

# Package ‘IMPCdata’

November 14, 2025

**Type** Package

**Title** Retrieves data from IMPC database

**Version** 1.47.0

**Date** 2016-06-09

**Author** Natalja Kurbatova, Jeremy Mason

**Maintainer** Jeremy Mason <jmason@ebi.ac.uk>

**Depends** R (>= 2.3.0)

**Imports** rjson

**Description** Package contains methods for data retrieval from IMPC Database.

**License** file LICENSE

**biocViews** ExperimentData

**git\_url** <https://git.bioconductor.org/packages/IMPCdata>

**git\_branch** devel

**git\_last\_commit** f877a20

**git\_last\_commit\_date** 2025-10-29

**Repository** Bioconductor 3.23

**Date/Publication** 2025-11-13

## Contents

getAlleles . . . . .	2
getGenes . . . . .	3
getIMPCDataset . . . . .	4
getIMPCTable . . . . .	5
getName . . . . .	6
getParameters . . . . .	6
getPhenCenters . . . . .	7
getPipelines . . . . .	8
getProcedures . . . . .	8
getStrains . . . . .	9

getZygositys	10
printAlleles	11
printGenes	12
printParameters	12
printPhenCenters	13
printPipelines	14
printProcedures	14
printStrains	15

<b>Index</b>	<b>17</b>
--------------	-----------

---

getAlleles	<i>Method "getAlleles"</i>
------------	----------------------------

---

## Description

Returns alleles (MGI IDs or temporary IDs if the MGI ID is not assigned yet) that are processed for a specified combination of parameter, procedure, pipeline and phenotyping center. Temporary allele ID starts with "NULL".

## Usage

```
getAlleles(PhenCenterName=NULL, PipelineID=NULL, ProcedureID=NULL,
ParameterID=NULL, StrainID=NULL)
```

## Arguments

PhenCenterName	IMPC phenotyping center; mandatory argument
PipelineID	IMPC pipeline ID; mandatory argument
ProcedureID	IMPC procedure ID; mandatory argument
ParameterID	IMPC parameter ID; mandatory argument
StrainID	IMPC strain ID; optional argument

## Value

Returns the list of IMPC alleles (IDS of alleles) that are processed measuring specified parameter within the procedure in the pipeline run by phenotyping center.

## Author(s)

Natalja Kurbatova, Jeremy Mason

**Examples**

```
listAlleles <- getAlleles("RBRC","IMPC_001","IMPC_GRS_001","IMPC_GRS_003_001")
for (alleleIndex in 1:min(length(listAlleles), 5)) {
  print(paste(listAlleles[alleleIndex],"-",
             getName("allele_accession_id","allele_symbol",
                    listAlleles[alleleIndex])))
}
```

getGenes

*Method "getGenes"***Description**

Returns the genes (MGI IDs) that are reported for a specified combination of parameter, procedure, pipeline and phenotyping center.

**Usage**

```
getGenes(PhenCenterName=NULL, PipelineID=NULL, ProcedureID=NULL, ParameterID=NULL, StrainID=NULL)
```

**Arguments**

PhenCenterName	IMPC phenotyping center; mandatory argument
PipelineID	IMPC pipeline ID; mandatory argument
ProcedureID	IMPC procedure ID; mandatory argument
ParameterID	IMPC parameter ID; mandatory argument
StrainID	IMPC strain ID; optional argument

**Value**

Returns the list of IMPC genes (IDS of genes) that are processed measuring specified parameter within the procedure in the pipeline run by phenotyping center.

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
listGenes <- getGenes("RBRC","IMPC_001","IMPC_GRS_001","IMPC_GRS_003_001")
for (geneIndex in 1:min(length(listGenes), 5)) {
  print(paste(listGenes[geneIndex],"-",
             getName("gene_accession_id","gene_symbol",listGenes[geneIndex])))
}
```

---

getIMPCDataset	<i>Method "getIMPCDataset"</i>
----------------	--------------------------------

---

### Description

Returns the IMPC dataset for a knockout line for a specified combination of parameter, procedure, pipeline, phenotyping centre, allele and strain (genetic background) that is used within the data analysis pipeline implemented on the IMPC database.

### Usage

```
getIMPCDataset(PhenCenterName=NULL, PipelineID=NULL, ProcedureID=NULL, ParameterID=NULL,
               AlleleID=NULL, StrainID=NULL)
```

### Arguments

PhenCenterName	IMPC phenotyping center; mandatory argument
PipelineID	IMPC pipeline ID; mandatory argument
ProcedureID	IMPC procedure ID; mandatory argument
ParameterID	IMPC parameter ID; mandatory argument
AlleleID	IMPC allele ID; mandatory argument
StrainID	IMPC strain ID; optional argument

### Value

Returns the IMPC dataset as a data frame. - metadata\_group string a string indicating a group of experimental and control mice that have the same metadata (see also biological\_sample\_group)  
- metadata list of strings list showing all relevant metadata in effect when the data was collected

### Author(s)

Natalja Kurbatova, Jeremy Mason

### Examples

```
df <- getIMPCDataset("WTSI", "MGP_001", "IMPC_CBC_001", "IMPC_CBC_008_001", "MGI:4433526")
```

---

getIMPCTable	<i>Method "getIMPCTable"</i>
--------------	------------------------------

---

### Description

Returns the IMPC dataset. There are optional arguments to subset the IMPC dataset. Be aware that without filters this file output will be very large and the query will be time intensive.

### Usage

```
getIMPCTable(fileName="IMPCdata",PhenCenterName=NULL, PipelineID=NULL, ProcedureID=NULL, ParameterID=NULL, AlleleID=NULL, StrainID=NULL, multipleFiles=TRUE,recordsPerFile=10000)
```

### Arguments

fileName	name of the file where to save resulting table with IMPC objects; mandatory argument; default value is set to 'IMPCdata'
PhenCenterName	IMPC phenotyping center
PipelineID	IMPC pipeline ID
ProcedureID	IMPC procedure ID
ParameterID	IMPC parameter ID
AlleleID	allele ID
StrainID	strain ID
multipleFiles	flag: "FALSE" value to get all records into one specified file; "TRUE" value (default) to split records across multiple files named starting with 'fileName'
recordsPerFile	number that specifies how many records to write into one file; default value is 10000

### Value

Returns the IMPC dataset.

### Author(s)

Natalja Kurbatova, Jeremy Mason

### Examples

```
getIMPCTable("./IMPCData_WTSI", "WTSI", "MGP_001", "IMPC_CBC_001", "IMPC_CBC_003_001", "MGI:4431644")
```

---

getName	<i>Method "getName"</i>
---------	-------------------------

---

**Description**

Returns IMPC object name by its ID. An internal function, if called directly please use with caution.

**Usage**

```
getName(fieldNameFrom, fieldNameTo, fieldValueFrom)
```

**Arguments**

fieldNameFrom name of the IMPC field for the object ID; mandatory argument  
 fieldNameTo name of the IMPC field for the object name; mandatory argument  
 fieldValueFrom actual IMPC object's ID to get the name for; mandatory argument

**Value**

Returns name of the object.

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
getName("pipeline_stable_id", "pipeline_name", "MGP_001")
```

---

getParameters	<i>Method "getParameters"</i>
---------------	-------------------------------

---

**Description**

Returns the list of parameters (IMPreSS IDs) that are measured within specified procedure for a pipeline run by phenotyping center.

**Usage**

```
getParameters(PhenCenterName=NULL, PipelineID=NULL, ProcedureID=NULL)
```

**Arguments**

PhenCenterName IMPC phenotyping center; mandatory argument  
 PipelineID IMPC pipeline ID; mandatory argument  
 ProcedureID IMPC procedure ID; mandatory argument

**Value**

Returns the list of IMPC parameters (IDS of parameters) that are measured within specified procedure in the pipeline run by phenotyping center.

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
listParameters <- getParameters("WTSI", "MGP_001", "IMPC_CBC_001")
for (parameterIndex in 1:min(length(listParameters), 5)) {
  print(paste(listParameters[parameterIndex], "-",
             getName("parameter_stable_id", "parameter_name", listParameters[parameterIndex])))
}
```

---

getPhenCenters

*Method "getPhenCenters"*

---

**Description**

Returns the list of IMPC phenotyping centers.

**Usage**

```
getPhenCenters(excludeLegacyPipelines=TRUE)
```

**Arguments**

excludeLegacyPipelines

flag: FALSE value to get all IMPC phenotyping centers that run all kind of pipelines including legacy pipelines; TRUE value (default) to get IMPC phenotyping centers excluding centers that run only legacy pipelines

**Value**

Returns the list of IMPC phenotyping centers.

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
getPhenCenters()
```

---

getPipelines	<i>Method "getPipelines"</i>
--------------	------------------------------

---

**Description**

Returns the list of IMPC pipelines (IMPreSS IDs) that are run by specified phenotyping center.

**Usage**

```
getPipelines(PhenCenterName=NULL,excludeLegacyPipelines=TRUE)
```

**Arguments**

PhenCenterName IMPC phenotyping center; mandatory argument

excludeLegacyPipelines

flag: "FALSE" value to get all pipelines including legacy pipelines; "TRUE" value (default) to get current IMPC pipelines excluding legacy pipelines

**Value**

Returns the list of IMPC pipelines (IDS of pipelines) that are run by specified phenotyping center.

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
listPipelines <- getPipelines("WTSI")
for (pipelineIndex in 1:length(listPipelines)) {
  print(paste(listPipelines[pipelineIndex],"-",
             getName("pipeline_stable_id","pipeline_name",listPipelines[pipelineIndex])))
}
```

---

getProcedures	<i>Method "getProcedures"</i>
---------------	-------------------------------

---

**Description**

Returns the list of procedures (IMPreSS IDs) that are run for a specified phenotyping center and pipeline.

**Usage**

```
getProcedures(PhenCenterName=NULL,PipelineID=NULL)
```

**Arguments**

PhenCenterName IMPC phenotyping center; mandatory argument  
PipelineID IMPC pipeline ID; mandatory argument

**Value**

Returns the list of IMPC procedures (IDS of procedures) that are run by specified phenotyping center and pipeline.

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
listProcedures <- getProcedures("WTSI","MGP_001")
for (procedureIndex in 1:min(length(listProcedures), 5)) { # Print the name of 5 procedures
  print(paste(listProcedures[procedureIndex],"-",
             getName("procedure_stable_id","procedure_name",listProcedures[procedureIndex])))
}
```

---

getStrains

*Method "getStrains"*

---

**Description**

Returns the genetic background (MGI IDs) from which the knockout mice were derived for a specific combination of pipeline, procedure and parameter for a phenotyping center.

**Usage**

```
getStrains(PhenCenterName=NULL,PipelineID=NULL,ProcedureID=NULL,ParameterID=NULL)
```

**Arguments**

PhenCenterName IMPC phenotyping center; mandatory argument  
PipelineID IMPC pipeline ID; mandatory argument  
ProcedureID IMPC procedure ID; mandatory argument  
ParameterID IMPC parameter ID; mandatory argument

**Value**

Returns the list of IMPC strains (IDS of strains) that are processed measuring specified parameter within the procedure in the pipeline run by phenotyping center.

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
listStrains <- getStrains("WTSI","MGP_001","IMPC_CBC_001","IMPC_CBC_003_001")
for (strainIndex in 1:length(listStrains)) {
  print(paste(listStrains[strainIndex],"-",
             getName("strain_accession_id","strain_name",listStrains[strainIndex])))
}
```

---

getZygosities                      *Method "getZygosities"*

---

**Description**

Returns the list of zygosities (homozygous, heterozygous and hemizygous) for mice that were measured for a gene/allele for a specified combination of parameter, procedure, pipeline and phenotyping center.

**Usage**

```
getZygosities(PhenCenterName=NULL,PipelineID=NULL,ProcedureID=NULL,ParameterID=NULL,
             StrainID=NULL,GeneID=NULL, AlleleID=NULL)
```

**Arguments**

PhenCenterName	IMPC phenotyping center; mandatory argument
PipelineID	IMPC pipeline ID; mandatory argument
ProcedureID	IMPC procedure ID; mandatory argument
ParameterID	IMPC parameter ID; mandatory argument
StrainID	IMPC strain ID; optional argument
GeneID	IMPC gene ID; optional argument
AlleleID	IMPC allele ID; optional argument

**Value**

Returns the list of IMPC genes (IDS of genes) that are processed measuring specified parameter within the procedure in the pipeline run by phenotyping center.

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
listZygosities <- getZygosities("WTSI","MGP_001","IMPC_CBC_001","IMPC_CBC_003_001",GeneID="MGI:1194894")
for (zygosityIndex in 1:length(listZygosities)) {
  print(listZygosities[zygosityIndex])
}
```

---

printAlleles	<i>Method "printAlleles"</i>
--------------	------------------------------

---

### Description

Prints out alleles (MGI IDs or temporary IDs if the MGI is not assigned yet and symbols) that are processed for a specified combination of parameter, procedure, pipeline and phenotyping center. Temporary allele ID starts with "NULL".

### Usage

```
printAlleles(PhenCenterName=NULL, PipelineID=NULL, ProcedureID=NULL,  
ParameterID=NULL, StrainID=NULL, n=NULL)
```

### Arguments

PhenCenterName	IMPC phenotyping center; mandatory argument
PipelineID	IMPC pipeline ID; mandatory argument
ProcedureID	IMPC procedure ID; mandatory argument
ParameterID	IMPC parameter ID; mandatory argument
StrainID	IMPC strain ID; optional argument
n	print only first n alleles; optional argument

### Value

Returns the character containing IMPC alleles (IDs and names of alleles)

### Author(s)

Natalja Kurbatova, Jeremy Mason

### Examples

```
printAlleles("RBRC", "IMPC_001", "IMPC_GRS_001", "IMPC_GRS_003_001", n = 5)
```

---

printGenes                      *Method "printGenes"*

---

**Description**

Prints out genes (MGI IDs and symbols) that are reported for a specified combination of parameter, procedure, pipeline and phenotyping center.

**Usage**

```
printGenes(PhenCenterName=NULL, PipelineID=NULL, ProcedureID=NULL,
ParameterID=NULL, StrainID=NULL, n=NULL)
```

**Arguments**

PhenCenterName	IMPC phenotyping center; mandatory argument
PipelineID	IMPC pipeline ID; mandatory argument
ProcedureID	IMPC procedure ID; mandatory argument
ParameterID	IMPC parameter ID; mandatory argument
StrainID	IMPC strain ID; optional argument
n	print only first n genes; optional argument

**Value**

Returns the character containing IMPC genes (IDs and names of genes)

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
printGenes("RBRC", "IMPC_001", "IMPC_GRS_001", "IMPC_GRS_003_001", n = 5)
```

---

printParameters                      *Method "printParameters"*

---

**Description**

Prints out the list of parameters (IMPreSS IDs and names) that are measured within specified procedure for a pipeline run by phenotyping center.

**Usage**

```
printParameters(PhenCenterName=NULL, PipelineID=NULL, ProcedureID=NULL, n=NULL)
```

**Arguments**

PhenCenterName IMPC phenotyping center; mandatory argument  
 PipelineID IMPC pipeline ID; mandatory argument  
 ProcedureID IMPC procedure ID; mandatory argument  
 n print only first n parameters; optional argument

**Value**

Returns the character containing IMPC parameters (IDs and names of parameters)

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
printParameters("WTSI", "MGP_001", "IMPC_CBC_001", n = 5)
```

---

printPhenCenters      *Method "printPhenCenters"*

---

**Description**

Prints out the list of IMPC phenotyping centers.

**Usage**

```
printPhenCenters(n=NULL, excludeLegacyPipelines=TRUE)
```

**Arguments**

n print only first n phenotyping centers; optional argument  
 excludeLegacyPipelines  
 flag: "FALSE" value to print all IMPC phenotyping centers that run all kind of pipelines including legacy pipelines; "TRUE" value (default) to print IMPC phenotyping centers excluding centers that run only legacy pipelines

**Value**

Returns the character with IMPC phenotyping centers.

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
printPhenCenters(excludeLegacyPipelines = FALSE)
```

---

printPipelines      *Method "printPipelines"*

---

**Description**

Prints out the list of IMPC pipelines (IMPreSS IDs and names) that are run by specified phenotyping center.

**Usage**

```
printPipelines(PhenCenterName=NULL,n=NULL, excludeLegacyPipelines=TRUE)
```

**Arguments**

PhenCenterName    IMPC phenotyping center; mandatory argument  
n                    print only first n pipelines; optional argument  
excludeLegacyPipelines    flag: "FALSE" value to print all pipelines including legacy pipelines; "TRUE" value (default) to print current IMPC pipelines excluding legacy pipelines

**Value**

Returns the character containing IMPC pipelines (IDs and names of pipelines)

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
printPipelines("WTSI")
```

---

printProcedures      *Method "printProcedures"*

---

**Description**

Prints out the list of procedures (IMPreSS IDs and names) that are run for a specified phenotyping center and pipeline.

**Usage**

```
printProcedures(PhenCenterName=NULL, PipelineID=NULL, n=NULL)
```

**Arguments**

PhenCenterName IMPC phenotyping center; mandatory argument  
PipelineID IMPC pipeline ID; mandatory argument  
n print only first n procedures; optional argument

**Value**

Returns the character containing IMPC procedures (IDs and names of procedures)

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
printProcedures("WTSI", "MGP_001", 5)
```

---

printStrains                    *Method "printStrains"*

---

**Description**

Prints out the genetic background (MGI IDs and strain name) from which the knockout mice were derived for a specific combination of pipeline, procedure and parameter for a phenotyping center.

**Usage**

```
printStrains(PhenCenterName=NULL, PipelineID=NULL, ProcedureID=NULL,  
ParameterID=NULL, n=NULL)
```

**Arguments**

PhenCenterName IMPC phenotyping center; mandatory argument  
PipelineID IMPC pipeline ID; mandatory argument  
ProcedureID IMPC procedure ID; mandatory argument  
ParameterID IMPC parameter ID; mandatory argument  
n print only first n strains; optional argument

**Value**

Returns the character containing IMPC strains (IDs and names of strains)

**Author(s)**

Natalja Kurbatova, Jeremy Mason

**Examples**

```
printStrains("WTSI","MGP_001","IMPC_CBC_001","IMPC_CBC_003_001")
```

# Index

getAlleles, 2  
getGenes, 3  
getIMPCDataset, 4  
getIMPCTable, 5  
getName, 6  
getParameters, 6  
getPhenCenters, 7  
getPipelines, 8  
getProcedures, 8  
getStrains, 9  
getZygositys, 10  
  
printAlleles, 11  
printGenes, 12  
printParameters, 12  
printPhenCenters, 13  
printPipelines, 14  
printProcedures, 14  
printStrains, 15