

# Package ‘hdImpute’

October 13, 2022

**Type** Package

**Title** A Batch Process for High Dimensional Imputation

**Version** 0.1.1

**BugReports** <https://github.com/pdwaggoner/hdImpute/issues>

**Maintainer** Philip Waggoner <philip.waggoner@gmail.com>

**Description** A correlation-based batch process for fast imputation for high dimensional missing data problems via chained random forests. See Stekhoven and Bühlmann (2012) <[doi:10.1093/bioinformatics/btr597](https://doi.org/10.1093/bioinformatics/btr597)> for more on missForest, and Mayer (2022) <<https://github.com/mayer79/missRanger>> for more on missRanger.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** missRanger, plyr, purrr, magrittr, tibble, dplyr, tidyselect, cli

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, usethis, missForest, tidyverse

**VignetteBuilder** knitr

**RoxygenNote** 7.1.2

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Philip Waggoner [aut, cre]

**Repository** CRAN

**Date/Publication** 2022-04-20 21:12:28 UTC

## R topics documented:

feature_cor	2
flatten_mat	2
hdImpute	3
impute_batches	4

<b>Index</b>	<b>6</b>
--------------	----------

---

feature_cor	<i>High dimensional imputation via batch processed chained random forests Build correlation matrix</i>
-------------	--

---

**Description**

High dimensional imputation via batch processed chained random forests Build correlation matrix

**Usage**

```
feature_cor(data, return_cor)
```

**Arguments**

data	A data object.
return_cor	Logical. Should the correlation matrix be printed? Default set to FALSE.

**Value**

A cross-feature correlation matrix

**References**

van Buuren S, Groothuis-Oudshoorn K (2011). "mice: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software*, 45(3), 1-67. doi: <10.18637/jss.v045.i03>

**Examples**

```
## Not run:
feature_cor(data = data, return_cor = FALSE)

## End(Not run)
```

---

flatten_mat	<i>Flatten and arrange cor matrix to be df</i>
-------------	--

---

**Description**

Flatten and arrange cor matrix to be df

**Usage**

```
flatten_mat(cor_mat, return_mat)
```

**Arguments**

cor\_mat            A correlation matrix output from running feature\_cor()  
 return\_mat        Logical. Should the flattened matrix be printed? Default set to FALSE.

**Value**

A vector of correlation-based ranked features

**Examples**

```
## Not run:
flatten_mat(cor_mat = cor_mat, return_mat = FALSE)

## End(Not run)
```

---

hdImpute	<i>Complete hdImpute process: correlation matrix, flatten, rank, create batches, impute, join</i>
----------	---

---

**Description**

Complete hdImpute process: correlation matrix, flatten, rank, create batches, impute, join

**Usage**

```
hdImpute(data, batch, pmm_k, n_trees, seed, save)
```

**Arguments**

data                Original data frame (with missing values)  
 batch              Numeric. Batch size.  
 pmm\_k              Integer. Number of neighbors considered in imputation. Default set at 5.  
 n\_trees            Integer. Number of trees used in imputation. Default set at 15.  
 seed                Integer. Seed to be set for reproducibility.  
 save                Should the list of individual imputed batches be saved as .rds file to working directory? Default set to FALSE.

**Details**

Step 1. group data by dividing the row\_number() by batch size (batch, number of batches set by user) using integer division. Step 2. pass through group\_split() to return a list. Step 3. impute each batch individually and time. Step 4. generate completed (unlisted/joined) imputed data frame

**Value**

A completed, imputed data set

## References

Stekhoven, D. J., & Bühlmann, P. (2012). MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1), 112-118.

## Examples

```
## Not run:
impute_batches(data = data,
batch = 2, pmm_k = 5, n_trees = 15,
seed = 123, save = FALSE)

## End(Not run)
```

---

impute_batches	<i>Impute batches and return completed data frame</i>
----------------	---

---

## Description

Impute batches and return completed data frame

## Usage

```
impute_batches(data, features, batch, pmm_k, n_trees, seed, save)
```

## Arguments

data	Original data frame (with missing values)
features	Correlation-based vector of ranked features output from running <code>flatten_mat()</code>
batch	Numeric. Batch size.
pmm_k	Integer. Number of neighbors considered in imputation. Default at 5.
n_trees	Integer. Number of trees used in imputation. Default at 15.
seed	Integer. Seed to be set for reproducibility.
save	Should the list of individual imputed batches be saved as .rds file to working directory? Default set to FALSE.

## Details

Step 1. group data by dividing the `row_number()` by batch size (batch, number of batches set by user) using integer division. Step 2. pass through `group_split()` to return a list. Step 3. impute each batch individually and time. Step 4. generate completed (unlisted/joined) imputed data frame

## Value

A completed, imputed data set

**References**

Stekhoven, D. J., & Bühlmann, P. (2012). MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1), 112-118.

**Examples**

```
## Not run:  
impute_batches(data = data, features = flat_mat,  
batch = 2, pmm_k = 5, n_trees = 15, seed = 123,  
save = FALSE)
```

```
## End(Not run)
```

# Index

`feature_cor`, [2](#)

`flatten_mat`, [2](#)

`hdImpute`, [3](#)

`impute_batches`, [4](#)