interlimb

an R Package for Assessing Inter-Limb Asymmetries

Aaron Pearson*

Abstract

Inter-limb asymmetries relate to increased incidence of injury and decreased athletic performance. Current methods of assessing inter-limb asymmetries requires custom MS Excel Spreadsheets. The *interlimb* R package provides an open-source alternative to assessing inter-limb asymmetry scores. This paper briefly explores inter-limb asymmetry mathematical models and their utilization, and outlines how users can generate asymmetry scores and reliability measures nearly instantaneously. Advanced users can extend the *interlimb* package by building data sets containing asymmetry scores from multiple indexes. Finally, users are shown how to track and plot changes in asymmetry scores relative to a baseline value and to the previous assessment.

Introduction

Inter-limb asymmetries, or inter-limb imbalances, of the lower-body greater than 10-15% are associated with increased incidences of injury Dos'Santos et al., 2019; Bishop et al., 2017; Tyler et al., 2001). Athletes displaying inter-limb asymmetries also experience decrements in repeated sprint and jumping ability, and general athletic performance (Bishop et al., 2021; Bishop et al., 2018b).

Assessing asymmetries is simple but understanding them requires contextualization. Individuals may present symmetrical power output during unilateral vertical jumping tests and asymmetrically when jumping horizontally. It is also documented that the preferred limb has greater task completion abilities approximately half the time. For example, an individual prefers jumping from their left limb yet display greater force outputs when jumping off the right limb (Bishop, et al., 2018b; Virgile & Bishop, 2021). Further uncertainty exists when selecting the most appropriate symmetry or asymmetry calculation.

Currently, calculating inter-limb asymmetries requires custom MS Excel spreadsheets (Bishop et al., 2018a; Bishop et al., 2016). The *interlimb* R Stats package

*School of Kinesiology and Exercise Science, Faculty of Medecine, University of Montreal

(v4.2.1; R Core Team 2022) is an open-source tool that helps researchers, rehabilitation specialists, and sports scientists effortlessly assess limb imbalances. Regardless of the test,balance, or power, *interlimb* returns multiple symmetry and asymmetry indexes, reliability measures, and changes in asymmetry values nearly instantaneously.

Below are brief discussions on the mathematical modelling processes and how users can install and utilize the package. The latter portion of the publication provides users insight into the intermediary calculations which extends the package's functionality.

Mathematical Models

The symmetry and asymmetry indexes in this package, henceforth referred to as asymmetry indexes, were explored by Bishop and colleagues (2016; 2018a) and Parkinson and colleagues (2021). Although asymmetry index names are not universally accepted, *interlimb* adopts the majority of the names that Bishop and colleagues (2018a) introduced. The remaining naming conventions try to follow those outlined by Parkinson and colleagues (2021).

The indexes typically rely on identifying limbs as either dominant or non-dominant, involved or uninvolved, and weak or strong. Identifying the limbs in this manner can introduce confusion when assessing inter-limb asymmetries (Bailey et al., 2021; Bishop, Read, et al., 2018; Parkinson et al., 2021). For example, limbs designated as strong or weak does not provide any indication as to which limb, left or right, is stronger. Moreover, when tracking magnitudes of asymmetry, it is impractical to assume that the stronger limb remains the same over time. Designating limbs as dominant and non-dominant faces similar issues. Since asymmetries are context dependent, the dominant limb during one test may not correspond with it being the dominant limb in another. Therefore, it is the author's suggestion that users assign limbs as left and right.

Finally, it is important to discuss asymmetry directionality. Symmetry and asymmetry calculations are returned as either a scalar or vector (Parkinson, 2021). Scalar values provide the magnitude of difference between limbs without indicating which limb is stronger. Conversely, vectorized values are returned as either positive or negative, or above and below a reference value which helps researchers and practitioners understand whether the left or right, dominant or non-dominant, etc. is stronger.

Index Selection

Users can opt to use the package's primary function, `interlimb()`, which selects the most-appropriate asymmetry index, depending on testing modality (Bailey et al., 2021; Bishop et al., 2016). Advanced users, and those familiar with inter-limb asymmetry research, can override the default selection via the `asym.algo` argument. The *Extending interlimb* section also outlines how users can return asymmetry values for the 16 indexes provided in the package.

Below discusses the the selection of asymmetry indexes depending on the testing modality, and provides a brief overview of the remaining indexes.

Bilateral Indexes

Bilateral indexes are calculated as a ratio of the difference between limbs and a reference value. Typically, the reference value is the summed output of both limbs but can also be referenced against a single limb. Bishop and colleagues (2018a) suggest that practitioners assess bilateral assessments using the *Bilateral Asymmetry Index 1* or the *Symmetry Index*.

The *Bilateral Asymmetry Index 1* is calculated as:

$$asym = \frac{dl - ndl}{dl + ndl} \times\ 100$$

where the dominant leg (dl) and non-dominant leg (ndl) are identified, and the asymmetry score ($asym$) is returned as a vectorized percentage. A resulting positive value indicates that the dominant leg is stronger or more powerful. Whereas, a negative value favors the non-dominant leg. The value returned is the difference in output between dominant and non-dominant legs as a percentage of the summed output.

The *Symmetry Index* is calculated as:

$$asym = \frac{strong - weak}{strong + weak} \times\ 100$$

where *strong* and *weak* correspond to the stronger and weaker limbs, respectively. The asymmetry value is returned as a scalar percentage which does not provide practitioners with insights as to which limb is stronger.

Both the *Bilateral Asymmetry Index 1* and *Symmetry Index* assign 0% as perfectly symmetrical and a magnitude of difference of 100% as perfectly asymmetrical.

Although the *Bilateral Asymmetry Index 1* and *Symmetry Index* return the same magnitude of asymmetry, the *Bilateral Asymmetry Index 1* is preferred because it provides directionality.

Unilateral Indexes

Bishop and colleagues (2016, 2018a) advocate that unilateral asymmetry assessments utilize the *Percentage Difference* and *Bilateral Strength Asymmetry* indexes.

Both the *Percentage Difference* and *Bilateral Strength Asymmetry* are simple calculations that assesses asymmetry as a ratio between the limbs. The difference between the indexes are semantic; the limbs are identified as either *high* and *low*, or *strong* and *weak*. For example, the *Percentage Difference* is calculated as:

$$asym = \left[\left(\frac{100}{high} \times \ low\right) \times \ -1\right] + 100$$

where *high* and *low* correspond to the greater and lower values, respectively. The *Bilateral Strength Asymmetry* is the same formula where *high* and *low* are defined as the stronger and weaker limbs, respectively.

The *Percentage Difference* and *Bilateral Strength Asymmetry* assign 0% as perfectly symmetrical and 100% as perfectly asymmetrical. It should be noted that the unilateral asymmetry indexes are bound between 0% and 100%.

The asymmetry is returned as a scalar which does not provide practitioners with asymmetry directionality. Therefore, users must retain which limb is stronger or more powerful to better contextualize the results.

Other Indexes

interlimb returns several other symmetry and asymmetry calculations. These indexes were included in the package because they are prominent within the literature. It is suggested that users utilize the *Bilateral Asymmetry Index 1* and the *Percentage Difference* for bilateral and unilateral assessment, respectively. Users that are familiar with the literature may opt for different calculations, depending on their population and assessment modality.

Table 1: Indices Included in the interlimb Package

Asymmetry Index	Formula	Provides Directionality	Function Name
Asymmetry Index 1	$\left(\frac{dl-ndl}{(dl+ndl)/2}\right)\times\ 100$	Context Dependent	`asym.index.1`
Asymmetry Index 2	$\left(-\ \frac{100}{s}\times\ w\right)+100$	No	`asym.index.2`
Bilateral Asymmetry Index 1	$\left(\frac{dl-ndl}{dl+ndl}\right)\times\ 100$	Context Dependent	`bilat.asym.index.1`
Bilateral Asymmetry Index 2	$2\times\ \left(\frac{dl-ndl}{dl+ndl}\right)\times\ 100$	Context Dependent	`bilat.asym.index.2`
Bilateral Strength Asymmetry 1	$\left(\frac{s-w}{s}\right)\times\ 100$	No	`bilat.strength.asym.1`
Bilateral Strength Asymmetry 2	$\left(\frac{w-s}{s}\right)\times\ 100$	No	`bilat.strength.asym.2`
Between Limb Imbalance	$\log\left(\frac{l}{r}\right)\times\ 100$	Yes	`btwn.limb.imbalance`
Limb Symmetry Index 1	$\left(\frac{ndl}{dl}\right)\times\ 100$	Context Dependent	`limb.sym.index.1`
Limb Symmetry Index 2	$\left(1-\frac{ndl}{dl}\right)\times\ 100$	Context Dependent	`limb.sym.index.2`
Limb Symmetry Index 3	$\left(\frac{r-l}{(r+l)/2}\right)\times\ 100$	Yes	`limb.sym.index.3`
Limb Symmetry Index 4	$\left(\frac{dl}{ndl}\right)\times\ 100$	Context Dependent	`limb.sym.index.4`
Percent Difference	$100+\left(-\ \frac{100}{s}\times\ w\right)$	No	`percent.diff`
Strength Asymmetry	$100-\left(\frac{ndl}{dl}\times\ 100\right)$	Context Dependent	`strength.asym`
Symmetry Angle	$\left[2\times\ \frac{\pi/4-\tan^{-1}\left(\frac{l}{r}\right)}{\pi}\right]\times\ 100$	Yes	`sym.angle`
Symmetry Index	$\left(\frac{l-r}{l+r}\right)\times\ 100$	Yes	`sym.index`
Symmetry Index Vectorized	$\left(\frac{s-w}{s+w}\right)\times\ 100$	No	`sym.index.vect`

Note: Adapted from Bishop and colleagues (2018a; 2016) and Parkinson and colleagues (2021); Symmetry Angle is in radians

The *interlimb* Package

interlimb is written with minimal dependencies and requires only the *reshape2* package to complete asymmetry assessments. If users do not already have *reshape2* on their machine, it will download and install automatically when installing the *interlimb* package.

Install and Load *interlimb*

interlimb is not yet published on CRAN. As such, users must install it directly from its GitHub repository.

Installing *interlimb*

```
devtools::install_github("aaronzpearson/interlimb")
```

Loading the package

```
library(interlimb)
```

Workflow

The examples utilize the data set below. The participant's abilities per limb, X, are normally distributed:

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

with mean, $\mu = 20$, and variance, $\sigma^2 = 1$. They performed three trials per week for five weeks. Finally, the participant reports that their right limb is the dominant limb. See Appendix 1 for the code to reproduce the sample data.

The package's primary function, `interlimb()`, returns a `data.frame` with values that are grouped by date and include the individual's: parameter tested, best or mean of their greatest `n` trials per limb, stronger limb for the given assessment, the selected asymmetry index , standard deviation of the asymmetry calculations for all trials if `n` is greater than 1, coefficient of variation for all asymmetry values if `n` is greater than 1, rolling average coefficient of variation from week to week, and changes in magnitudes of asymmetry in reference to the first and previous tests.

Table 2: Example Data Set

test.date	trial	left	right
Week 1	1	19.10	17.69
Week 1	2	20.18	20.88
Week 1	3	21.59	20.04
Week 2	1	18.87	21.01
Week 2	2	19.92	20.43
Week 2	3	20.13	22.09
Week 3	1	20.71	18.80
Week 3	2	19.76	21.59
Week 3	3	21.98	21.95
Week 4	1	19.86	20.00
Week 4	2	20.42	17.55
Week 4	3	20.98	20.48
Week 5	1	19.61	19.40
Week 5	2	18.96	20.79
Week 5	3	21.78	20.29

`interlimb()` takes on the following arguments:

- `test.data`: initial data set
- `test.dates`: column name for the test dates
- `dominant limb`: the dominant limb as either `"r"` or `"l"`
- `parameter`: either the parameter tested or can be left blank
- `right.limb`: column name for the right limb
- `left.limb`: column name for the left limb
- `n.limbs`: set as 1 for unilateral tests or 2 for bilateral tests
- `best.n`: evaluate the best `n` trials per date, set as either a positive integer or `"all"`
- `asym.algo`: asymmetry index to use, names following those in Table 1 or `"auto"`
- `na.fill`: if `TRUE`, fills missing asymmetry values matching those from the previous week and assumes that the magnitude of asymmetry hasn't changed

Assessing Unilateral and Bilateral Tests

Users have the ability to assess asymmetry values using either a specified asymmetry index or automatically. The `asym.algo` argument takes on either asymmetry index values that are represented in Table 1 or `"auto"`. When set to `"auto"`, the `interlimb()` function selects the asymmetry index based on the `n.limbs` argument. When `n.limbs = 1`, *Percent Difference* is selected whereas, when `n.limbs = 2`, the *Bilateral Asymmetry Index 1* is selected.

```
interlimb(
  test.data = asymmetry.tests,
  test.dates = "test.date",
  dominant.limb = "r",
  parameter = "example",
  right.limb = "right",
  left.limb = "left",
```

When testing participants unilaterally:

```
  n.limbs = 1,
```

and when testing participants bilaterally:

```
  n.limbs = 2,
```

followed by:

```
  best.n = 3,
  asym.algo = "auto",
  na.fill = TRUE
  )
```

Mean vs Best Trial Selection

Selecting either the best or mean of the best n trials depends on the `best.n` argument. `best.n` can take on either a positive integer,

$$\mathbb{Z}^+ = \{1, 2, 3, ...\}$$

or `"all"`. Opting for `"all"` provides the advantage that the number of trials per week can be inconsistent. This is crucial when trials must be excluded from being analyzed due to tester or participant error.

The code below illustrates how users can select either the best, mean of the best 3, or all trials:

```
interlimb(
  test.data = asymmetry.tests,
  test.dates = "test.date",
  dominant.limb = "r",
  parameter = "example",
  right.limb = "right",
  left.limb = "left",
  n.limbs = 1,
```

When selecting the best trial:

```
  best.n = 1,
```

when selecting the mean of the best 3 trials:

```
  best.n = 3,
```

and when selecting all trials:

```
  best = "all"
```

followed by

```
  asym.algo = "auto",
  na.fill = TRUE
  )
```

See Appendix 2 for the code to generate Tables 3, 4, and 5

Table 3: Assessing Inter-limb Asymmetries from the Best Unilateral Trial

test.date	left.limb	right.limb	parameter	asymmetry.index	dominant.limb	stronger.limb	asymmetry	asym.cv	asym.change	rel.change
Week 1	21.59	20.88	example	percent.diff	r	l	3.29	0.00	0.00	0.00
Week 2	20.13	22.09	example	percent.diff	r	r	8.87	64.90	5.58	5.58
Week 3	21.98	21.95	example	percent.diff	r	l	0.14	107.83	-8.73	-3.15
Week 4	20.98	20.48	example	percent.diff	r	l	2.38	101.11	2.24	-0.91
Week 5	21.78	20.79	example	percent.diff	r	l	4.55	84.18	2.17	1.26

Note: `n.limbs = 1`, `best.n = 1`, and `asym.algo = "auto"`

Table 4: Assessing Inter-limb Asymmetries from the Best 3 Unilateral Trials

test.date	left.limb	right.limb	parameter	asymmetry.index	dominant.limb	stronger.limb	asymmetry	sd	daily.cv	asym.cv	asym.change	rel.change
Week 1	20.29	19.54	example	percent.diff	r	l	3.71	2.27	38.04	0.00	0.00	0.00
Week 2	19.64	21.18	example	percent.diff	r	r	7.26	4.11	57.22	45.77	3.55	3.55
Week 3	20.82	20.78	example	percent.diff	r	l	0.18	5.04	84.79	95.25	-7.08	-3.53
Week 4	20.42	19.34	example	percent.diff	r	l	5.27	7.27	127.34	72.91	5.09	1.56
Week 5	20.12	20.16	example	percent.diff	r	r	0.21	4.02	72.14	93.89	-5.06	-3.50

Note: `n.limbs = 1`, `best.n = 3`, and `asym.algo = "auto"`

Table 5: Assessing Inter-limb Asymmetries from All Bilateral Trials

test.date	left.limb	right.limb	parameter	asymmetry.index
Week 1	20.29	19.54	example	bilat.asym.index.1
Week 2	19.64	21.18	example	bilat.asym.index.1
Week 3	20.82	20.78	example	bilat.asym.index.1
Week 4	20.42	19.34	example	bilat.asym.index.1
Week 5	20.12	20.16	example	bilat.asym.index.1

dominant.limb	stronger.limb	asymmetry	sd	daily.cv
r	l	-1.89	1.20	38.90
r	r	3.76	2.19	58.37
r	l	-0.09	2.64	84.90
r	l	-2.71	3.94	129.54
r	r	0.11	2.11	72.78

asym.cv	asym.change	rel.change
0.00	0.00	0.00
46.81	5.65	5.65
95.91	-3.85	1.80
73.39	-2.62	-0.82
94.27	2.82	2.00

Note: `n.limbs = 2`, `best.n = "all"`, and `asym.algo = "auto"`

Extending *interlimb*

`interlimb.wide()` and `interlimb.long()` are supplementary functions that are called upon by the `interlimb()` function. They return a `data.frame` in the *wide* and *long* formats, respectively. Wide `data.frame`s typically return many variables whereas, long `data.frame`s return fewer variables and more observations. Using the sample data above, the wide data set returns 22 variables, each of which with 15 observations. Conversely, the long data set returns nine variables, each of which with 240 observations.

Finally, `track.asym()` accepts the long formatted `data.frame` and returns weekly changes in asymmetry scores. Users must note that `track.asym()` requires that only one asymmetry index be present in the long `data.frame` to return true values which can be set via the `asym.algo` argument.

Reliability Values

The `interlimb()` function returns the standard deviation and coefficient of variation for the inter-limb asymmetry assessments, when appropriate. This is achieved by calling `interlimb.wide()` and `interlimb.long()` twice in *base R*. Since the *base R* is not concise, and for the sake of brevity, the examples below omit the reliability values. Appendix 3 illustrates how to concisely generate reliability values via *Tidyverse* functions. This method was not included in the *interlimb* package to minimize dependencies.

Wide and Long Data Sets

The examples below utilize the sample data generated earlier, and returns asymmetry scores for all trials by a right-limb dominant participant.

`interlimb.wide()`

`interlimb.wide()` takes on the following arguments:

- `test.data`: initial data set
- `test.dates`: column name for the test dates
- `dominant limb`: the dominant limb as either `"r"` or `"l"`
- `parameter`: either the parameter tested or can be left blank
- `right.limb`: column name for the right limb
- `left.limb`: column name for the left limb

```
wide.asym <- interlimb.wide(test.data = asymmetry.tests,
                            test.dates = "test.date",
                            dominant.limb = "r",
                            parameter = "example",
                            right.limb = "right",
                            left.limb = "left")
```

interlimb.long()

`interlimb.long()` takes on only one argument:

- `data.set`: wide format `data.frame` that was returned by `interlimb.wide()`

```
long.asym <- interlimb.long(data.set = wide.asym)
```

track.asym()

Tracking weekly asymmetries scores is accomplished via the `track.asym()` function. The `track.asym()` function generates three variables: `asym.change`, `rel.change` and `rolling.cv`. When assessing magnitudes of change in asymmetries, `asym.change` and `rel.change` should be prioritized.

It should be noted that `asym.change` assesses the magnitude of asymmetry score change from the previous week. `rel.change` assesses the magnitude of asymmetry score change against the first asymmetry score value.

`track.asym()` takes on three arguments:

- `data.set`: long format `data.frame` that was returned by `interlimb.long()`
- `asym.algo`: asymmetry index to use, names following those in Table 1
- `na.fill`: if `TRUE`, fills missing asymmetry values equal to those from the previous week and assumes the magnitude of asymmetry hasn't changed

Unlike the `interlimb()` function, `asym.algo` is not set automatically. Therefore, it will return an error if it is not stated explicitly.

```
changes.in.asym <- track.asym(data.set = long.asym,
                              asym.algo = "percent.diff",
                              na.fill = TRUE)
```

The example below sets `asym.algo = "percent.diff"` and only utilizes the *first* trial per week.

Table 6: Wide Inter-Limb Asymmetry Data Frame

test.date	parameter	right.limb	left.limb	dominant.limb	stronger.limb
Week 1	example	17.69	19.10	r	l
Week 1	example	20.88	20.18	r	r
Week 1	example	20.04	21.59	r	l

asym.index.1	asym.index.2	bilat.asym.index.1	bilat.asym.index.2
-7.67	7.38	-3.83	-7.67
3.41	3.35	1.70	3.41
-7.45	7.18	-3.72	-7.45

bilat.strength.asym.1	bilat.strength.asym.2	btwn.limb.imbalance	limb.sym.index.1
7.38	-7.38	7.67	107.97
3.35	-3.35	-3.41	96.65
7.18	-7.18	7.45	107.73

limb.sym.index.2	limb.sym.index.3	limb.sym.index.4	percent.diff
-7.97	-7.67	92.62	7.38
3.35	3.41	103.47	3.35
-7.73	-7.45	92.82	7.18

strength.asym	sym.angle	sym.index	sym.index.vect
-7.97	-2.44	3.83	3.83
3.35	1.09	1.70	-1.70
-7.73	-2.37	3.72	3.72

Table 7: Long Inter-Limb Asymmetry Data Frame

test.date	parameter	left.limb	right.limb	dominant.limb	stronger.limb	asymmetry.index	asymmetry
Week 1	example	19.10	17.69	r	l	asym.index.1	-7.67
Week 2	example	18.87	21.01	r	r	asym.index.1	10.73
Week 3	example	20.71	18.80	r	l	asym.index.1	-9.67
Week 4	example	19.86	20.00	r	r	asym.index.1	0.70
Week 5	example	19.61	19.40	r	l	asym.index.1	-1.08
Week 1	example	19.10	17.69	r	l	asym.index.2	7.38
Week 2	example	18.87	21.01	r	r	asym.index.2	10.19
Week 3	example	20.71	18.80	r	l	asym.index.2	9.22
Week 4	example	19.86	20.00	r	r	asym.index.2	0.70
Week 5	example	19.61	19.40	r	l	asym.index.2	1.07
Week 1	example	19.10	17.69	r	l	bilat.asym.index.1	-3.83
Week 2	example	18.87	21.01	r	r	bilat.asym.index.1	5.37
Week 3	example	20.71	18.80	r	l	bilat.asym.index.1	-4.83
Week 4	example	19.86	20.00	r	r	bilat.asym.index.1	0.35
Week 5	example	19.61	19.40	r	l	bilat.asym.index.1	-0.54
Week 1	example	19.10	17.69	r	l	bilat.asym.index.2	-7.67
Week 2	example	18.87	21.01	r	r	bilat.asym.index.2	10.73
Week 3	example	20.71	18.80	r	l	bilat.asym.index.2	-9.67
Week 4	example	19.86	20.00	r	r	bilat.asym.index.2	0.70
Week 5	example	19.61	19.40	r	l	bilat.asym.index.2	-1.08
Week 1	example	19.10	17.69	r	l	bilat.strength.asym.1	7.38
Week 2	example	18.87	21.01	r	r	bilat.strength.asym.1	10.19
Week 3	example	20.71	18.80	r	l	bilat.strength.asym.1	9.22
Week 4	example	19.86	20.00	r	r	bilat.strength.asym.1	0.70
Week 5	example	19.61	19.40	r	l	bilat.strength.asym.1	1.07
Week 1	example	19.10	17.69	r	l	bilat.strength.asym.2	-7.38
Week 2	example	18.87	21.01	r	r	bilat.strength.asym.2	-10.19
Week 3	example	20.71	18.80	r	l	bilat.strength.asym.2	-9.22
Week 4	example	19.86	20.00	r	r	bilat.strength.asym.2	-0.70
Week 5	example	19.61	19.40	r	l	bilat.strength.asym.2	-1.07

Only the first 30 rows are printed

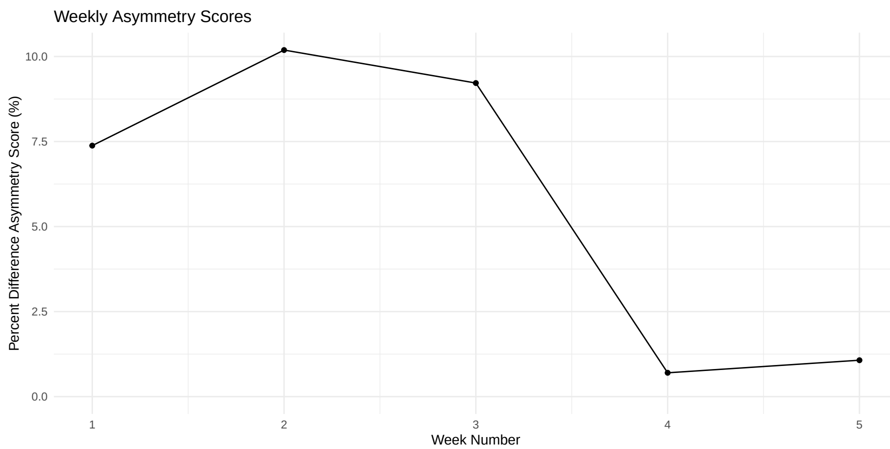
Table 8: Weekly Changes in Asymmetry Score

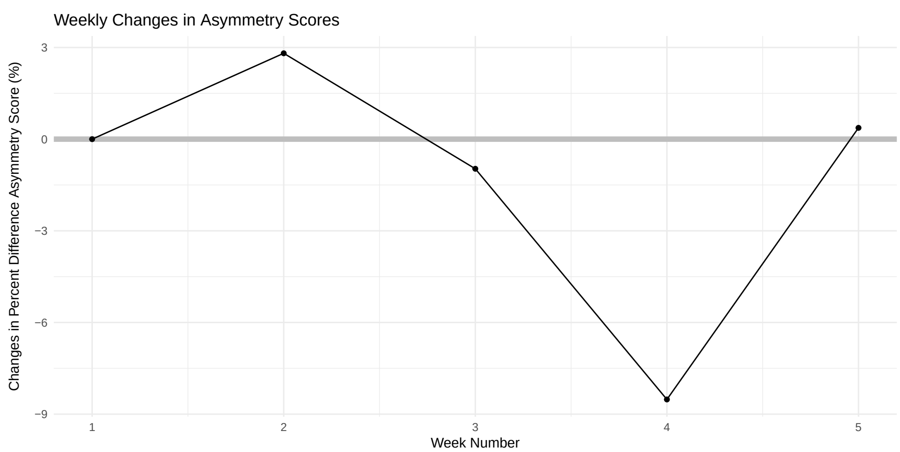
test.date	parameter	left.limb	right.limb	dominant.limb	stronger.limb	asymmetry.index	asymmetry	asym.cv	asym.change	rel.change
Week 1	example	19.10	17.69	r	l	percent.diff	7.38	0.00	0.00	0.00
Week 2	example	18.87	21.01	r	r	percent.diff	10.19	22.62	2.81	2.81
Week 3	example	20.71	18.80	r	l	percent.diff	9.22	15.98	-0.97	1.84
Week 4	example	19.86	20.00	r	r	percent.diff	0.70	62.23	-8.52	-6.68
Week 5	example	19.61	19.40	r	l	percent.diff	1.07	79.17	0.37	-6.31

Plotting Results

Below is a simple example of how users can plot week to week asymmetries and changes in asymmetries. The code to generate the plots is found in Appendix 4.

Note that a change of inter-limb asymmetry score of 0% suggests that there was no change in the magnitude of the individual's asymmetry.





Supplemental Readings

Below are selected publications for users wishing to gain greater information on inter-limb asymmetries and their utilization.

Asymmetries of the Lower Limb: The Calculation Conundrum in Strength Training and Conditioning [Link]

Inter-Limb Asymmetries: Understanding how to Calculate Differences From Bilateral and Unilateral Tests [Link]

A Technical Report on Reliability Measurement in Asymmetry Studies [Link]

Inter-Limb Asymmetries: Are Thresholds a Usable Concept? [Link]

Assessing Inter-Limb Asymmetries: Are We Heading in the Right Direction? [Link]

The Calculation, Thresholds and Reporting of Inter-Limb Strength Asymmetry: A Systematic Review [Link]

References

Bailey, C. A., Sato, K., & McInnis, T. C. (2021). A Technical Report on Reliability Measurement in Asymmetry Studies. Journal of Strength and Conditioning Research, Publish Ahead of Print. https://doi.org/10.1519/JSC.0000000000004024

Bishop, C., Read, P., Chavda, S., & Turner, A. (2016). Asymmetries of the Lower Limb: The Calculation Conundrum in Strength Training and Conditioning. Strength & Conditioning Journal, 38(6), 27–32. https://doi.org/10.1519/SSC.0000000000000264

Bishop, C., Read, P., Lake, J., Chavda, S., & Turner, A. (2018a). Interlimb Asymmetries: Understanding How to Calculate Differences From Bilateral and Unilateral Tests. Strength & Conditioning Journal, 40(4), 1–6. https://doi.org/10.1519/SSC.0000000000000371

Bishop C, Turner A, Read P. Effects of inter-limb asymmetries on physical and sports performance: a systematic review. Journal of Sports Sciences. 2018;36(10):1135-1144. doi:10.1080/02640414.2017.1361894

Bishop, C., Turner, A., & Read, P. (2018b). Effects of inter-limb asymmetries on physical and sports performance: A systematic review. Journal of Sports Sciences, 36(10), 1135–1144. https://doi.org/10.1080/02640414.2017.1361894

Bishop, C. (2021). Interlimb Asymmetries: Are Thresholds a Usable Concept? Strength & Conditioning Journal, 43(1), 32–36. https://doi.org/10.1519/SSC.0000000000000554

Dos'Santos, T., Bishop, C., Thomas, C., Comfort, P., & Jones, P. A. (2019). The effect of limb dominance on change of direction biomechanics: A systematic review of its importance for injury risk. Physical Therapy in Sport, 37, 179–189. https://doi.org/10.1016/j.ptsp.2019.04.005

Dos'Santos, Thomas & Thomas, Christopher & Jones, Paul. (2020). Assessing Interlimb Asymmetries: Are We Heading in the Right Direction?. Strength and conditioning journal. Publish Ahead of Print. 10.1519/SSC.0000000000000590

Parkinson, A. O., Apps, C. L., Morris, J. G., Barnett, C. T., & Lewis, M. G. C. (2021). The Calculation, Thresholds and Reporting of Inter-Limb Strength Asymmetry: A Systematic Review. Journal of Sports Science and Medicine, 594–617. https://doi.org/10.52082/jssm.2021.594

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for statistical Computing, Vienna, Austria. URL https://www.R-project.org/

Tyler, T. F., Nicholas, S. J., Campbell, R. J., & McHugh, M. P. (2001). The Association of Hip Strength and Flexibility with the Incidence of Adductor Muscle Strains in Professional Ice Hockey Players. The American Journal of Sports Medicine https://doi.org/10.1177/03635465010290020301

Virgile, A., & Bishop, C. (2021). A Narrative Review of Limb Dominance: Task Specificity and the Importance of Fitness Testing. Journal of Strength and Conditioning Research, 35(3), 846–858. https://doi.org/10.1519/JSC.0000000000003851

Appendix 1: Generating Sample Data

```
set.seed(2)

asymmetry.tests <- data.frame(
  test.date = rep(c("Week 1",
                    "Week 2",
                    "Week 3",
                    "Week 4",
                    "Week 5"),
                  each = 3),
  trial = rep(1:3, times = 5),
  left = round(rnorm(15, mean = 20, 1),2),
  right = round(rnorm(15, mean = 20, 1), 2)
)
```

Appendix 2: Generating Tables 3, 4, and 5

Table 3: `n.limbs = 1`, `best.n = 1`, and `asym.algo = "auto"`

```
unilateral.asym.best <- interlimb(
  test.data = asymmetry.tests,
  test.dates = "test.date",
  dominant.limb = "r",
  parameter = "example",
  right.limb = "right",
  left.limb = "left",
  n.limbs = 1,
  best.n = 1,
  asym.algo = "auto",
  na.fill = TRUE
  )

unilateral.asym.best
```

Table 4: `n.limbs = 1`, `best.n = 3`, and `asym.algo = "auto"`

```
unilateral.asym.mean<- interlimb(
  test.data = asymmetry.tests,
  test.dates = "test.date",
  dominant.limb = "r",
  parameter = "example",
  right.limb = "right",
  left.limb = "left",
  n.limbs = 1,
  best.n = 3,
  asym.algo = "auto",
  na.fill = TRUE
  )

unilateral.asym.mean
```

Table 5: `n.limbs = 2`, `best.n = "all"`, and `asym.algo = "auto"`

```
bilat.asym.mean <- interlimb(
  test.data = asymmetry.tests,
  test.dates = "test.date",
  dominant.limb = "r",
  parameter = "example",
  right.limb = "right",
  left.limb = "left",
  n.limbs = 2,
  best.n = "all",
  asym.algo = "auto",
  na.fill = TRUE
  )

bilat.asym.mean
```

Appendix 3: Calculating Reliability Values

To maintain tidy code, the code below utilizes *Tidyverse* functions `group_by()`, `arrange()`, `slice()`, `summarize()`, `inner_join()`, and `mutate()`. These functions are piped together using `%>%` which provides the ability to perform multiple data transformation steps sequentially within the same code chunk.

Users who prefer working in base R can find the equivalent code in the *inter-limb.R* file.

Generate Mean Left and Right Limb Values

```
right <- asymmetry.tests %>%
  group_by(test.date) %>%
  arrange(., desc(right),
          .by_group = TRUE) %>%
  slice(1:3) %>%
  summarize(right.limb = mean(right))

left <- asymmetry.tests %>%
  group_by(test.date) %>%
  arrange(., desc(left),
          .by_group = TRUE) %>%
  slice(1:3) %>%
  summarize(left.limb = mean(left))

grouped.asym <- inner_join(left, right) %>%
  mutate(parameter = "example")
```

Calculate Asymmetry Scores & Reliability Values

If users know which asymmetry index that they want to analyze further, they can calculate the inter-limb asymmetry standard deviation and coefficient of variation. To do so, they must:

1. calculate asymmetry scores per trial
2. group the data by date
3. summarize the data by calling `mean()` and `sd()`
4. call `mutate()` to build the coefficient of variation variable

The example below assumes that the user wants to calculate reliability values for the `percent.diff` asymmetry index.

```
reliability.asym <- asymmetry.tests %>%
  mutate(percent.diff = interlimb::percent.diff(l = left,
                                                 r = right)) %>%
  group_by(test.date) %>%
  summarize(asymmetry = mean(percent.diff),
            sd = sd(percent.diff)) %>%
  mutate(cv = sd/asymmetry * 100) %>%
  select(-asymmetry)
```

Final Output

Finally, to build a `data.frame` that is compatible with `track.asym`, `long.asym` (from above) must be joined to the `reliability.asym` `data.frame`. Subsequently, the `asymmetry.index` variable must be filtered for `"percent.diff"`.

The table below nearly mirrors that that was created by the `interlimb()` function when `asym.algo == "percent.diff"`. By passing `percent.diff.asym` to `tack.asym()`, the output will match that in table 8.

```
percent.diff.asym <- long.asym %>%
  left_join(.,
            reliability.asym,
            by = c("test.date")) %>%
  filter(asymmetry.index == "percent.diff") %>%
  mutate_if(is.double, round, 2) # optional
```

Appendix 4: Tracking Asymmetry Plot Code

Weekly Asymmetry Scores

```
require(ggplot2)

theme_set(theme_minimal()) # set the theme

ggplot(changes.in.asym %>%
         mutate(test.date = 1:nrow(.)),
         aes(x = test.date, y = asymmetry)) +
  geom_point(size = 1.5) +
  geom_line() +
  ggtitle("Weekly Asymmetry Scores") +
  xlab("Week Number") +
  ylab("Percent Difference Asymmetry Score (%)")
```

Changes in Weekly Assymetry Scores

```
ggplot(changes.in.asym %>%
         mutate(test.date = 1:nrow(.)),
         aes(x = test.date, y = asym.change)) +
  geom_hline(yintercept = 0, colour = "grey", size = 2) +
  geom_point(size = 1.5) +
  geom_line() +
  ggtitle("Changes in Weekly Asymmetry Scores") +
  xlab("Week Number") +
  ylab("Changes in Percent Difference Asymmetry Score (%)")
```

Appendix 5: Converting Scalar to Vector

The *interlimb* package provides users the ability to track the stronger limb via the `stronger.limb` variable in the final output.

Recent research states that practitioners can also convert scalar values to vectorized values by multiplying the asymmetry score by -1 if the left limb is stronger. This is achieved in MS Excel usign the IF function. Below, the equivalent is possible utilizing an `ifelse()` function.

The code below utilizes the `unilateral.asym.best` `data.frame` that was produced in the *Mean vs Best Trial Selection* section. This `data.frame` was chosen because the asymmetry index that was selected automatically is the `percent.diff` which returns values as a scalar.

A similar approach can be taken when assessing changes in asymmetry scores. The accompanying table is not printed.

```
unilateral.asym.best.vector <- unilateral.asym.best
unilateral.asym.best.vector$asymmetry <- ifelse(unilateral.asym.best.vector$stronger.limb == "l",
                                                unilateral.asym.best.vector$asymmetry * -1,
                                                unilateral.asym.best.vector$asymmetry)
```

Table 9: Comparison of Scalar and Vectorized percent.diff Scores

left.limb	right.limb	asymmetry.index	stronger.limb	asymmetry
21.59	20.88	percent.diff	l	3.29
20.13	22.09	percent.diff	r	8.87
21.98	21.95	percent.diff	l	0.14
20.98	20.48	percent.diff	l	2.38
21.78	20.79	percent.diff	l	4.55

left.limb	right.limb	asymmetry.index	stronger.limb	asymmetry
21.59	20.88	percent.diff	l	-3.29
20.13	22.09	percent.diff	r	8.87
21.98	21.95	percent.diff	l	-0.14
20.98	20.48	percent.diff	l	-2.38
21.78	20.79	percent.diff	l	-4.55

Note: The following variables are printed: `left.limb`, `right.limb`, `asymmetry.index`, `stronger.limb`, `asymmetry`