# Package 'weaveDraft'

November 8, 2026

**Type** Package

**Title** Structural Analysis of Woven Textile Drafts from WIF Files

**Version** 0.1.0

**Description** Imports weaving drafts stored in the Weaving Information File (WIF) interchange format, reconstructs the binary warp and weft drawdown from either a liftplan or a tie-up with treadling, and computes structural descriptors of the cloth: float length distributions, interlacement frequency, weave repeat size, a directional float ratio, and thread colour sequences. A flat-colour simulation of the fabric face helps to separate patterns carried by the interlacement structure from patterns carried by the thread colours. Intended for computational analysis and documentation of handloom and other multi-shaft textiles.

**License** MIT + file LICENSE

**Depends** R (>= 3.6.0)

**Imports** graphics, grDevices, stats, tools

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**NeedsCompilation** no

**Author** Mark Joshua R. Secretario [aut, cre]

**Maintainer** Mark Joshua R. Secretario <markjoshuasecretario@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-11-08 17:50:02 UTC

## Contents

as_weave_draft . . . . . 2
check_wif_consistency . . . . . 3

color_runs . . . 3
compute_drawdown . . . 4
compute_drawdown_liftplan . . . 5
directional_float_ratio . . . 6
float_lengths . . . 6
float_summary . . . 7
interlacement_frequency . . . 8
plot.weave_draft . . . 9
read_wif . . . 10
weave_draft_from_wif . . . 11
weave_metrics . . . 12
weave_repeat . . . 12

**Index** **14**

---

`as_weave_draft` *Construct a weave_draft object*

---

## Description

Wraps a binary drawdown matrix (and optional colour information) in an object of class `"weave_draft"` that the analysis and plotting functions of this package accept.

## Usage

```
as_weave_draft(drawdown, meta = list(), colors = NULL)
```

## Arguments

| | |
|---|---|
| `drawdown` | A numeric or logical matrix with only 0/1 (or `FALSE`/`TRUE`) entries and no `NA`. Rows are weft picks, columns are warp ends; 1 means the warp end is up (on the face). |
| `meta` | A named list of metadata, e.g. `name`, `source`, `draft_style`, `shafts`. |
| `colors` | `NULL`, or a list with elements `warp` (one colour per warp end) and `weft` (one colour per pick), both valid R colours, and optionally `palette`. |

## Value

An object of class `"weave_draft"`.

## Examples

```
plain <- outer(1:4, 1:4, function(i, j) (i + j) %% 2)
draft <- as_weave_draft(plain, meta = list(name = "plain weave"))
draft
```

## check_wif_consistency *Check that a WIF's two encodings agree*

### Description

Many WIF files store both a liftplan and a tie-up/treadling pair. This rebuilds the drawdown from each and compares them, which is a useful data quality check before analysis.

### Usage

```
check_wif_consistency(wif)
```

### Arguments

| | |
|---|---|
| wif | An object from read_wif (or a file path). |

### Value

A list with comparable (FALSE if the file has only one encoding), consistent (TRUE/FALSE, or NA if not comparable), and n_mismatch_picks, the number of picks whose rows differ.

### Examples

```
f <- system.file("extdata", "gingham_plain.wif", package = "weaveDraft")
check_wif_consistency(read_wif(f))
```

## color_runs *Colour runs along the warp or weft*

### Description

Many traditional patterns, including colour-and-weave designs on a plain weave structure, live in the thread colours rather than in the interlacement. This function summarises the colour sequence of the warp (or weft) as runs of identical colours.

### Usage

```
color_runs(draft, direction = c("warp", "weft"))
```

### Arguments

| | |
|---|---|
| draft | A weave_draft object that carries colour information (drafts built from a WIF with a colour table do). |
| direction | "warp" or "weft". |

## Value

A data frame with one row per run: `start` and `end` (thread numbers), `length`, `hex` (the colour), and `index` (its palette index when the colour is in the palette, else `NA`).

## Examples

```
f <- system.file("extdata", "gingham_plain.wif", package = "weaveDraft")
color_runs(weave_draft_from_wif(f), "warp")
```

---

`compute_drawdown` *Compute a drawdown from threading, tie-up and treadling*

---

## Description

Reconstructs the binary warp/weft interlacement matrix of a treadle-loom draft. Warp end $j$ is "up" (value 1, warp visible on the face) on pick $i$ when at least one of the shafts it is threaded through is raised by a treadle pressed on that pick. Internally this is the matrix product of the treadling, tie-up and threading indicator matrices.

## Usage

```
compute_drawdown(
  threading,
  treadling,
  tieup,
  n_shafts = NULL,
  rising_shed = TRUE
)
```

## Arguments

| | |
|---|---|
| `threading` | Integer vector (one shaft per warp end) or a list of integer vectors (several shafts per end allowed). Zero denotes an inactive entry. |
| `treadling` | Integer vector (one treadle per pick) or a list of integer vectors (several treadles per pick allowed). Treadle numbers below 1 mean "no treadle". |
| `tieup` | A list in which element `i` is the integer vector of shafts raised by treadle `i`. |
| `n_shafts` | Optional number of shafts. Defaults to the largest shaft number found in `threading` and `tieup`. Explicit counts must cover every referenced shaft. Inputs must contain finite integers. |
| `rising_shed` | Logical; `TRUE` (default) for a rising-shed loom. |

## Details

For a sinking-shed loom (`rising_shed = FALSE`) the listed shafts are the ones that sink, so the result is inverted.

## Value

An integer matrix with one row per weft pick and one column per warp end; 1 means warp up (on the face), 0 means weft on the face.

## Examples

```
# Plain weave: two shafts, two treadles
compute_drawdown(
  threading = rep(1:2, 4),
  treadling = rep(1:2, 4),
  tieup = list(1L, 2L)
)
```

---

`compute_drawdown_liftplan` *Compute a drawdown from threading and a liftplan*

---

## Description

Same as `compute_drawdown` for dobby-style drafts, where each pick directly lists the shafts raised (no tie-up or treadles).

## Usage

```
compute_drawdown_liftplan(
  threading,
  liftplan,
  n_shafts = NULL,
  rising_shed = TRUE
)
```

## Arguments

| | |
|---|---|
| `threading` | Integer vector (one shaft per warp end) or a list of integer vectors. |
| `liftplan` | A list with one integer vector per pick, giving the shafts raised on that pick (`integer(0)` for none). |
| `n_shafts` | Optional number of shafts. Defaults to the largest shaft number found in `threading` and `tieup`. Explicit counts must cover every referenced shaft. Inputs must contain finite integers. |
| `rising_shed` | Logical; `TRUE` (default) for a rising-shed loom. |

## Value

An integer matrix, rows = weft picks, columns = warp ends.

## Examples

```
compute_drawdown_liftplan(
  threading = c(1, 2, 3, 4),
  liftplan = list(c(1, 2), c(2, 3), c(3, 4), c(4, 1))
)
```

---

`directional_float_ratio` *Directional float ratio*

---

## Description

Mean warp float length divided by mean weft float length: a lightweight indicator of structural directionality. Values near 1 indicate a balanced structure; values far from 1 indicate warp- or weft-dominated floats. It describes the drawing only and is not a measured mechanical property.

## Usage

```
directional_float_ratio(draft, side = c("face", "back", "both"))
```

## Arguments

| | |
|---|---|
| `draft` | A `weave_draft` object. |
| `side` | `"face"`, `"back"` or `"both"`; see `float_lengths`. |

## Value

A number, or `NA` if either mean float length is undefined.

## Examples

```
f <- system.file("extdata", "diamond_liftplan.wif", package = "weaveDraft")
directional_float_ratio(weave_draft_from_wif(f))
```

---

`float_lengths` *Float length distributions*

---

## Description

Counts how many consecutive threads each thread passes over.

## Usage

```
float_lengths(draft, side = c("face", "back", "both"))
```

### Arguments

| | |
|---|---|
| `draft` | A `weave_draft` object. |
| `side` | One of `"face"`, `"back"` or `"both"`. |

### Details

A *warp float* is a run of consecutive picks over which one warp end stays up (a run of 1s down a column of the drawdown). A *weft float* is a run of consecutive warp ends over which one weft pick stays on the face (a run of 0s along a row). A length of 1 is a plain interlacement; longer floats give a looser, more lustrous, more snag-prone structure.

The `side` argument chooses which fabric side is described. On the `"face"` (default), warp floats are runs of 1s and weft floats are runs of 0s. On the `"back"` the roles of 0 and 1 swap. With `"both"` the runs of both values are pooled for each direction.

Floats are counted within the drawing only; no wrap-around is assumed at the edges.

### Value

A list with integer vectors `warp` and `weft`, the float lengths pooled over all warp ends and all picks respectively.

### Examples

```
f <- system.file("extdata", "twill_2_2.wif", package = "weaveDraft")
d <- weave_draft_from_wif(f)
table(float_lengths(d)$warp)
```

---

## `float_summary` *Mean and maximum float length*

---

### Description

Mean and maximum float length

### Usage

```
mean_float_length(
  draft,
  direction = c("warp", "weft"),
  side = c("face", "back", "both")
)

max_float_length(
  draft,
  direction = c("warp", "weft"),
  side = c("face", "back", "both")
)
```

### Arguments

| | |
|---|---|
| draft | A weave_draft object. |
| direction | "warp" or "weft". |
| side | "face", "back" or "both"; see float_lengths. |

### Value

A single number, or NA if the draft has no floats of that kind (for example, no warp floats on the face of an all-zero drawdown).

### Examples

```
f <- system.file("extdata", "twill_2_2.wif", package = "weaveDraft")
d <- weave_draft_from_wif(f)
mean_float_length(d, "warp")
max_float_length(d, "weft")
```

---

## interlacement_frequency

*Interlacement frequency*

---

### Description

The fraction of adjacent cell pairs in which the drawdown changes between warp-up and weft-up. A plain weave interlaces at every opportunity (1); long-float structures such as satins score low. This is a common proxy for how tightly a fabric is bound.

### Usage

```
interlacement_frequency(draft, direction = c("warp", "weft", "both"))
```

### Arguments

| | |
|---|---|
| draft | A weave_draft object. |
| direction | "warp" (changes down each column, i.e. along the warp threads), "weft" (changes along each row) or "both" (mean of the two). |

### Value

A number between 0 and 1, or NA if the draft is too small for the requested direction (fewer than 2 picks or ends).

### Examples

```
f <- system.file("extdata", "gingham_plain.wif", package = "weaveDraft")
interlacement_frequency(weave_draft_from_wif(f), "both")
```

`plot.weave_draft` *Plot a weave draft*

## Description

Draws either the binary drawdown (black = warp up) or, for drafts that carry colour information, a simulated fabric face (warp colour where the warp is up, weft colour elsewhere). The simulation is flat colour only; it ignores yarn thickness and thread spacing.

## Usage

```
## S3 method for class 'weave_draft'
plot(
  x,
  type = c("drawdown", "fabric"),
  pick_origin = c("top", "bottom"),
  end_origin = c("left", "right"),
  main = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | A `weave_draft` object. |
| `type` | `"drawdown"` (default) or `"fabric"`. |
| `pick_origin` | Where pick 1 is drawn: `"top"` or `"bottom"`. |
| `end_origin` | Where warp end 1 is drawn: `"left"` or `"right"`. |
| `main` | Plot title; defaults to the draft name. |
| `...` | Currently unused. |

## Details

By default pick 1 is at the top and warp end 1 at the left. Weaving software often draws the drawdown with pick 1 at the bottom and/or end 1 at the right; use `pick_origin` and `end_origin` to match it. Axis labels always show the original pick and end numbers.

## Value

`x`, invisibly.

## Examples

```
f <- system.file("extdata", "gingham_plain.wif", package = "weaveDraft")
d <- weave_draft_from_wif(f)
plot(d)
plot(d, type = "fabric")
```

read_wif *Read a WIF (Weaving Information File)*

## Description

Parses a '.wif' file, the open, text-based, INI-style weaving interchange format written by WeavePoint, Fiberworks, ArahWeave and most other weaving software.

## Usage

```
read_wif(path)
```

## Arguments

| | |
|---|---|
| path | Path to a '.wif' file. |

## Details

The reader extracts the sections needed to rebuild a drawdown: [WEAVING], [WARP], [WEFT], [THREADING], and either [LIFTPLAN] (dobby-style, one entry per pick listing the shafts raised) or [TIEUP] plus [TREADLING] (treadle-loom style). Files may contain both encodings; both are kept. If colour sections ([COLOR PALETTE], [COLOR TABLE], [WARP COLORS], [WEFT COLORS]) are present they are converted to hexadecimal colours. Unknown sections are ignored.

Section and key names are matched case-insensitively. Windows (CRLF) line endings and non-UTF-8 text in notes are tolerated. Full-line comments beginning with a semicolon or hash are ignored. Duplicate sections or keys and malformed integer indices are rejected. Missing thread entries are treated as inactive. Unusable colours fall back to neutral grey.

## Value

An object of class "wif", a list with elements:

**source_file** The path that was read.

**shafts, treadles** Declared numbers of shafts and treadles (NA if not declared).

**rising_shed** Logical; TRUE for a rising-shed loom.

**n_warp, n_weft** Number of warp ends and weft picks.

**threading** List with one integer vector per warp end: the shafts that end is threaded through.

**liftplan** List with one integer vector per pick (shafts raised), or NULL if the file has no liftplan.

**treadling** List with one integer vector per pick (treadles pressed), or NULL.

**tieup** List whose element i holds the shafts raised by treadle i, or NULL.

**has_liftplan, has_tieup** Logical flags for the two encodings.

**colors** NULL, or a list with palette (data frame), warp and weft (hex colour per thread).

## See Also

weave_draft_from_wif to go straight to a drawdown, check_wif_consistency to compare the two encodings.

## Examples

```
f <- system.file("extdata", "twill_2_2.wif", package = "weaveDraft")
wif <- read_wif(f)
wif
```

---

weave_draft_from_wif *Build a weave_draft from a WIF file*

---

## Description

Reads a WIF file (or takes an object from read_wif), reconstructs the drawdown and returns a weave_draft.

## Usage

```
weave_draft_from_wif(
  x,
  style = c("auto", "liftplan", "tieup_treadling"),
  name = NULL
)
```

## Arguments

| | |
|---|---|
| x | Path to a '.wif' file, or an object returned by read_wif. |
| style | Which encoding to use: "auto" (default), "liftplan" or "tieup_treadling". |
| name | Optional short name for the draft; defaults to the file name without extension. |

## Details

If the file contains both a liftplan and a tie-up/treadling pair, style = "auto" uses the liftplan, which states the sheds directly. Use check_wif_consistency to confirm the two encodings agree.

## Value

A weave_draft object.

## Examples

```
f <- system.file("extdata", "gingham_plain.wif", package = "weaveDraft")
draft <- weave_draft_from_wif(f)
draft
```

## weave_metrics *One-row summary of a draft's structural metrics*

### Description

Collects the main structural descriptors of a draft in a one-row data frame, convenient for comparing many drafts with `rbind`.

### Usage

```
weave_metrics(draft, side = c("face", "back", "both"))
```

### Arguments

| | |
|---|---|
| `draft` | A `weave_draft` object. |
| `side` | `"face"` (default), `"back"` or `"both"`, used for the float statistics. |

### Value

A one-row data frame with columns `name`, `draft_style`, `shafts`, `picks`, `ends`, `repeat_picks`, `repeat_ends`, `mean_warp_float`, `mean_weft_float`, `max_warp_float`, `max_weft_float`, `interlace_warp`, `interlace_weft`, `interlace_both` and `float_ratio`.

### Examples

```
f <- system.file("extdata", "twill_2_2.wif", package = "weaveDraft")
weave_metrics(weave_draft_from_wif(f))
```

## weave_repeat *Detect the repeat of a drawdown*

### Description

Finds the smallest number of picks and of ends after which the drawdown repeats. A partial final repeat is allowed, but the repeat must fit at least `min_repeats` times, so a pattern that only matches itself over a tiny overlap is not reported as repeating.

### Usage

```
weave_repeat(draft, min_repeats = 2)
```

### Arguments

| | |
|---|---|
| `draft` | A `weave_draft` object. |
| `min_repeats` | Minimum number of times the repeat must fit into the drawdown (default 2). If no such repeat exists the full dimension is returned. |

## Value

A named integer vector with `picks` (repeat length down the rows, along warp threads) and `ends` (repeat length across the columns, along weft threads).

## Examples

```
f <- system.file("extdata", "twill_2_2.wif", package = "weaveDraft")
weave_repeat(weave_draft_from_wif(f))
```

# Index


as_weave_draft, 2

check_wif_consistency, 3, *11*
color_runs, 3
compute_drawdown, 4, *5*
compute_drawdown_liftplan, 5

directional_float_ratio, 6

float_lengths, *6*, 6, *8*
float_summary, 7

interlacement_frequency, 8

max_float_length *(*float_summary*)*, 7
mean_float_length *(*float_summary*)*, 7

plot.weave_draft, 9

read_wif, *3*, 10, *11*

weave_draft_from_wif, *11*, 11
weave_metrics, 12
weave_repeat, 12