

Package: PoseFixeR (via r-universe)

October 3, 2026

Type Package

Title Anomaly Detection and Correction for OpenPose Gait Analysis

Version 0.2.1

Description Detects and corrects the anatomical, biomechanical, physical, and detection-confidence anomalies that arise in two-dimensional video-based gait analysis with OpenPose, then imputes and smooths the corrected keypoint trajectories. Implements the method of Sugiyama, Uno and Matsui (2023) <[doi:10.1371/journal.pcbi.1009989](https://doi.org/10.1371/journal.pcbi.1009989)>.

License GPL-3

URL <https://github.com/x-biosignal/PoseFixeR>,
<https://x-biosignal.r-universe.dev/PoseFixeR>,
<https://x-biosignal.github.io/PoseFixeR>

BugReports <https://github.com/x-biosignal/PoseFixeR/issues>

Encoding UTF-8

Imports stats, imputeTS, graphics, rjson

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev

Repository <https://x-biosignal.r-universe.dev>

Date/Publication 2026-10-03 13:11:49 UTC

RemoteUrl <https://github.com/x-biosignal/PoseFixeR>

RemoteRef HEAD

RemoteSha b96ed35373d384e065a03b0f601a73c397675d8f

Contents

anomaly	2
diplay_frames	3
distance	3
hdegree	4
import_json	5
kdegree	6
normalize	6
plt.segment	7
posefixer	8
reshaping	9
Index	10

anomaly	<i>Flag statistical outliers</i>
---------	----------------------------------

Description

Flags values lying more than k standard deviations from the mean.

Usage

```
anomaly(x, k = 2)
```

Arguments

- x A numeric vector.
- k Number of standard deviations for the threshold (default 2).

Value

A list with caselist (logical outlier flags) and abnormal.values (the sorted flagged values).

Examples

```
set.seed(1)
x <- c(rnorm(30), 8)      # a series with one clear outlier
flags <- anomaly(x, k = 2)
which(flags$caselist)     # position of the flagged value(s)
flags$abnormal.values     # the flagged value(s)
```

diplay_frames	<i>Plot a sequence of pose frames</i>
---------------	---------------------------------------

Description

Lays out one scatter-plus-skeleton panel per frame (see [plt.segment\(\)](#)).

Usage

```
diplay_frames(obj, layout = c(5, 5), xlim = c(450, 800), ylim = c(700, 200))
```

Arguments

- obj Reshaped keypoint data (see [reshaping\(\)](#)): a list with x and y matrices (frames x keypoints).
- layout Length-2 vector c(rows, cols) passed to [graphics::par\(\)](#).
- xlim Length-2 vector for the x-axis range.
- ylim Length-2 vector for the y-axis range (top-down image coordinates).

Value

Called for its side effect of plotting; returns NULL.

Examples

```
parts <- c("nose", "neck", "Rshoulder", "Relbow", "Rwrist", "Lshoulder", "Lelbow",
  "Lwrist", "Rhip", "Rknee", "Rankle", "Lhip", "Lknee", "Lankle", "Reye", "Leye",
  "Rear", "Lear")
bx <- c(0,0,-.25,-.3,-.32,.25,.3,.32,-.18,-.2,-.22,.18,.2,.22,-.05,.05,-.1,.1)
by <- c(-.3,0,.05,.5,.95,.05,.5,.95,1,1.65,2.28,1,1.65,2.28,-.35,-.35,-.3,-.3)
obj <- list(x = matrix(bx, 2, 18, byrow = TRUE, dimnames = list(NULL, parts)),
  y = matrix(by, 2, 18, byrow = TRUE, dimnames = list(NULL, parts)))
op <- graphics::par(no.readonly = TRUE)
diplay_frames(obj, layout = c(1, 2), xlim = c(-1, 1), ylim = c(2.5, -0.5))
graphics::par(op)
```

distance	<i>Euclidean distance between two keypoints</i>
----------	---

Description

Euclidean distance between two keypoints

Usage

```
distance(dat, a, b, time)
```

Arguments

dat	Reshaped keypoint data (see reshaping()).
a, b	Keypoint indices or names.
time	Frame index or vector of frames.

Value

The distance(s) between keypoints a and b.

Examples

```
## a minimal reshaped pose: 18 keypoints over three frames
parts <- c("nose", "neck", "Rshoulder", "Relbow", "Rwrist", "Lshoulder", "Lelbow",
  "Lwrist", "Rhip", "Rknee", "Rankle", "Lhip", "Lknee", "Lankle", "Reye", "Leye",
  "Rear", "Lear")
bx <- c(0,0,-.25,-.3,-.32,.25,.3,.32,-.18,-.2,-.22,.18,.2,.22,-.05,.05,-.1,.1)
by <- c(-.3,0,.05,.5,.95,.05,.5,.95,1,1.65,2.28,1,1.65,2.28,-.35,-.35,-.3,-.3)
dat <- list(x = matrix(bx, 3, 18, byrow = TRUE, dimnames = list(NULL, parts)),
  y = matrix(by, 3, 18, byrow = TRUE, dimnames = list(NULL, parts)))
## shoulder width over frames 1:3 (by name) and hip width (by index)
distance(dat, "Rshoulder", "Lshoulder", 1:3)
distance(dat, 9, 12, 1)
```

hdegree	<i>Hip joint angle between three keypoints</i>
---------	--

Description

Hip joint angle between three keypoints

Usage

```
hdegree(dat, a, b, c, time)
```

Arguments

dat	Reshaped keypoint data (see reshaping()).
a, b, c	Keypoint indices or names (angle measured at b).
time	Frame index or vector of frames.

Value

The hip angle(s) in degrees.

Examples

```
## a minimal reshaped pose (one frame is enough for a single angle)
parts <- c("nose", "neck", "Rshoulder", "Relbow", "Rwrist", "Lshoulder", "Lelbow",
  "Lwrist", "Rhip", "Rknee", "Rankle", "Lhip", "Lknee", "Lankle", "Reye", "Leye",
  "Rear", "Lear")
bx <- c(0,0,-.25,-.3,-.32,.25,.3,.32,-.18,-.2,-.22,.18,.2,.22,-.05,.05,-.1,.1)
by <- c(-.3,0,.05,.5,.95,.05,.5,.95,1,1.65,2.28,1,1.65,2.28,-.35,-.35,-.3,-.3)
dat <- list(x = matrix(bx, 2, 18, byrow = TRUE, dimnames = list(NULL, parts)),
  y = matrix(by, 2, 18, byrow = TRUE, dimnames = list(NULL, parts)))
## right hip flexion at the hip (neck - Rhip - Rknee) in frame 1
hdegree(dat, 2, 9, 10, 1)
```

import_json

*Import OpenPose JSON files in a batch***Description**

Import OpenPose JSON files in a batch

Usage

```
import_json(
  files,
  parts = c("nose", "neck", "Rshoulder", "Relbow", "Rwrist", "Lshoulder", "Lelbow",
    "Lwrist", "Rhip", "Rknee", "Rankle", "Lhip", "Lknee", "Lankle", "Reye", "Leye",
    "Rear", "Lear")
)
```

Arguments

files Character vector of OpenPose JSON file paths (one per frame).
parts Keypoint names in row order (default: the 18-keypoint layout).

Value

A list of per-frame keypoint matrices (keypoint x c("X", "Y", "P")).

Examples

```
## Each file holds one frame as a flat vector of 54 values
## (18 keypoints x X, Y, confidence), written to a temporary directory here.
bx <- c(0,0,-.25,-.3,-.32,.25,.3,.32,-.18,-.2,-.22,.18,.2,.22,-.05,.05,-.1,.1)
by <- c(-.3,0,.05,.5,.95,.05,.5,.95,1,1.65,2.28,1,1.65,2.28,-.35,-.35,-.3,-.3)
frame <- as.numeric(t(cbind(bx, by, 1))) # row-major X, Y, P per keypoint
f <- tempfile(fileext = ".json")
writeLines(rjson::toJSON(frame), f)
keypoints <- import_json(f)
dim(keypoints[[1]]) # 18 keypoints x 3 columns
head(keypoints[[1]])
```

kdegree	<i>Knee joint angle between three keypoints</i>
---------	---

Description

Knee joint angle between three keypoints

Usage

kdegree(dat, a, b, c, time)

Arguments

- dat Reshaped keypoint data (see [reshaping\(\)](#)).
- a, b, c Keypoint indices or names (angle measured at b).
- time Frame index or vector of frames.

Value

The knee angle(s) in degrees.

Examples

```
## a minimal reshaped pose (one frame is enough for a single angle)
parts <- c("nose", "neck", "Rshoulder", "Relbow", "Rwrist", "Lshoulder", "Lelbow",
  "Lwrist", "Rhip", "Rknee", "Rankle", "Lhip", "Lknee", "Lankle", "Reye", "Leye",
  "Rear", "Lear")
bx <- c(0,0,-.25,-.3,-.32,.25,.3,.32,-.18,-.2,-.22,.18,.2,.22,-.05,.05,-.1,.1)
by <- c(-.3,0,.05,.5,.95,.05,.5,.95,1,1.65,2.28,1,1.65,2.28,-.35,-.35,-.3,-.3)
dat <- list(x = matrix(bx, 2, 18, byrow = TRUE, dimnames = list(NULL, parts)),
  y = matrix(by, 2, 18, byrow = TRUE, dimnames = list(NULL, parts)))
## right knee angle (Rhip - Rknee - Rankle) in frame 1
kdegree(dat, 9, 10, 11, 1)
```

normalize	<i>Normalizing coordinates from OpenPose</i>
-----------	--

Description

Normalizing coordinates from OpenPose

Usage

normalize(obj)

Arguments

obj Reshaped keypoint data (see [reshaping\(\)](#)): a list with x, y and confidence matrices whose columns are named keypoints.

Value

The neck-normalised keypoint data, in the same x/y/p structure.

Examples

```
## a minimal reshaped pose over three frames
parts <- c("nose", "neck", "Rshoulder", "Relbow", "Rwrist", "Lshoulder", "Lelbow",
  "Lwrist", "Rhip", "Rknee", "Rankle", "Lhip", "Lknee", "Lankle", "Reye", "Leye",
  "Rear", "Lear")
bx <- c(0,0,-.25,-.3,-.32,.25,.3,.32,-.18,-.2,-.22,.18,.2,.22,-.05,.05,-.1,.1)
by <- c(-.3,0,.05,.5,.95,.05,.5,.95,1,1.65,2.28,1,1.65,2.28,-.35,-.35,-.3,-.3)
dat <- list(x = matrix(bx, 3, 18, byrow = TRUE, dimnames = list(NULL, parts)),
  y = matrix(by, 3, 18, byrow = TRUE, dimnames = list(NULL, parts)),
  p = matrix(1, 3, 18, dimnames = list(NULL, parts)))
norm <- normalize(dat)
norm$x[, "neck"] # neck is the origin after normalisation
```

plt.segment

*Create segment line for frame plot***Description**

Create segment line for frame plot

Usage

```
plt.segment(x, y)
```

Arguments

x Numeric vector of keypoint x-coordinates for one frame.
y Numeric vector of keypoint y-coordinates for one frame.

Value

Called for its side effect of drawing the segments; returns NULL.

Examples

```
bx <- c(0,0,-.25,-.3,-.32,.25,.3,.32,-.18,-.2,-.22,.18,.2,.22,-.05,.05,-.1,.1)
by <- c(-.3,0,.05,.5,.95,.05,.5,.95,1,1.65,2.28,1,1.65,2.28,-.35,-.35,-.3,-.3)
## open a plot of the keypoints first, then draw the skeleton segments
plot(bx, by, ylim = c(2.5, -0.5), asp = 1, pch = 19, main = "one pose")
plt.segment(bx, by)
```

posefixer

*Detect and correct OpenPose gait anomalies***Description**

The main PoseFixeR workflow. Given imported and reshaped OpenPose keypoint trajectories, it normalises them, detects and corrects left/right leg swaps, anatomical (segment-length), biomechanical (joint-angle / range-of-motion) and physical (frame-transition) anomalies as well as low-confidence detections, sets anomalous keypoints to NA, standardises segment lengths, then imputes and smooths the trajectories.

Usage

```
posefixer(dat, race, fps)
```

Arguments

dat	Imported and reshaped OpenPose data (see <code>import_json()</code> and <code>reshaping()</code>): a list with x, y and confidence keypoint matrices (frames x keypoints).
race	Reference-population index for the anatomical proportions (1 or 2).
fps	Frames per second of the source video.

Value

The corrected data in the same list structure as dat, or NA if the recording fails the quality criteria.

References

Sugiyama Y, Uno K, Matsui Y (2023). Types of anomalies in two-dimensional video-based gait analysis in uncontrolled environments. *PLOS Computational Biology* 19:e1009989. doi:10.1371/journal.pcbi.1009989

Examples

```
## Build a small synthetic, neck-normalised gait recording (40 frames) with
## the legs swinging in anti-phase, then detect and correct anomalies.
parts <- c("nose", "neck", "Rshoulder", "Relbow", "Rwrist", "Lshoulder", "Lelbow",
  "Lwrist", "Rhip", "Rknee", "Rankle", "Lhip", "Lknee", "Lankle", "Reye", "Leye",
  "Rear", "Lear")
bx <- c(0, 0, -.25, -.3, -.32, .25, .3, .32, -.18, -.2, -.22, .18, .2, .22, -.05, .05, -.1, .1)
by <- c(-.3, 0, .05, .5, .95, .05, .5, .95, 1, 1.65, 2.28, 1, 1.65, 2.28, -.35, -.35, -.3, -.3)
n <- 40
swing <- 0.45 * sin(seq(0, 4 * pi, length.out = n))
x <- matrix(bx, n, 18, byrow = TRUE, dimnames = list(NULL, parts))
y <- matrix(by, n, 18, byrow = TRUE, dimnames = list(NULL, parts))
x[, "Rankle"] <- bx[11] - swing
x[, "Lankle"] <- bx[14] + swing
x[, "Rknee"] <- bx[10] - 0.5 * swing
```

```
x[, "Lknee"] <- bx[13] + 0.5 * swing
dat <- list(x = x, y = y, p = matrix(1, n, 18, dimnames = list(NULL, parts)))
fixed <- posefixer(dat, race = 1, fps = 25)
str(fixed, max.level = 1) # corrected x / y / p trajectories
```

reshaping

*Reshape imported keypoints into frame-by-keypoint matrices***Description**

Reshape imported keypoints into frame-by-keypoint matrices

Usage

```
reshaping(obj, norm = TRUE)
```

Arguments

obj A list of per-frame keypoint matrices (from [import_json\(\)](#)).
norm Logical; normalise coordinates relative to the neck (default TRUE).

Value

A list with x, y and confidence matrices (frames x keypoints).

Examples

```
## five frames, each an 18 x 3 keypoint matrix as returned by import_json()
parts <- c("nose", "neck", "Rshoulder", "Relbow", "Rwrist", "Lshoulder", "Lelbow",
  "Lwrist", "Rhip", "Rknee", "Rankle", "Lhip", "Lknee", "Lankle", "Reye", "Leye",
  "Rear", "Lear")
bx <- c(0,0,-.25,-.3,-.32,.25,.3,.32,-.18,-.2,-.22,.18,.2,.22,-.05,.05,-.1,.1)
by <- c(-.3,0,.05,.5,.95,.05,.5,.95,1,1.65,2.28,1,1.65,2.28,-.35,-.35,-.3,-.3)
frames <- lapply(1:5, function(i) {
  m <- cbind(X = bx, Y = by, P = 1)
  rownames(m) <- parts
  m
})
resh <- reshaping(frames)
dim(resh$x) # 5 frames x 18 keypoints
colnames(resh$x)[1:4]
```

Index

anomaly, [2](#)

diplay_frames, [3](#)

distance, [3](#)

graphics::par(), [3](#)

hdegree, [4](#)

import_json, [5](#)

import_json(), [8](#), [9](#)

kdegree, [6](#)

normalize, [6](#)

plt.segment, [7](#)

plt.segment(), [3](#)

posefixer, [8](#)

reshaping, [9](#)

reshaping(), [3](#), [4](#), [6–8](#)