

# Package ‘FastHamming’

April 25, 2025

**Title** Fast Computation of Pairwise Hamming Distances

**Type** Package

**Version** 1.1

**Depends** R (>= 4.0.0)

**Description** Pairwise Hamming distances are computed between the rows of a binary (0/1) matrix using highly optimized C code. The input is an integer matrix where each row represents a binary feature vector and returns a symmetric integer matrix of pairwise distances. Internally, rows are bit-packed into 64-bit words for fast XOR-based comparisons, with hardware-accelerated popcount operations to count differences. OpenMP parallelization ensures efficient performance for large matrices.

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**SystemRequirements** C compiler (C99), OpenMP

**NeedsCompilation** yes

**Author** Ravi Varadhan [aut, cre]

**Maintainer** Ravi Varadhan <ravi.varadhan@jhu.edu>

**Repository** CRAN

**Date/Publication** 2025-04-25 12:30:18 UTC

## Contents

hamming_distance . . . . .	2
<b>Index</b>	3

---

hamming_distance	<i>Pairwise Hamming distances</i>
------------------	-----------------------------------

---

**Description**

Computes the pairwise Hamming distances between rows of a binary matrix.

**Usage**

```
hamming_distance(X, nthreads = NULL)
```

**Arguments**

X	A binary (0/1) numeric matrix.
nthreads	Integer; number of OpenMP threads to use. If NULL (the default) use all available cores,

**Value**

An integer matrix of pairwise Hamming distances.

**Examples**

```
n <- 10000
m <- 1000
set.seed(2468)
X <- matrix(sample(0:1, n * m, replace = TRUE), nrow = n)
# Use all available threads
system.time(result <- hamming_distance(X))
# limit to 2 threads
system.time(hamming_distance(X, nthreads = 2))
```

# Index

hamming\_distance, [2](#)