

# Package: gromovlab (via r-universe)

September 1, 2024

**Version** 0.8-3

**Date** 2021-02-04

**Title** Gromov-Hausdorff Type Distances for Labeled Metric Spaces

**Maintainer** Volkmar Liebscher <volkmar.liebscher@uni-greifswald.de>

**Author** Volkmar Liebscher

**Description** Computes Gromov-Hausdorff type  $l^p$  distances for labeled metric spaces. These distances were introduced in V.Liebscher, Gromov meets Phylogenetics - new Animals for the Zoo of Metrics on Tree Space <[arXiv:1504.05795](https://arxiv.org/abs/1504.05795)> for phylogenetic trees, but may apply to a diversity of scenarios.

**Imports** methods,igraph, ape, glpkAPI , quadprog, cluster, stats

**Suggests** phangorn

**Depends** R (>= 2.15.1)

**License** GPL (>= 2)

**NeedsCompilation** no

**Date/Publication** 2021-02-04 07:50:02 UTC

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

**RemoteUrl** <https://github.com/cran/gromovlab>

**RemoteRef** HEAD

**RemoteSha** 8acc556f88feb0bfff1db9a1797f530bf2b446db

## Contents

gromovdist . . . . .	2
<b>Index</b>	<b>4</b>

gromovdist

*Gromov-Hausdorff-type distances of labelled metric spaces***Description**

The function `gromovdist` calculates the matched Gromov- $\ell^p$  distances of two metrics on a finite space  $X$ :

$$D_p(\rho_1, \rho_2) = \inf\{\|(|d(\phi_1(x), \phi_2(x))|)_{x \in X}\|_p\}.$$

There, the infimum is taken over all isometric embeddings  $\phi_1$  of  $(X, \rho_1)$ ,  $\phi_2$  of  $(X, \rho_2)$  into a common metric space  $(Y, d)$ . Only  $1 \leq p \leq \infty$  is considered.

At the basis is the reformulation of the metric as value of a convex program, see Liebscher (2015).

Methods for various classes are provided:

- `dist`, `dissimilarity` which are distance matrices.
- `matrix` for matrices containing the individual distances of the elements of  $X$ .
- `igraph` for connected graphs. The metric on the nodes or just the leaves (nodes of degree 1) of the graph is the length of the shortest path.
- `phylo` for phylogenetic trees. Again the metric is induced by the graph.

**Usage**

```
gromovdist(d1, d2=NULL, type="l1", p=NULL, ...)

## S3 method for class 'list'
gromovdist(d1, d2=NULL, type="l1", p=NULL, ...)
## S3 method for class 'multiPhylo'
gromovdist(d1, d2=NULL, type="l1", p=NULL, ...)
## S3 method for class 'phylo'
gromovdist(d1, d2=NULL, type="l1", p=NULL, ...)
## S3 method for class 'dist'
gromovdist(d1, d2=NULL, type="l1", p=NULL, ...)
## S3 method for class 'dissimilarity'
gromovdist(d1, d2=NULL, type="l1", p=NULL, ...)
## S3 method for class 'matrix'
gromovdist(d1, d2=NULL, type="l1", p=NULL, ...)
## S3 method for class 'igraph'
gromovdist(d1, d2=NULL, type="l1", p=NULL, leavesonly=TRUE, ...)
```

**Arguments**

<code>type</code>	type of metric to use
<code>d1, d2</code>	distance object(s).
<code>p</code>	if <code>type="lp"</code> the value of $p$ . If not supplied, $p = 2$ is used.
<code>leavesonly</code>	compute the distances between the leaves of the graph/tree only?
<code>...</code>	further parameters

**Details**

type="l1" yields  $p = 1$

type="l2" yields  $p = 2$

type="linfinity" yields  $p = \infty$

type="lp" is for (not so efficient) computation using [constrOptim](#) for arbitrary  $1 \leq p < \infty$

If d1 is a list, the distance matrix between all elements of the list is computed. It is represented as an object of class `dissimilarity`, see [dissimilarity.object](#).

The distance is only computed for that part of the objects where the labels are present in both objects. If there are no labels the elements are numbered consecutively.

**Value**

The distance (one numeric) or a distance matrix for the `list` and `multiPhylo` methods

**Author(s)**

Volkmar Liebscher

**References**

V.Liebscher, Gromov meets Phylogenetics - new Animals for the Zoo of Metrics on Tree Space. preprint 2015 arXiv:1504.05795

**See Also**

[dist](#), [dissimilarity.object](#), [phangorn-package](#), [igraph-package](#), [constrOptim](#).

**Examples**

```
library("ape")
tr1<-rtree(n=10)
tr2<-rtree(n=10)
gromovdist(tr1,tr2,"l1")
gromovdist(tr1,tr2,"l2")
#thesame, but slower
gromovdist(d1=tr1,d2=tr2,type="lp",p=2)
gromovdist(tr1,tr2,"linf")
```

# Index

- \* **Gromov-Hausdorff metric**
  - [gromovdist](#), [2](#)
- \* **dissimilarity**
  - [gromovdist](#), [2](#)
- \* **finite labeled metric space**
  - [gromovdist](#), [2](#)
- \* **tree space**
  - [gromovdist](#), [2](#)

[constrOptim](#), [3](#)

[dissimilarity.object](#), [3](#)

[dist](#), [3](#)

[gromovdist](#), [2](#)

[gromovdist](#), [dissimilarity-method](#)  
    ([gromovdist](#)), [2](#)

[gromovdist](#), [dist-method](#) ([gromovdist](#)), [2](#)

[gromovdist](#), [igraph-method](#) ([gromovdist](#)), [2](#)

[gromovdist](#), [list-method](#) ([gromovdist](#)), [2](#)

[gromovdist](#), [matrix-method](#) ([gromovdist](#)), [2](#)

[gromovdist](#), [multiPhylo-method](#)  
    ([gromovdist](#)), [2](#)

[gromovdist](#), [phylo-method](#) ([gromovdist](#)), [2](#)

[gromovdist.default](#) ([gromovdist](#)), [2](#)

[gromovdist.dissimilarity](#) ([gromovdist](#)), [2](#)

[gromovdist.dist](#) ([gromovdist](#)), [2](#)

[gromovdist.igraph](#) ([gromovdist](#)), [2](#)

[gromovdist.list](#) ([gromovdist](#)), [2](#)

[gromovdist.matrix](#) ([gromovdist](#)), [2](#)

[gromovdist.multiPhylo](#) ([gromovdist](#)), [2](#)

[gromovdist.phylo](#) ([gromovdist](#)), [2](#)

[igraph-package](#), [3](#)

[phangorn-package](#), [3](#)