>, overall_intensity_mean <dbl>,
```

```
#> #   ig_slope_mean <dbl>
```

10. Output tables (Output1–Output5) and variable lists

This manual assumes the Part 2 runner produces:

Output1 — Bins (distribution)

Purpose: minute distribution across bins, with zone assignment.
Common variables: - `id`, `day`, `epoch_sec`, `location`, `metric` - `lower`, `upper`, `midpoint` (*bin boundaries*) - `time_bin_min` (*minutes in bin*) - `intensity_zone_raw` (*zone label*)

Output2 — Zone summary + NHANES anchors and "equated" values

Purpose: summarize minutes and intensity within zones; attach NHANES references.
Common variables:

- `minutes_in_zone` (*observed minutes*)
- `observed_intensity` (*observed time-weighted mean within zone*)
- `nhanes_anchor_*` (*NHANES reference values for the same metric/anchor*)
- `equated_*` (*NHANES mean/SE for other metrics mapped onto the same anchor-zone definition*)

Output3 — Percentile positioning (AA + IG), anchor-conditioned NHANES lookup

Purpose: (1) find your anchor percentile; (2) report NHANES values for other metrics at that same anchor percentile.
Common variables:

- `anchor_value_obs` (*observed AA or IG for the anchor metric*)
- `anchor_nearest_centile` (*nearest NHANES centile for that value*)
- `nhanes_<m>_value_at_anchor_centile` (*NHANES value for metric m at that same centile under the same anchor panel*)
- `nhanes_<m>_centile_used` (*centile actually used; should match anchor centile unless snapped*)

Output4 — Weekly rollups (optional)

Purpose: quick weekly averaging across days.
Common variables: - `week`, `n_days`, `overall_intensity_mean`, `ig_slope_mean`

Output5 — Run report (plain language)

Purpose: a plain-language summary of what ran, what was produced, and how to interpret the columns.

Reference

Part1 Implementation https://doi.org/10.1123/jmpb.2025-0024