

# Package: Rtskit (via r-universe)

September 20, 2026

**Title** Native Access to Succinct Tree Sequences

**Version** 0.0.0.9000

**Description** Provides a native R interface to the 'tskit' C API for loading, inspecting, traversing, and writing succinct tree sequences. Tree sequences remain owned by native external pointers while typed R functions expose selected tables, opaque metadata, tree intervals, and declared source-population ancestry without a Python runtime.

**License** GPL (>= 2)

**Encoding** UTF-8

**Language** en-US

**Depends** R (>= 4.4.0)

**Imports** S7 (>= 0.2.2)

**Suggests** knitr, rmarkdown, roxygen2, tinytest

**VignetteBuilder** knitr

**SystemRequirements** C99

**URL** <https://rgenomicsetl.github.io/Rtskit/>,  
<https://github.com/RGenomicsETL/Rtskit>

**BugReports** <https://github.com/RGenomicsETL/Rtskit/issues>

**Roxygen** list(markdown = TRUE)

**Config/roxygen2/version** 8.0.0

**Collate** 'validation.R' 'tree-sequence.R' 'zzz.R'

**Repository** <https://rgenomicsetl.r-universe.dev>

**Date/Publication** 2026-07-27 20:01:57 UTC

**RemoteUrl** <https://github.com/RGenomicsETL/Rtskit>

**RemoteRef** HEAD

**RemoteSha** 89812d004f989e29dd2b78880f6617ba282f26e0

Contents

|                                    |           |
|------------------------------------|-----------|
| TreeSequence . . . . .             | 2         |
| tskit_ancestry_intervals . . . . . | 3         |
| tskit_dump . . . . .               | 4         |
| tskit_edges . . . . .              | 4         |
| tskit_example . . . . .            | 5         |
| tskit_individuals . . . . .        | 5         |
| tskit_load . . . . .               | 6         |
| tskit_metadata . . . . .           | 6         |
| tskit_metadata_schema . . . . .    | 7         |
| tskit_nodes . . . . .              | 7         |
| tskit_populations . . . . .        | 8         |
| tskit_samples . . . . .            | 8         |
| tskit_summary . . . . .            | 9         |
| tskit_trees . . . . .              | 9         |
| <b>Index</b>                       | <b>10</b> |

---

|              |                                        |
|--------------|----------------------------------------|
| TreeSequence | <i>A native succinct tree sequence</i> |
|--------------|----------------------------------------|

---

Description

TreeSequence owns one tsk\_treeseq\_t value through a native external pointer. IDs returned by table accessors retain tskit’s zero-based indexing.

Usage

```
TreeSequence(ptr = NULL, source = character(0))
```

Arguments

- ptr Internal live native pointer.
- source Source path or "<memory>".

---

tskit\_ancestry\_intervals

*Extract source-population ancestry intervals*


---

## Description

For every focal sample and marginal tree, this function walks toward the root and records the first node whose population is one of `source_populations`. Adjacent tree intervals with the same source are merged. This is exact ancestry truth only when the simulation's population and node-retention semantics make those source populations authoritative.

## Usage

```
tskit_ancestry_intervals(
  x,
  source_populations,
  samples = NULL,
  require_resolved = TRUE
)
```

## Arguments

|                                 |                                                                                                                                                  |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>                  | A <code>TreeSequence</code> .                                                                                                                    |
| <code>source_populations</code> | A non-empty vector of zero-based source-population IDs. Names, when supplied, become the returned source labels and should be dataset-qualified. |
| <code>samples</code>            | Zero-based focal sample-node IDs. <code>NULL</code> uses every sample.                                                                           |
| <code>require_resolved</code>   | Whether to fail if a focal lineage reaches a root without encountering a declared source population.                                             |

## Value

A data frame of sample IDs, half-open intervals, source labels, and source-population IDs, grouped by sample and ordered by coordinate.

## References

Haller et al. (2019), *Tree-sequence recording in SLiM opens new horizons for forward-time simulation of whole genomes*, doi:[10.1111/17550998.12968](https://doi.org/10.1111/17550998.12968). The published local-ancestry example is pinned at SLiMTreeSeqPub commit 6715c28b02942bc4757c9f8bcab133ad4a0bfcfb.

---

|            |                                       |
|------------|---------------------------------------|
| tskit_dump | <i>Write a succinct tree sequence</i> |
|------------|---------------------------------------|

---

### Description

Write a succinct tree sequence

### Usage

```
tskit_dump(x, path, overwrite = FALSE)
```

### Arguments

|           |                                                           |
|-----------|-----------------------------------------------------------|
| x         | A TreeSequence.                                           |
| path      | Destination .trees path. Its parent directory must exist. |
| overwrite | Whether to replace an existing file.                      |

### Value

The normalized destination path, invisibly.

---

|             |                                   |
|-------------|-----------------------------------|
| tskit_edges | <i>Materialize the edge table</i> |
|-------------|-----------------------------------|

---

### Description

Metadata remains native and is not decoded by this initial accessor.

### Usage

```
tskit_edges(x)
```

### Arguments

|   |                 |
|---|-----------------|
| x | A TreeSequence. |
|---|-----------------|

### Value

A data frame with zero-based IDs and edge coordinates and nodes.

---

|               |                                                 |
|---------------|-------------------------------------------------|
| tskit_example | Create deterministic example tree-sequence data |
|---------------|-------------------------------------------------|

---

**Description**

The example contains two sample nodes from one individual and two source ancestors spanning adjacent trees over a sequence of length 10. Population and individual metadata, one site, and one mutation exercise the native constructor without external software.

**Usage**

```
tskit_example()
```

**Value**

An in-memory TreeSequence.

---

|                   |                                  |
|-------------------|----------------------------------|
| tskit_individuals | Materialize the individual table |
|-------------------|----------------------------------|

---

**Description**

Ragged locations and parent IDs are returned as list columns. Individual metadata remains opaque and is available through [tskit\\_metadata\(\)](#).

**Usage**

```
tskit_individuals(x)
```

**Arguments**

|   |                 |
|---|-----------------|
| x | A TreeSequence. |
|---|-----------------|

**Value**

A data frame with zero-based individual IDs, flags, location and parent list columns, and metadata lengths.

---

|            |                                      |
|------------|--------------------------------------|
| tskit_load | <i>Load a succinct tree sequence</i> |
|------------|--------------------------------------|

---

**Description**

Load a succinct tree sequence

**Usage**

tskit\_load(path)

**Arguments**

path                      Existing .trees file.

**Value**

A TreeSequence owning the loaded native value.

---

|                |                                   |
|----------------|-----------------------------------|
| tskit_metadata | <i>Read opaque metadata bytes</i> |
|----------------|-----------------------------------|

---

**Description**

Rtskit deliberately returns raw bytes and does not infer a codec from a schema. Row tables return a list of raw vectors in the requested ID order. Top-level tree-sequence and reference-sequence metadata return one raw vector and do not accept id.

**Usage**

tskit\_metadata(x, table, id = NULL)

**Arguments**

x                          A TreeSequence.  
table                      One of "tree\_sequence", "reference\_sequence", "individuals", "nodes", "edges", "migrations", "sites", "mutations", or "populations".  
id                          Optional zero-based row IDs. NULL requests every row of a row table.

**Value**

A raw vector for top-level metadata or a list of raw vectors for a row table.

---

|                       |                                       |
|-----------------------|---------------------------------------|
| tskit_metadata_schema | <i>Read an opaque metadata schema</i> |
|-----------------------|---------------------------------------|

---

**Description**

The schema is returned exactly as stored. Convert or decode it explicitly only when the declared codec is understood by the caller.

**Usage**

```
tskit_metadata_schema(x, table)
```

**Arguments**

|       |                                                                                                                                      |
|-------|--------------------------------------------------------------------------------------------------------------------------------------|
| x     | A TreeSequence.                                                                                                                      |
| table | One of "tree_sequence", "reference_sequence", "individuals", "nodes", "edges", "migrations", "sites", "mutations", or "populations". |

**Value**

A raw vector containing the table metadata schema.

---

|             |                                   |
|-------------|-----------------------------------|
| tskit_nodes | <i>Materialize the node table</i> |
|-------------|-----------------------------------|

---

**Description**

Metadata remains native and is not decoded by this initial accessor.

**Usage**

```
tskit_nodes(x)
```

**Arguments**

|   |                 |
|---|-----------------|
| x | A TreeSequence. |
|---|-----------------|

**Value**

A data frame with zero-based IDs, flags, time, population, and individual columns.

---

|                   |                                         |
|-------------------|-----------------------------------------|
| tskit_populations | <i>Materialize the population table</i> |
|-------------------|-----------------------------------------|

---

**Description**

Population metadata remains opaque. Use `tskit_metadata()` to request its bytes explicitly.

**Usage**

```
tskit_populations(x)
```

**Arguments**

`x` A TreeSequence.

**Value**

A data frame with zero-based population IDs and metadata lengths.

---

|               |                               |
|---------------|-------------------------------|
| tskit_samples | <i>Return sample node IDs</i> |
|---------------|-------------------------------|

---

**Description**

Return sample node IDs

**Usage**

```
tskit_samples(x)
```

**Arguments**

`x` A TreeSequence.

**Value**

An integer vector of zero-based node IDs.

---

|               |                                           |
|---------------|-------------------------------------------|
| tskit_summary | <i>Summarize a succinct tree sequence</i> |
|---------------|-------------------------------------------|

---

**Description**

Summarize a succinct tree sequence

**Usage**

```
tskit_summary(x)
```

**Arguments**

x                    A TreeSequence.

**Value**

A named list of sequence length and table/tree counts. Counts are doubles because tskit sizes can exceed R's integer range.

---

|             |                                |
|-------------|--------------------------------|
| tskit_trees | <i>Traverse tree intervals</i> |
|-------------|--------------------------------|

---

**Description**

Traverse tree intervals

**Usage**

```
tskit_trees(x)
```

**Arguments**

x                    A TreeSequence.

**Value**

A data frame containing each tree index, half-open interval, root count, and active edge count.

# Index

TreeSequence, [2](#)  
tskit\_ancestry\_intervals, [3](#)  
tskit\_dump, [4](#)  
tskit\_edges, [4](#)  
tskit\_example, [5](#)  
tskit\_individuals, [5](#)  
tskit\_load, [6](#)  
tskit\_metadata, [6](#)  
tskit\_metadata(), [5](#), [8](#)  
tskit\_metadata\_schema, [7](#)  
tskit\_nodes, [7](#)  
tskit\_populations, [8](#)  
tskit\_samples, [8](#)  
tskit\_summary, [9](#)  
tskit\_trees, [9](#)