universalaccel (UA) Manual

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

Contents

Abbreviations

- **UA** (*universalaccel*): the package ecosystem (functions + reference tables + output contracts).
- **ENMO** (*Euclidean Norm Minus One*): acceleration magnitude above 1g after removing the static component (device-calibrated where applicable).
- **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 Magnitude*): change-based activity metric derived from successive acceleration magnitude samples.
- **AC** (*ActiGraph counts*): traditional "counts" (including vector magnitude counts when configured).
- **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.
- **LUT** (*Lookup Table*): mapping table used for zones/percentiles.
- **CLI** (*Command Line Interface*): non-interactive run mode.
- **YAML** (*YAML Ain't Markup Language*): configuration file format (e.g., `config.yml`).

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 (e.g., 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 typically supports metrics such as ENMO (Euclidean Norm Minus One), MAD (Mean Amplitude Deviation), AI (Activity Index), MIMS-unit (Monitor-Independent Movement Summary), ROCAM (Rate of Change Acceleration Magnitude), and AC (ActiGraph counts / activity counts).

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

What you provide

Depending on device type, you provide either: - Raw device files (e.g., ActiGraph `.gt3x`, Axivity `resampled.csv`, GENEActiv `.bin`), **or** - A precomputed/processed table already aligned to an epoch and containing metric columns.

Key expectations

- A consistent epoch for the output you want (`epoch_sec`; commonly 5 seconds or 60 seconds).
- Correct sample rate and dynamic range for raw devices (when required).
- Consistent timezone handling (recommended: be explicit).

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

```
library(universalaccel)

actigraph_path <- system.file("extdata/actigraph", package = "universalaccel")
list.files(actigraph_path)
#> [1] "example.gt3x"

## Define output directory
output_dir <- tempdir()

## Run the pipeline
accel_summaries(
  device        = "actigraph",
  data_folder   = actigraph_path,
  output_folder = output_dir,
  epochs        = 60,
  sample_rate   = 100,
  dynamic_range = c(-8, 8),
  metrics       = c("AI","Counts","ENMO","MAD","MIMS","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`.

```
# List generated CSVs
list.files(output_dir, pattern = "UA_actigraph", full.names = TRUE)
#> [1] "C:\\Users\\user\\AppData\\Local\\Temp\\RtmpgfgQgP/UA_actigraph_epoch60s_2025-12-24.csv"
readr::read_csv(list.files(output_dir, pattern = "UA_actigraph", full.names = TRUE)[1], n_max = 5)
#> # A tibble: 3 x 15
#>   time                   AI ID    Axis1 Axis2 Axis3 Vector.Magnitude  ENMO   MAD
#>   <dttm>              <dbl> <chr> <dbl> <dbl> <dbl>            <dbl> <dbl> <dbl>
#> 1 2023-06-13 08:34:00 19.8  exam~  2606  3116  3542             5389  107.  147.
#> 2 2023-06-13 08:35:00 22.5  exam~  1738  3943  2840             5161  161.  204.
#> 3 2023-06-13 08:36:00  9.88 exam~  2172  3363  2646             4799  144.  160.
#> # i 6 more variables: MIMS_UNIT <dbl>, MIMS_UNIT_X <dbl>, MIMS_UNIT_Y <dbl>,
#> #   MIMS_UNIT_Z <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 Magnitude*)
- `Axis1`, `Axis2`, `Axis3` or `X`, `Y`, `Z` (*axis components; optional*)
- `Vector.Magnitude` (*optional*)
- `choi_nonwear` (*nonwear flag; optional; depends on settings*)

5. Reproducibility & long-term stability checklist (avoid "degradable" workflows)

Recommended stability practices

- Use `renv` to pin package versions.
- Keep a stable `config.yml` (YAML Ain't Markup Language) with explicit settings.
- Avoid relying on "latest" upstream behavior when possible.
- Store raw inputs read-only; write outputs to versioned folders.
- Save the UA version used (package version) alongside outputs.

```
# 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

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_actigraph_epoch`

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

```r
library(universalaccel)

# Precomputed demo file shipped with the package
pre_base <- "UA_actigraph_epoch60s_2025-12-24"
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
analysis <- ua_analyze_precomputed(
  in_path     = pre_csv,
  out_dir     = out_root,
```

```r
  location    = "ndw",
  make_weekly = TRUE,
  overwrite   = TRUE
)

# Inspect files written
part2_dir <- file.path(out_root, paste0(pre_base, "_UA_outputs"))
list.files(part2_dir, full.names = TRUE)
#> [1] "C:\\Users\\user\\AppData\\Local\\Temp\\RtmpgfgQgP/UA_actigraph_epoch60s_2025-12-24_UA_outputs/UA
#> [2] "C:\\Users\\user\\AppData\\Local\\Temp\\RtmpgfgQgP/UA_actigraph_epoch60s_2025-12-24_UA_outputs/UA
#> [3] "C:\\Users\\user\\AppData\\Local\\Temp\\RtmpgfgQgP/UA_actigraph_epoch60s_2025-12-24_UA_outputs/UA
#> [4] "C:\\Users\\user\\AppData\\Local\\Temp\\RtmpgfgQgP/UA_actigraph_epoch60s_2025-12-24_UA_outputs/UA
#> [5] "C:\\Users\\user\\AppData\\Local\\Temp\\RtmpgfgQgP/UA_actigraph_epoch60s_2025-12-24_UA_outputs/UA

# Safely preview outputs
preview <- function(pattern) {
  f <- list.files(part2_dir, pattern = pattern, full.names = TRUE)
  if (length(f)) readr::read_csv(f[1], n_max = 3)
}

preview("UA_Output1_BINS")
#> # A tibble: 3 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 UA00001 enmo       1     1   998          998     0    25     12.5    25
#> 2 UA00001 enmo       2     1    22           22    25    50     37.5    25
#> 3 UA00001 enmo       1     2  1439         1439     0    25     12.5    25
#> # 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: 3 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 UA00~     1        60 ndw      ac     ac     M        61 Age-18-~ Age-18-64_S~
#> 2 UA00~     1        60 ndw      ac     ac     M        61 Age-18-~ Age-18-64_S~
#> 3 UA00~     1        60 ndw      ac     ac     M        61 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: 3 x 19
#>   id        day epoch_sec location sex_name   age age_range anchor    stat
#>   <chr>   <dbl>     <dbl> <chr>    <chr>    <dbl> <chr>     <chr>     <chr>
#> 1 UA00001     1        60 ndw      M           61 Age 20-80 ac        Average a~
#> 2 UA00001     1        60 ndw      M           61 Age 20-80 enmo      Average a~
#> 3 UA00001     1        60 ndw      M           61 Age 20-80 mims_unit Average a~
#> # 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: 3 x 12
#>   id      metric    epoch_sec location sex     age category   week n_days
#>   <chr>   <chr>         <dbl> <chr>    <chr> <dbl> <chr>     <dbl>  <dbl>
#> 1 UA00001 enmo             60 ndw      M        61 Age-18-64     1      7
#> 2 UA00002 enmo             60 ndw      F        48 Age-18-64     1      7
#> 3 UA00001 mims_unit        60 ndw      M        61 Age-18-64     1      7
#> # 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.