

# Package ‘xCell2’

February 6, 2025

**Type** Package

**Title** A Tool for Generic Cell Type Enrichment Analysis

**Version** 0.99.108

**Description** xCell2 provides methods for cell type enrichment analysis using cell type signatures.

It includes three main functions -

1. xCell2Train for training custom references objects from bulk or single-cell RNA-seq datasets.
2. xCell2Analysis for conducting the cell type enrichment analysis using the custom reference.
3. xCell2GetLineage for identifying dependencies between different cell types using ontology.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** false

**Depends** R (>= 4.0.0)

**biocViews** GeneExpression, Transcriptomics, Microarray, RNASeq,  
SingleCell, DifferentialExpression, ImmunoOncology,  
GeneSetEnrichment

**Imports** SummarizedExperiment, SingleCellExperiment, Rfast, singscore,  
AnnotationHub, ontologyIndex, tibble, dplyr, BiocParallel,  
Matrix, minpack.lm, pracma, methods, readr, magrittr, progress,  
quadprog

**Suggests** testthat, knitr, rmarkdown, ggplot2, randomForest, tidyR,  
EnhancedVolcano, BiocStyle

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**URL** <https://github.com/AlmogAngel/xCell2>

**BugReports** <https://github.com/AlmogAngel/xCell2/issues>

**git\_url** <https://git.bioconductor.org/packages/xCell2>

**git\_branch** devel

**git\_last\_commit** 0209e7d

**git\_last\_commit\_date** 2025-01-22

**Repository** Bioconductor 3.21

**Date/Publication** 2025-02-05

**Author** Almog Angel [aut, cre] (ORCID: <<https://orcid.org/0009-0001-3297-6935>>),  
Dvir Aran [aut] (ORCID: <<https://orcid.org/0000-0001-6334-5039>>)

**Maintainer** Almog Angel <almog.angel@campus.technion.ac.il>

## Contents

BlueprintEncode.xCell2Ref . . . . .	2
dependencies . . . . .	3
DICE_demo.xCell2Ref . . . . .	4
dice_demo_ref . . . . .	5
genesUsed . . . . .	5
ImmGenData.xCell2Ref . . . . .	6
ImmuneCompendium.xCell2Ref . . . . .	7
LM22.xCell2Ref . . . . .	8
mix_demo . . . . .	9
MouseRNAseqData.xCell2Ref . . . . .	9
PanCancer.xCell2Ref . . . . .	10
params . . . . .	11
signatures . . . . .	12
spillMat . . . . .	13
TabulaMurisBlood.xCell2Ref . . . . .	14
TabulaSapiensBlood.xCell2Ref . . . . .	15
TMECompendium.xCell2Ref . . . . .	16
xCell2Analysis . . . . .	17
xCell2GetLineage . . . . .	19
xCell2Object-class . . . . .	21
xCell2Train . . . . .	22
<b>Index</b>	<b>25</b>

---

BlueprintEncode.xCell2Ref

*Blueprint and ENCODE Projects Reference*

---

### Description

Pre-trained xCell2 reference for use in xCell2Analysis or extending via xCell2Train.

### Usage

```
data(BlueprintEncode.xCell2Ref, package = "xCell2")
```

**Format**

An `xCell2Object` containing:

`params` Linear transformation parameters.  
`signatures` Gene signatures for cell types.  
`dependencies` Cell lineage dependencies.  
`spill_mat` Spillover correction matrix.  
`genes_used` Genes included in the reference.

**Details**

Blueprint and ENCODE Projects Reference (human)

A pre-trained `xCell2` reference object based on the Blueprint and ENCODE projects datasets.

**Source**

Martens JHA and Stunnenberg HG (2013); The ENCODE Project Consortium (2012), curated by Aran D (2019); trained by Angel A, et al. (2024).

**See Also**

[xCell2Analysis](#) and [xCell2Train](#).

---

dependencies	<i>Access Cell Type Dependencies</i>
--------------	--------------------------------------

---

**Description**

Retrieve or assign hierarchical dependencies between cell types for an `xCell2Object`.

Retrieve or assign the hierarchical dependencies between cell types for an `xCell2Object`.

**Usage**

```
getDeps(object)

setDeps(object) <- value

## S4 method for signature 'xCell2Object'
getDeps(object)

## S4 replacement method for signature 'xCell2Object'
setDeps(object) <- value
```

**Arguments**

`object` An `xCell2Object`.  
`value` A list of hierarchical dependencies (for the setter).

**Value**

For 'getDeps', a list of hierarchical dependencies. For 'setDeps<-', the updated [xCell2Object](#).  
For 'getDeps', a list of hierarchical dependencies. For 'setDeps<-', the updated [xCell2Object](#).

**See Also**

[xCell2Object-class](#)

[xCell2Object-class](#)

**Examples**

```
data(DICE_demo.xCell2Ref, package = "xCell2")
setDeps(DICE_demo.xCell2Ref) <- list("Parent" = "ChildType1")
getDeps(DICE_demo.xCell2Ref)
data(DICE_demo.xCell2Ref, package = "xCell2")
setDeps(DICE_demo.xCell2Ref) <- list("ParentType" = "ChildType1")
```

---

DICE\_demo.xCell2Ref     *Demo xCell2 Reference Object*

---

**Description**

Pre-trained xCell2 reference object based on the DICE dataset.

**Usage**

```
data(DICE_demo.xCell2Ref, package = "xCell2")
```

**Format**

An xCell2Object containing:

params    Linear transformation parameters.  
signatures    Cell-type-specific gene signatures.  
dependencies    Cell type lineage dependencies.  
spill\_mat    Spillover correction matrix.  
genes\_used    Genes included in the reference.

**Details**

Demo xCell2 Reference Object from DICE Subset (human)

A demo xCell2 reference object derived from a subset of the DICE dataset. Suitable for demonstrating the use of xCell2Analysis.

**Source**

Schmiedel B et al. (2018).

**See Also**

[xCell2Analysis](#) for enrichment analysis, and [xCell2Train](#) for training custom references.

---

dice\_demo\_ref

*Subset of the DICE Reference*

---

**Description**

Demo reference object derived from the DICE dataset for training xCell2 references.

**Usage**

```
data(dice_demo_ref, package = "xCell2")
```

**Format**

A [SummarizedExperiment](#) object.

**Details**

Subset of the DICE Reference

A subset of the DICE reference stored as a ‘SummarizedExperiment’ object for the xCell 2.0 vignette. This demo reference object demonstrates the use of [xCell2Train](#) for generating a custom xCell2 reference.

**Source**

Schmiedel B et al. (2018).

**See Also**

[xCell2Train](#) for generating references, and [xCell2Analysis](#) for enrichment analysis.

---

genesUsed

*Access Genes Used*

---

**Description**

Retrieve or assign the genes used in training the reference for an [xCell2Object](#).

Retrieve or assign the genes used in training the reference for an [xCell2Object](#).

**Usage**

```
getGenesUsed(object)

setGenesUsed(object) <- value

## S4 method for signature 'xCell2Object'
getGenesUsed(object)

## S4 replacement method for signature 'xCell2Object'
setGenesUsed(object) <- value
```

**Arguments**

object	An <a href="#">xCell2Object</a> .
value	A character vector of genes (for the setter).

**Value**

For 'getGenesUsed', a character vector of genes. For 'setGenesUsed<-', the updated [xCell2Object](#).  
For 'getGenesUsed', a character vector of genes. For 'setGenesUsed<-', the updated [xCell2Object](#).

**See Also**

[xCell2Object-class](#)  
[xCell2Object-class](#)

**Examples**

```
data(DICE_demo.xCell12Ref, package = "xCell12")
setGenesUsed(DICE_demo.xCell12Ref) <- c("GeneA", "GeneB", "GeneC")
getGenesUsed(DICE_demo.xCell12Ref)
data(DICE_demo.xCell12Ref, package = "xCell12")
setGenesUsed(DICE_demo.xCell12Ref) <- c("GeneA", "GeneB", "GeneC")
```

---

ImmGenData.xCell12Ref *Immunologic Genome Project Reference*

---

**Description**

Pre-trained xCell2 reference for use in xCell2Analysis or extending via xCell2Train.

**Usage**

```
data(ImmGenData.xCell12Ref, package = "xCell12")
```

**Format**

An xCell2object containing:

params Linear transformation parameters.  
signatures Gene signatures for cell types.  
dependencies Cell lineage dependencies.  
spill\_mat Spillover correction matrix.  
genes\_used Genes included in the reference.

**Details**

Immunologic Genome Project Reference

A pre-trained xCell2 reference object based on the Immunologic Genome Project dataset.

**Source**

The Immunological Genome Project Consortium (2008), curated by Aran D (2019); trained by Angel A, et al. (2024).

**See Also**

[xCell2Analysis](#) and [xCell2Train](#).

---

ImmuneCompendium.xCell2Ref

*Immune Compendium Reference*

---

**Description**

Pre-trained xCell2 reference for use in xCell2Analysis or extending via xCell2Train.

**Usage**

```
data(ImmuneCompendium.xCell2Ref, package = "xCell2")
```

**Format**

An xCell2object containing:

params Linear transformation parameters.  
signatures Gene signatures for cell types.  
dependencies Cell lineage dependencies.  
spill\_mat Spillover correction matrix.  
genes\_used Genes included in the reference.

**Details**

Immune Compendium Reference (human)

A pre-trained xCell2 reference object based on the Immune Compendium dataset for immune cell profiling.

**Source**

Curated by Zaitsev A (2022); trained by Angel A, et al. (2024).

**See Also**

[xCell2Analysis](#) and [xCell2Train](#).

---

LM22.xCell2Ref

*LM22 Reference*

---

**Description**

Pre-trained xCell2 reference for use in [xCell2Analysis](#) or extending via [xCell2Train](#).

**Usage**

```
data(LM22.xCell2Ref, package = "xCell2")
```

**Format**

An xCell2Object containing:

params Linear transformation parameters.

signatures Gene signatures for cell types.

dependencies Cell lineage dependencies.

spill\_mat Spillover correction matrix.

genes\_used Genes included in the reference.

**Details**

LM22 Reference (human)

A pre-trained xCell2 reference object based on the LM22 dataset.

**Source**

Newman AM (2015); trained by Angel A, et al. (2024).

**See Also**

[xCell2Analysis](#) and [xCell2Train](#).



---

`mix_demo`*Demo Bulk Gene Expression Data*

---

**Description**

Example RNA-Seq data to demonstrate xCell2Analysis.

**Usage**

```
data(mix_demo, package = "xCell2")
```

**Format**

A matrix with genes (rows) and samples (columns).

**Details**

Demo Bulk Gene Expression Data (RNA-Seq)

A demo mixture matrix for bulk RNA-Seq gene expression data. Use this dataset to test xCell2Analysis with pre-trained xCell2 references.

**See Also**

[xCell2Analysis](#) for enrichment analysis.

---

`MouseRNAseqData.xCell2Ref`*Mouse RNA-Seq Data Reference*

---

**Description**

Pre-trained xCell2 reference for use in xCell2Analysis or extending via xCell2Train.

**Usage**

```
data(MouseRNAseqData.xCell2Ref, package = "xCell2")
```

**Format**

An xCell2Object containing:

`params` Linear transformation parameters.

`signatures` Gene signatures for cell types.

`dependencies` Cell lineage dependencies.

`spill_mat` Spillover correction matrix.

`genes_used` Genes included in the reference.

**Details**

Mouse RNA-Seq Data Reference

A pre-trained xCell2 reference object based on the MouseRNAseqData dataset.

**Source**

Benayoun B (2019); trained by Angel A, et al. (2024).

**See Also**

[xCell2Analysis](#) and [xCell2Train](#).

---

PanCancer.xCell2Ref     *PanCancer Reference*

---

**Description**

Pre-trained xCell2 reference for use in xCell2Analysis or extending via xCell2Train.

**Usage**

```
data(PanCancer.xCell2Ref, package = "xCell2")
```

**Format**

An xCell2Object containing:

params Linear transformation parameters.

signatures Gene signatures for cell types.

dependencies Cell lineage dependencies.

spill\_mat Spillover correction matrix.

genes\_used Genes included in the reference.

**Details**

PanCancer Reference (human)

A pre-trained xCell2 reference object based on the PanCancer dataset for cancer-specific analyses.

**Source**

Nofech-Mozes I (2023); trained by Angel A, et al. (2024).

**See Also**

[xCell2Analysis](#) and [xCell2Train](#).

---

params

*Access Transformation Parameters*

---

## Description

Retrieve or assign linear transformation parameters for an [xCell2Object](#).

Retrieve or assign linear transformation parameters for cell types for an [xCell2Object](#).

## Usage

```
getParams(object)
```

```
setParams(object) <- value
```

```
## S4 method for signature 'xCell2Object'  
getParams(object)
```

```
## S4 replacement method for signature 'xCell2Object'  
setParams(object) <- value
```

## Arguments

object            An [xCell2Object](#).

value            A data frame of transformation parameters (for the setter).

## Value

For 'getParams', a data frame of transformation parameters. For 'setParams<-', the updated [xCell2Object](#).

For 'getParams', a data frame of transformation parameters. For 'setParams<-', the updated [xCell2Object](#).

## See Also

[xCell2Object-class](#)

[xCell2Object-class](#)

## Examples

```
data(DICE_demo.xCell2Ref, package = "xCell2")  
setParams(DICE_demo.xCell2Ref) <- data.frame(celltype = "T_cells", a = 0.5, b = 2, m = 1.5)  
getParams(DICE_demo.xCell2Ref)  
data(DICE_demo.xCell2Ref, package = "xCell2")  
setParams(DICE_demo.xCell2Ref) <- data.frame(celltype = "B_cells", a = 0.6, b = 1.8, m = 2.1)
```

---

signatures

*Access Cell Type Signatures*

---

### Description

Retrieve or assign the cell type-specific gene signatures for an [xCell2Object](#).

Retrieve or assign the cell type-specific gene signatures for an [xCell2Object](#).

### Usage

```
getSignatures(object)

setSignatures(object) <- value

## S4 method for signature 'xCell2Object'
getSignatures(object)

## S4 replacement method for signature 'xCell2Object'
setSignatures(object) <- value
```

### Arguments

`object` An [xCell2Object](#).

`value` A list of cell type-specific gene signatures (for the setter).

### Value

For ‘getSignatures’, a list of cell type-specific gene signatures. For ‘setSignatures<-’, the updated [xCell2Object](#).

For ‘getSignatures’, a list of cell type-specific gene signatures. For ‘setSignatures<-’, the updated [xCell2Object](#).

### See Also

[xCell2Object-class](#)

[xCell2Object-class](#)

### Examples

```
data(DICE_demo.xCell2Ref, package = "xCell2")
setSignatures(DICE_demo.xCell2Ref) <- list("T_cells" = c("GeneA", "GeneB"), "B_cells" = c("GeneC"))
getSignatures(DICE_demo.xCell2Ref)
data(DICE_demo.xCell2Ref, package = "xCell2")
setSignatures(DICE_demo.xCell2Ref) <- list("T_cells" = c("GeneA", "GeneB"))
```

---

spillMat	<i>Access Spillover Matrix</i>
----------	--------------------------------

---

### Description

Retrieve or assign the spillover correction matrix for an [xCell2Object](#).

Retrieve or assign the spillover correction matrix for an [xCell2Object](#).

### Usage

```
getSpillMat(object)
```

```
setSpillMat(object) <- value
```

```
## S4 method for signature 'xCell2Object'
```

```
getSpillMat(object)
```

```
## S4 replacement method for signature 'xCell2Object'
```

```
setSpillMat(object) <- value
```

### Arguments

object           An [xCell2Object](#).

value            A matrix of spillover correction factors (for the setter).

### Value

For 'getSpillMat', a matrix of spillover correction factors. For 'setSpillMat<-', the updated [xCell2Object](#).

For 'getSpillMat', a matrix of spillover correction factors. For 'setSpillMat<-', the updated [xCell2Object](#).

### See Also

[xCell2Object-class](#)

[xCell2Object-class](#)

### Examples

```
data(DICE_demo.xCell2Ref, package = "xCell2")
spill_mat <- matrix(c(1, 0.1, 0.1, 1), nrow = 2, byrow = TRUE)
rownames(spill_mat) <- colnames(spill_mat) <- c("T_cells", "B_cells")
setSpillMat(DICE_demo.xCell2Ref) <- spill_mat
getSpillMat(DICE_demo.xCell2Ref)
data(DICE_demo.xCell2Ref, package = "xCell2")
spill_mat <- matrix(c(1, 0.05, 0.05, 1), nrow = 2, byrow = TRUE)
rownames(spill_mat) <- colnames(spill_mat) <- c("T_cells", "B_cells")
setSpillMat(DICE_demo.xCell2Ref) <- spill_mat
```

---

TabulaMurisBlood.xCell2Ref

*Tabula Muris Blood Reference*

---

## Description

Pre-trained xCell2 reference for use in xCell2Analysis or extending via xCell2Train.

## Usage

```
data(TabulaMurisBlood.xCell2Ref, package = "xCell2")
```

## Format

An xCell2Object containing:

params Linear transformation parameters.

signatures Gene signatures for cell types.

dependencies Cell lineage dependencies.

spill\_mat Spillover correction matrix.

genes\_used Genes included in the reference.

## Details

Tabula Muris Blood Reference (mouse)

A pre-trained xCell2 reference object based Tabula Muris dataset.

## Source

The Tabula Muris Consortium (2018); trained by Angel A, et al. (2024).

## See Also

[xCell2Analysis](#) and [xCell2Train](#).

---

`TabulaSapiensBlood.xCell2Ref`*Tabula Sapiens Blood Reference*

---

**Description**

Pre-trained xCell2 reference for use in xCell2Analysis or extending via xCell2Train.

**Usage**

```
data(TabulaSapiensBlood.xCell2Ref, package = "xCell2")
```

**Format**

An xCell2Object containing:

`params` Linear transformation parameters.

`signatures` Gene signatures for cell types.

`dependencies` Cell lineage dependencies.

`spill_mat` Spillover correction matrix.

`genes_used` Genes included in the reference.

**Details**

Tabula Sapiens Blood Reference (human)

A pre-trained xCell2 reference object based on the Tabula Sapiens dataset.

**Source**

The Tabula Sapiens Consortium (2022); trained by Angel A, et al. (2024).

**See Also**

[xCell2Analysis](#) and [xCell2Train](#).

---

TMECompendium.xCell2Ref

*Tumor Microenvironment Compendium Reference*

---

## Description

Pre-trained xCell2 reference object for analyzing tumor microenvironments.

## Usage

```
data(TMECompendium.xCell2Ref, package = "xCell2")
```

## Format

An xCell2Object containing:

params Linear transformation parameters.

signatures Cell-type-specific gene signatures.

dependencies Cell type lineage dependencies.

spill\_mat Spillover correction matrix.

genes\_used Genes included in the reference.

## Details

Tumor Microenvironment Compendium Reference (human)

An xCell2 reference object created from the Tumor Microenvironment Compendium dataset.

Normalized data for training can be accessed at: <https://science.bostongene.com/kassandra/downloads>.

## Source

Curated by Zaitsev A (2022) and trained by Angel A, et al. (2024).

## References

Zaitsev, A., et al. (2022). Cancer Cell, 40(8), 879-894.

## See Also

[xCell2Analysis](#) and [xCell2Train](#).



---

xCell2Analysis      *Perform Cell Type Enrichment Analysis*

---

## Description

Estimates the relative enrichment of cell types in a bulk gene expression mixture. This function uses gene signatures from a pre-trained xCell2Object to compute enrichment scores, with options for linear transformation and spillover correction to improve specificity.

## Usage

```
xCell2Analysis(  
  mix,  
  xcell2object,  
  minSharedGenes = 0.9,  
  rawScores = FALSE,  
  spillover = TRUE,  
  spilloverAlpha = 0.5,  
  BPPARAM = BiocParallel::SerialParam()  
)
```

## Arguments

mix	A bulk mixture of gene expression data (genes in rows, samples in columns). The input must use the same gene annotation system as the reference object.
xcell2object	A pre-trained reference object of class xCell2Object, created using <a href="#">xCell2Train</a> . Pre-trained references for common cases are provided within the package.
minSharedGenes	Minimum fraction of shared genes required between the mixture and the reference object (default: 0.9). If the shared fraction is below this threshold, the function stops with an error or warning, as sufficient overlap is necessary for accurate analysis.
rawScores	Logical; if TRUE, returns raw enrichment scores without transformation or spillover correction (default: FALSE).
spillover	Logical; enables spillover correction on enrichment scores (default: TRUE). Spillover occurs when closely related cell types share gene expression patterns, inflating enrichment scores. Correction enhances specificity, particularly for related cell types.
spilloverAlpha	Numeric value controlling spillover correction strength (default: 0.5). Lower values apply weaker correction, while higher values apply stronger correction.
BPPARAM	A <a href="#">BiocParallelParam</a> instance to define parallelization strategy (see "Details"). Default is <code>BiocParallel::SerialParam()</code> .

## Details

The `xCell2Analysis` function performs cell type enrichment analysis by leveraging gene signatures from a pre-trained `xCell2Object`. It computes enrichment scores for each cell type in the provided bulk gene expression mixture (`mix`), applies linear transformations, and corrects for spillover. Spillover correction addresses the overlap of gene expression patterns between closely related cell types, improving the specificity of the enrichment scores.

**## Parallelization with BPPARAM** To achieve faster processing by running computations in parallel, `xCell2Analysis` supports parallelization through the `BPPARAM` parameter. Users can define a parallelization strategy using `BiocParallelParam` from the `BiocParallel` package. For example, `MulticoreParam` is suitable for multi-core processing on Linux and macOS, while `SnowParam` or `SerialParam` are better suited for Windows systems. Refer to the [BiocParallel documentation](#) for further guidance on parallelization strategies.

**## Relationship with Other Function(s)** The pre-trained `xCell2Object` used in `xCell2Analysis` is created via the `xCell2Train` function.

The `xCell2Analysis` function computes enrichment scores for each cell type using gene signatures from a pre-trained `xCell2Object`. Linear transformations and spillover corrections refine the results, improving specificity when cell types have overlapping gene expression patterns.

**Parallelization with BPPARAM:** Computations can be parallelized using the `BPPARAM` parameter. Supported strategies include:

- `MulticoreParam` for multi-core processing (Linux/macOS).
- `SnowParam` or `SerialParam` for Windows systems.

See the [BiocParallel documentation](#).

**Relationship with Other Functions:** The input reference object (`xCell2Object`) is created via `xCell2Train`.

## Value

A data frame containing the cell type enrichment for each sample in the input matrix, as estimated by `xCell2`. Each row corresponds to a cell type, and each column corresponds to a sample.

A data frame containing enrichment scores for each cell type and sample. Rows correspond to cell types and columns to samples.

## Author(s)

Almog Angel and Dvir Aran

## See Also

`xCell2Train`, for creating the reference object used in this analysis.

## Examples

```
# For detailed example read xCell2 vignette.  
  
library(xCell2)
```

```

# Load "ready to use" xCell2 reference object or generate a new one using `xCell2Train`
data(DICE_demo.xCell2Ref, package = "xCell2")

# Load demo bulk RNA-Seq gene expression mixture

# For detailed examples, see the xCell2 vignette.

library(xCell2)

# Load pre-trained reference object
data(DICE_demo.xCell2Ref, package = "xCell2")

# Load demo bulk gene expression mixture
data(mix_demo, package = "xCell2")

# Perform cell type enrichment analysis
xcell2_res <- xCell2::xCell2Analysis(
  mix = mix_demo,
  xcell2object = DICE_demo.xCell2Ref
)

# Parallel processing example with BiocParallel
library(BiocParallel)
parallel_param <- MulticoreParam(workers = 2)
xcell2_res_parallel <- xCell2::xCell2Analysis(
  mix = mix_demo,
  xcell2object = DICE_demo.xCell2Ref,
  BPPARAM = parallel_param
)

```

---

xCell2GetLineage

*Identify Cell Type Lineage Dependencies*


---

## Description

Identifies cell type dependencies based on the Cell Ontology, including both descendants and ancestors for each cell type. Enables manual inspection and refinement of lineage relationships to improve biological accuracy in xCell2 analyses.

## Usage

```
xCell2GetLineage(labels, outFile = NULL)
```

## Arguments

labels	A data frame with the following required columns: <ul style="list-style-type: none"> <li>"ont": Cell type ontology ID (e.g., "CL:0000545"). Use NA if unavailable. Ontologies can be accessed via <a href="#">EBI Ontology Lookup Service (OLS)</a> or the <a href="#">ontologyIndex</a> package.</li> </ul>
--------	--

	<ul style="list-style-type: none"><li>• "label": Cell type name (e.g., "T-helper 1 cell").</li><li>• "sample": Sample or cell identifier matching column names in the gene expression matrix.</li><li>• "dataset": Dataset or subject source. Use a constant value if not applicable.</li></ul>
outFile	Optional. Output file name for saving dependencies as a TSV file. The file includes columns for "ont", "label", "descendants", and "ancestors". Suitable for manual inspection and refinement before use in downstream analyses.

## Details

The `xCell2GetLineage` function generates lineage relationships for cell types based on the Cell Ontology. These relationships refine lineage-based dependencies, improving the biological relevance of gene signatures. Users can:

- Use the generated TSV file for manual adjustments before training custom references via [xCell2Train](#).
- Skip this step entirely, allowing `xCell2Train` to infer dependencies automatically.

If no ontology IDs ("ont") are provided, the function outputs empty dependencies with a message for user guidance.

### Relationship with Other Functions:

- [xCell2Train](#): Incorporates lineage relationships during reference training.
- [xCell2Analysis](#): Uses trained references for enrichment analysis.

## Value

If `outFile` is:

- NULL: Returns a list of dependencies for each cell type, with descendants and ancestors as components.
- Specified: Writes a TSV file and warns the user to inspect and validate results manually.

## Author(s)

Almog Angel and Dvir Aran

## See Also

[xCell2Train](#) for training custom references with lineage data. [xCell2Analysis](#) for enrichment analysis using trained references. [AnnotationHub](#) to access ontology data. [ontologyIndex](#) to programmatically explore ontologies.

**Examples**

```

# For detailed examples, see the xCell2 vignette.

library(xCell2)

# Load demo reference object
data(dice_demo_ref, package = "xCell2")

# Prepare labels data frame
dice_labels <- SummarizedExperiment::colData(dice_demo_ref)
dice_labels <- as.data.frame(dice_labels)
dice_labels$ont <- NA
dice_labels$sample <- colnames(dice_demo_ref)
dice_labels$dataset <- "DICE"

# Assign ontology IDs
dice_labels[dice_labels$label == "B cells", ]$ont <- "CL:0000236"
dice_labels[dice_labels$label == "Monocytes", ]$ont <- "CL:0000576"
dice_labels[dice_labels$label == "NK cells", ]$ont <- "CL:0000623"
dice_labels[dice_labels$label == "T cells, CD8+", ]$ont <- "CL:0000625"
dice_labels[dice_labels$label == "T cells, CD4+", ]$ont <- "CL:0000624"
dice_labels[dice_labels$label == "T cells, CD4+, memory", ]$ont <- "CL:0000897"

# Generate cell type lineage dependencies
xCell2::xCell2GetLineage(labels = dice_labels)

# Manually inspect and adjust saved dependencies for refined lineage relationships
# Use the adjusted file as input to xCell2Train via the `lineageFile` parameter.

```

---

xCell2Object-class      *xCell2Object Class*

---

**Description**

An S4 class to represent the xCell2 reference object. This object contains cell type-specific gene signatures, hierarchical dependencies, linear transformation parameters, spillover correction matrices, and genes used for training.

**Slots**

**signatures** A list of cell type-specific gene signatures.

**dependencies** A list of hierarchical dependencies between cell types.

**params** A data frame containing linear transformation parameters for cell types.

**spill\_mat** A matrix containing spillover correction factors for cell types.

**genes\_used** A character vector of genes used for training the xCell2 reference object.

---

xCell2Train

*Train Custom xCell2 Reference Object*


---

## Description

This function creates a custom reference object for `xCell2Analysis`, enabling cell type enrichment analysis. It supports references derived from RNA-Seq, microarray, and scRNA-Seq data and can be derived from various tissues and organisms.

## Usage

```
xCell2Train(
  ref,
  mix = NULL,
  labels = NULL,
  refType,
  lineageFile = NULL,
  BPPARAM = BiocParallel::SerialParam(),
  useOntology = TRUE,
  returnSignatures = FALSE,
  returnAnalysis = FALSE,
  useSpillover = TRUE,
  spilloverAlpha = 0.5,
  minPbCells = 30,
  minPbSamples = 10,
  minScGenes = 10000
)
```

## Arguments

ref	<p>A reference gene expression matrix (genes in rows, samples/cells in columns) or a <a href="#">SummarizedExperiment/SingleCellExperiment</a> object with expression data in the assays slot.</p> <p><b>Valid Assays:</b></p> <p>"tpm" Transcripts Per Million (recommended for RNA-Seq).</p> <p>"logcounts" Log-transformed normalized counts.</p> <p>"normcounts" Normalized counts.</p> <p>"counts" Raw counts (required for microarray references).</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"> <li>• If multiple assays exist, "tpm" is prioritized.</li> <li>• For microarray data, the "counts" assay must be used.</li> </ul>
mix	<p>A bulk mixture of gene expression matrix (genes in rows, samples in columns) (optional). This parameter is required if <code>returnAnalysis</code> is set to TRUE, as it is used for enrichment analysis.</p>
labels	<p>A data frame with the following columns:</p>

- "ont": The cell type ontology ID (e.g., "CL:0000545"). Set to NA if not available. Ontologies can be found at [EBI Ontology Lookup Service \(OLS\)](#) or by using the [ontologyIndex](#) package.
- "label": The cell type name (e.g., "T-helper 1 cell").
- "sample": The sample or cell identifier, matching column names in the reference matrix.
- "dataset": The dataset source for each sample. If not applicable, use a constant value for all samples.

This parameter is unnecessary if ref is a SummarizedExperiment or SingleCellExperiment object, as metadata should be in colData.

refType	The type of reference data: "rnaseq" for RNA-Seq, "array" for microarray, or "sc" for scRNA-Seq.
lineageFile	Path to a manually curated cell type lineage file generated with <a href="#">xCell2GetLineage</a> (optional).
BPPARAM	A <a href="#">BiocParallelParam</a> instance that determines the parallelization strategy (more in "Details"). Default is <code>BiocParallel::SerialParam()</code> .
useOntology	A Boolean indicating whether to use ontological integration for cell type dependencies (default: TRUE). Lineage relationships are determined using the Cell Ontology (CL). Users can refine these dependencies with <a href="#">xCell2GetLineage</a> and provide them via the lineageFile parameter.
returnSignatures	A Boolean to return only cell type signatures (default: FALSE).
returnAnalysis	A Boolean to return <a href="#">xCell2Analysis</a> results instead of a reference object (default: FALSE).
useSpillover	A Boolean to use spillover correction during analysis when returnAnalysis is TRUE (default: TRUE). Spillover correction enhances the specificity of enrichment scores by accounting for overlaps between cell types.
spilloverAlpha	Numeric value controlling spillover correction strength (default: 0.5). Lower values apply weaker correction, while higher values apply stronger correction.
minPbCells	Minimum number of cells in a pseudo-bulk sample for scRNA-Seq references (default: 30).
minPbSamples	Minimum number of pseudo-bulk samples for scRNA-Seq references (default: 10).
minScGenes	Minimum number of genes for pseudo-bulk samples for scRNA-Seq references (default: 1e4).

## Details

**Ontological Integration:** Ontological integration (useOntology) leverages hierarchical cell type relationships to ensure biologically meaningful signatures. Dependencies can be refined using [xCell2GetLineage](#), which generates lineage files for manual review.

**Spillover Correction:** Spillover correction enhances the specificity of enrichment scores by reducing overlaps between related cell types. Use the spilloverAlpha parameter to tune the strength of correction.

**Contribute Your xCell2 Reference Object:** Users are encouraged to share their reference objects via the [xCell2 Reference Repository](#).

**Value**

An xCell2Object containing:

- **signatures**: Cell type-specific gene signatures.
- **dependencies**: Lineage-based dependencies.
- **params**: Linear transformation parameters.
- **spill\_mat**: A spillover correction matrix.
- **genes\_used**: Genes used for training.

**Author(s)**

Almog Angel and Dvir Aran

**See Also**

[xCell2Analysis](#), for enrichment analysis. [xCell2GetLineage](#), for refining cell type dependencies.

**Examples**

```
library(xCell2)
data(dice_demo_ref, package = "xCell2")
dice_ref <- SummarizedExperiment::assay(dice_demo_ref, "logcounts")
colnames(dice_ref) <- make.unique(colnames(dice_ref))
dice_labels <- as.data.frame(SummarizedExperiment::colData(dice_demo_ref))
dice_labels$ont <- NA
dice_labels$sample <- colnames(dice_ref)
dice_labels$dataset <- "DICE"
DICE.xCell2Ref <- xCell2::xCell2Train(ref = dice_ref, labels = dice_labels, refType = "rnaseq")
```



# Index

- \* **datasets**,
  - DICE\_demo.xCell2Ref, 4
  - dice\_demo\_ref, 5
  - mix\_demo, 9
  - TMECompendium.xCell2Ref, 16
- \* **datasets**
  - BlueprintEncode.xCell2Ref, 2
  - ImmGenData.xCell2Ref, 6
  - ImmuneCompendium.xCell2Ref, 7
  - LM22.xCell2Ref, 8
  - MouseRNAseqData.xCell2Ref, 9
  - PanCancer.xCell2Ref, 10
  - TabulaMurisBlood.xCell2Ref, 14
  - TabulaSapiensBlood.xCell2Ref, 15
- \* **examples**
  - mix\_demo, 9
- \* **references**
  - DICE\_demo.xCell2Ref, 4
  - dice\_demo\_ref, 5
  - TMECompendium.xCell2Ref, 16
- AnnotationHub, 20
- BiocParallelParam, 17, 23
- BlueprintEncode.xCell2Ref, 2
- dependencies, 3
- DICE\_demo.xCell2Ref, 4
- dice\_demo\_ref, 5
- genesUsed, 5
- getDeps (dependencies), 3
- getDeps,xCell2Object-method (dependencies), 3
- getGenesUsed (genesUsed), 5
- getGenesUsed,xCell2Object-method (genesUsed), 5
- getParams (params), 11
- getParams,xCell2Object-method (params), 11
- getSignatures (signatures), 12
- getSignatures,xCell2Object-method (signatures), 12
- getSpillMat (spillMat), 13
- getSpillMat,xCell2Object-method (spillMat), 13
- ImmGenData.xCell2Ref, 6
- ImmuneCompendium.xCell2Ref, 7
- LM22.xCell2Ref, 8
- mix\_demo, 9
- MouseRNAseqData.xCell2Ref, 9
- MulticoreParam, 18
- ontologyIndex, 19, 20, 23
- PanCancer.xCell2Ref, 10
- params, 11
- SerialParam, 18
- setDeps<- (dependencies), 3
- setDeps<-,xCell2Object-method (dependencies), 3
- setGenesUsed<- (genesUsed), 5
- setGenesUsed<-,xCell2Object-method (genesUsed), 5
- setParams<- (params), 11
- setParams<-,xCell2Object-method (params), 11
- setSignatures<- (signatures), 12
- setSignatures<-,xCell2Object-method (signatures), 12
- setSpillMat<- (spillMat), 13
- setSpillMat<-,xCell2Object-method (spillMat), 13
- signatures, 12
- SingleCellExperiment, 22
- SnowParam, 18
- spillMat, 13

SummarizedExperiment, [5](#), [22](#)

TabulaMurisBlood.xCell2Ref, [14](#)

TabulaSapiensBlood.xCell2Ref, [15](#)

TMECompendium.xCell2Ref, [16](#)

xCell2Analysis, [3](#), [5](#), [7–10](#), [14–16](#), [17](#), [20](#),  
[22–24](#)

xCell2GetLineage, [19](#), [23](#), [24](#)

xCell2Object, [3–6](#), [11–13](#)

xCell2Object-class, [4](#), [6](#), [11–13](#), [21](#)

xCell2Train, [3](#), [5](#), [7](#), [8](#), [10](#), [14–18](#), [20](#), [22](#)