

immunoClust - Automated Pipeline for Population Detection in Flow Cytometry

*Till Sörensen*¹

May 1, 2024

Contents

1	Licensing	2
2	Overview.	2
3	Getting started	2
4	Example Illustrating the immunoClust Pipeline	2
4.1	Cell Event Clustering	3
4.2	Meta Clustering	6
4.3	Meta Annotation	7
5	Session Info	10

¹till-antoni.soerensen@charite.de

1 Licensing

Under the Artistic License, you are free to use and redistribute this software. However, we ask you to cite the following paper if you use this software for publication.

Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. and Häupl, T.
 immunoClust - an automated analysis pipeline for the identification of
 immunophenotypic signatures in high-dimensional cytometric datasets.
Cytometry A (accepted).

2 Overview

immunoClust presents an automated analysis pipeline for uncompensated fluorescence and mass cytometry data and consists of two parts. First, cell events of each sample are grouped into individual clusters (cell-clustering). Subsequently, a classification algorithm assorts these cell event clusters into populations comparable between different samples (meta-clustering). The clustering of cell events is designed for datasets with large event counts in high dimensions as a global unsupervised method, sensitive to identify rare cell types even when next to large populations. Both parts use model-based clustering with an iterative Expectation Maximization (EM) algorithm and the Integrated Classification Likelihood (ICL) to obtain the clusters.

The cell-clustering process fits a mixture model with *t*-distributions. Within the clustering process a optimisation of the *asinh*-transformation for the fluorescence parameters is included.

The meta-clustering fits a Gaussian mixture model for the meta-clusters, where adjusted Bhattacharyya-Coefficients give the probability measures between cell- and meta-clusters.

Several plotting routines are available visualising the results of the cell- and meta-clustering process. Additional helper-routines to extract population features are provided.

3 Getting started

The installation on *immunoClust* is normally done within the Bioconductor.

The core functions of *immunoClust* are implemented in C/C++ for optimal utilization of system resources and depend on the GNU Scientific Library (GSL) and Basic Linear Subprogram (BLAS). When installing *immunoClust* from source using Rtools be aware to adjust the GSL library and include pathes in `src/Makevars.in` or `src/Makevars.win` (on Windows systems) repectively to the correct installation directory of the GSL-library on the system.

immunoClust relies on the *flowFrame* structure imported from the *flowCore*-package for accessing the measured cell events from a flow cytometer device.

4 Example Illustrating the immunoClust Pipeline

The functionality of the immunoClust pipeline is demonstrated on a dataset of blood cell samples of defined composition that were depleted of particular cell subsets by magnetic cell sorting. Whole blood leukocytes taken from three healthy individuals, which were experimen-

tally modified by the depletion of one particular cell type per sample, including granulocytes (using CD15-MACS-beads), monocytes (using CD14-MACS-beads), T lymphocytes (CD3-MACS-beads), T helper lymphocytes (using CD4-MACS-beads) and B lymphocytes (using CD19-MACS-beads).

The example datasets contain reduced (10.000 cell-events) of the first Flow Cytometry (FC) sample in `dat.fcs` and the *immunoClust* cell-clustering results of all 5 reduced FC samples for the first donor in `dat.exp`. The full sized dataset is published and available under <http://flowrepository.org/id/FR-FCM-ZZWB>.

4.1 Cell Event Clustering

```
> library(immunoClust)
```

The cell-clustering is performed by the `cell.process` function for each FC sample separately. Its major input are the measured cell-events in a *flowFrame*-object imported from the *flowCore*-package.

```
> data(dat.fcs)
> dat.fcs

flowFrame object '2d36b4cf-da0f-4b8d-9a4c-fc7e4f5fccc8'
with 10000 cells and 7 observables:

      name desc  range minRange maxRange
$P2      FSC-A  NA   262144     0.00   262143
$P5      SSC-A  NA   262144    -111.00  262143
$P8      FITC-A CD14   262144    -111.00  262143
$P9       PE-A CD19   262144    -111.00  262143
$P12     APC-A CD15   262144    -111.00  262143
$P13  APC-Cy7-A CD4   262144    -111.00  262143
$P14 Pacific Blue-A CD3  262144    -98.94  262143
171 keywords are stored in the 'description' slot
```

In the `parameters` argument the parameters (named as observables in the *flowFrame*) used for cell-clustering are specified. When omitted all determined parameters are used.

```
> pars=c("FSC-A", "SSC-A", "FITC-A", "PE-A", "APC-A", "APC-Cy7-A", "Pacific Blue-A")
> res.fcs <- cell.process(dat.fcs, parameters=pars)
```

The `summary` method for an *immunoClust*-object gives an overview of the clustering results.

```
> summary(res.fcs)

** Experiment Information **
Experiment name: 12443.fcs
Data Filename:   fcs/12443.fcs
Parameters:      FSC-A SSC-A FITC-A PE-A APC-A APC-Cy7-A Pacific Blue-A
Description:     NA NA CD14 CD19 CD15 CD4 CD3

** Data Information **
Number of observations: 10000
Number of parameters:   7
Removed from above:     318 (3.18%)
```

```

Removed from below:    0 (0%)

** Transformation Information **
htrans-A:  0.000000 0.000000 0.010000 0.010000 0.010000 0.010000 0.010000
htrans-B:  0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
htrans-decade:  -1

** Clustering Summary **
ICL bias: 0.30
Number of clusters: 13
Cluster      Proportion  Observations
    1         0.087743         860
    2         0.029863         279
    3         0.040332         391
    4         0.021595         211
    5         0.025424         247
    6         0.010183         101
    7         0.037431         366
    8         0.054961         527
    9         0.006017          58
   10         0.637931        6170
   11         0.012622         122
   12         0.007336          70
   13         0.028563         280

    Min.         0.006017          58
    Max.         0.637931        6170

** Information Criteria **
Log likelihood: -254010.2 -254253.6 -172928.2
BIC: -254010.2
ICL: -254253.6

```

With the `bias` argument of the `cell.process` function the number of clusters in the final model is controlled.

```

> res2 <- cell.process(dat.fcs, bias=0.25)
> summary(res2)

** Experiment Information **
Experiment name: 12443.fcs
Data Filename:   fcs/12443.fcs
Parameters:      FSC-A SSC-A FITC-A PE-A APC-A APC-Cy7-A Pacific Blue-A
Description:     NA NA CD14 CD19 CD15 CD4 CD3

** Data Information **
Number of observations: 10000
Number of parameters:   7
Removed from above:     318 (3.18%)
Removed from below:     0 (0%)

** Transformation Information **

```

```

htrans-A:  0.000000 0.000000 0.010000 0.010000 0.010000 0.010000 0.010000
htrans-B:  0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
htrans-decade:  -1

```

**** Clustering Summary ****

ICL bias: 0.25

Number of clusters: 16

Cluster	Proportion	Observations
1	0.021740	210
2	0.026293	255
3	0.009076	89
4	0.020489	199
5	0.019864	192
6	0.075707	744
7	0.038262	361
8	0.036697	360
9	0.055588	533
10	0.005985	58
11	0.638144	6175
12	0.009659	95
13	0.003751	35
14	0.002865	26
15	0.028561	280
16	0.007319	70
Min.	0.002865	26
Max.	0.638144	6175

**** Information Criteria ****

Log likelihood: -254186.2 -254480.7 -172660.9

BIC: -254186.2

ICL: -254480.7

An ICL-bias of 0.3 is reasonable for fluorescence cytometry data based on our experiences, whereas the number of clusters increase dramatically when a `bias` below 0.2 is applied. A principal strategy for the ICL-bias in the whole pipeline is the use of a moderately small `bias` (0.2 - 0.3) for cell-clustering and to optimise the `bias` on meta-clustering level to retrieve the common populations across all samples.

For plotting the clustering results on cell event level, the optimised *asinh*-transformation has to be applied to the raw FC data first.

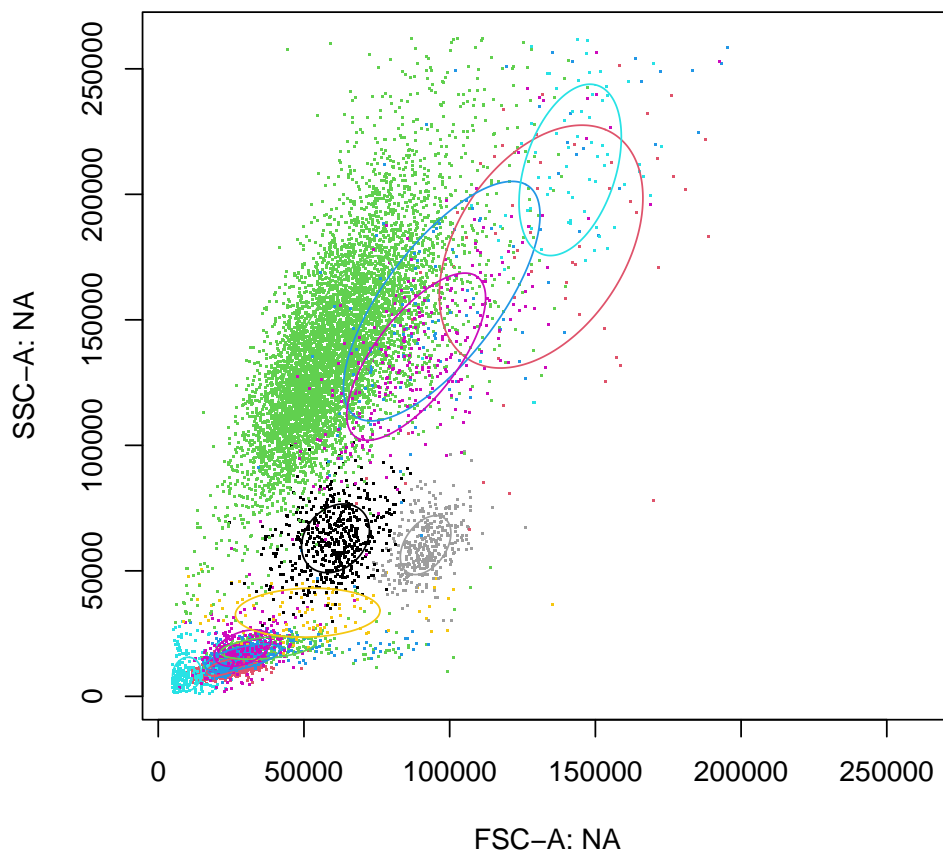
```
> dat.transformed <- trans.ApplyToData(res.fcs, dat.fcs)
```

A scatter plot matrix of all used parameters for clustering is obtained by the `splom` method.

```
> splom(res.fcs, dat.transformed, N=1000)
```

For a scatter plot of 2 particular parameters the `plot` method can be used, where parameters of interest are specified in the `subset` argument.

```
> plot(res.fcs, data=dat.transformed, subset=c(1,2))
```



4.2 Meta Clustering

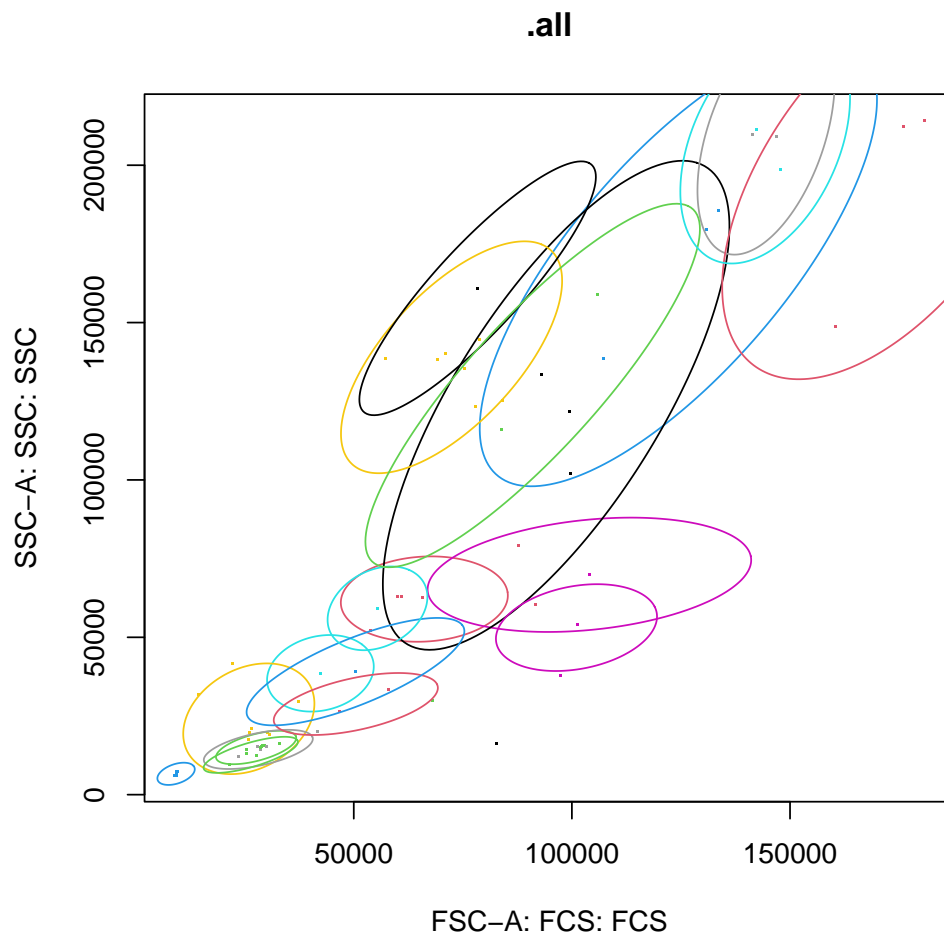
For meta-clustering the cell-clustering results of all FC samples obtained by the `cell.process` function are collected in a vector of *immunoClust*-objects and processed by the `meta.process` function.

```
> data(dat.exp)
> meta<-meta.process(dat.exp, meta.bias=0.3)
```

The obtained *immunoMeta*-object contains the meta-clustering result in `$res.clusters`, and the used cell-clusters information in `$dat.clusters`. Additionally, the clusters can be structures manually in a hierarchical mannner using methods of the *immunoMeta*-object.

A scatter plot matrix of the meta-clustering is obtained by the `plot` method.

```
> plot(meta, c(), plot.subset=c(1,2))
```



In these scatter plots each cell-cluster is marked by a point of its centre. With the default `plot.ellipse=TRUE` argument the meta-clusters are outlined by ellipses of the 90% quantile.

4.3 Meta Annotation

We take a look and first sort the meta-clusters according to the scatter parameter into five major areas

```
> cls <- clusters(meta,c())
> inc <- mu(meta,cls,1) > 20000 & mu(meta,cls,1) < 150000
> addLevel(meta,c(1),"leucocytes") <- cls[inc]
> cls <- clusters(meta,c(1))
> sort(mu(meta,cls,2))
```

cls-10	cls-7	cls-2	cls-6	cls-9	cls-4	cls-11	cls-13
12653.96	14379.67	14987.31	24119.51	28807.31	38562.50	39089.67	53084.06
cls-12	cls-1	cls-5	cls-8	cls-18	cls-14	cls-16	cls-3

immunoClust

```

59202.13  62146.93  69861.55 123717.22 130008.05 138973.89 160894.71 168585.76
  cls-20    cls-15
206575.52 209360.68

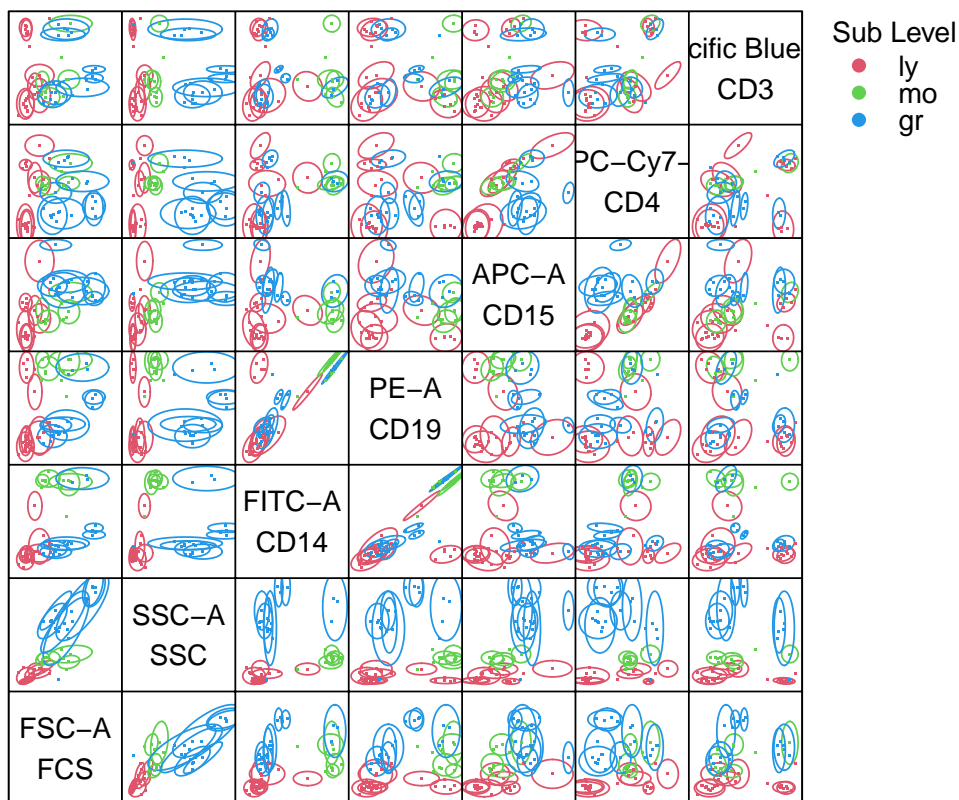
> inc <- (mu(meta,cls,2)) < 40000
> addLevel(meta,c(1,1), "ly") <- cls[inc]
> addLevel(meta,c(1,2), "mo") <- c()
> inc <- (mu(meta,cls,2)) > 100000
> addLevel(meta,c(1,3), "gr") <- cls[inc]
> move(meta,c(1,2)) <- unclassified(meta,c(1))

```

In the plot of this level the three major scatter population are seen easily

```
> plot(meta, c(1))
```

1.all_leucocytes



and we identify the clusters for the particular populations successively by their expression levels.

```

> cls <- clusters(meta,c(1,1))
> sort(mu(meta,cls,7)) ## CD3 expression

```



```

      cls-6   cls-7   cls-9   cls-4   cls-11   cls-2   cls-10
1.017751 1.023148 1.501441 2.043337 2.686877 5.339878 5.503499

> sort(mu(meta,cls,6)) ## CD4 expression

      cls-2   cls-7   cls-6   cls-9   cls-4   cls-10   cls-11
0.3526607 0.4631971 0.5680941 3.0448631 3.3933842 4.1704618 5.3378243

> inc <- mu(meta,cls,7) > 5 & mu(meta,cls,6) > 4
> addLevel(meta,c(1,1,1), "CD3+CD4+") <- cls[inc]
> inc <- mu(meta,cls,7) > 5 & mu(meta,cls,6) < 4
> addLevel(meta,c(1,1,2), "CD3+CD4-") <- cls[inc]
> cls <- unclassified(meta,c(1,1))
> inc <- (mu(meta,cls,4)) > 3
> addLevel(meta,c(1,1,3), "CD19+") <- cls[inc]
> cls <- clusters(meta,c(1,2))
> inc <- mu(meta,cls,3) > 5 & mu(meta,cls,7) < 5
> addLevel(meta,c(1,2,1), "CD14+") <- cls[inc]
> cls <- clusters(meta,c(1,3))
> inc <- mu(meta,cls,5) > 3 & mu(meta,cls,7) < 5
> addLevel(meta,c(1,3,1), "CD15+") <- cls[inc]

```

The whole analysis is performed on uncompensated FC data, thus the high CD19 values on the CD14-population is explained by spillover of FITC into PE.

The event numbers of each meta-cluster and each sample are extracted in a numeric matrix by the `meta.numEvents` function.

```

> tbl <- meta.numEvents(meta, out.unclassified=FALSE)
> tbl[,1:5]

      12543 12546 12549 12552 12555
measured 10000 10000 10000 10000 10000
.all      9682  9842  9736  9736  9510
1.all_leucocytes 9531  9244  9479  9489  9232
1.1.all_leucocytes_ly 1911  6663  3391  1291  771
1.1.1.all_leucocytes_ly_CD3+CD4+ 1107  3425  1585    0    0
1.1.2.all_leucocytes_ly_CD3+CD4-  389  1079   574  433   46
1.1.3.all_leucocytes_ly_CD19+    0   926   452  331  325
1.2.all_leucocytes_mo  898  2472    0   761  950
1.2.1.all_leucocytes_mo_CD14+  898  2370    0   761  950
1.3.all_leucocytes_gr  6722  109  6088  7437  7511
1.3.1.all_leucocytes_gr_CD15+  6459  101  5849  7280  7417

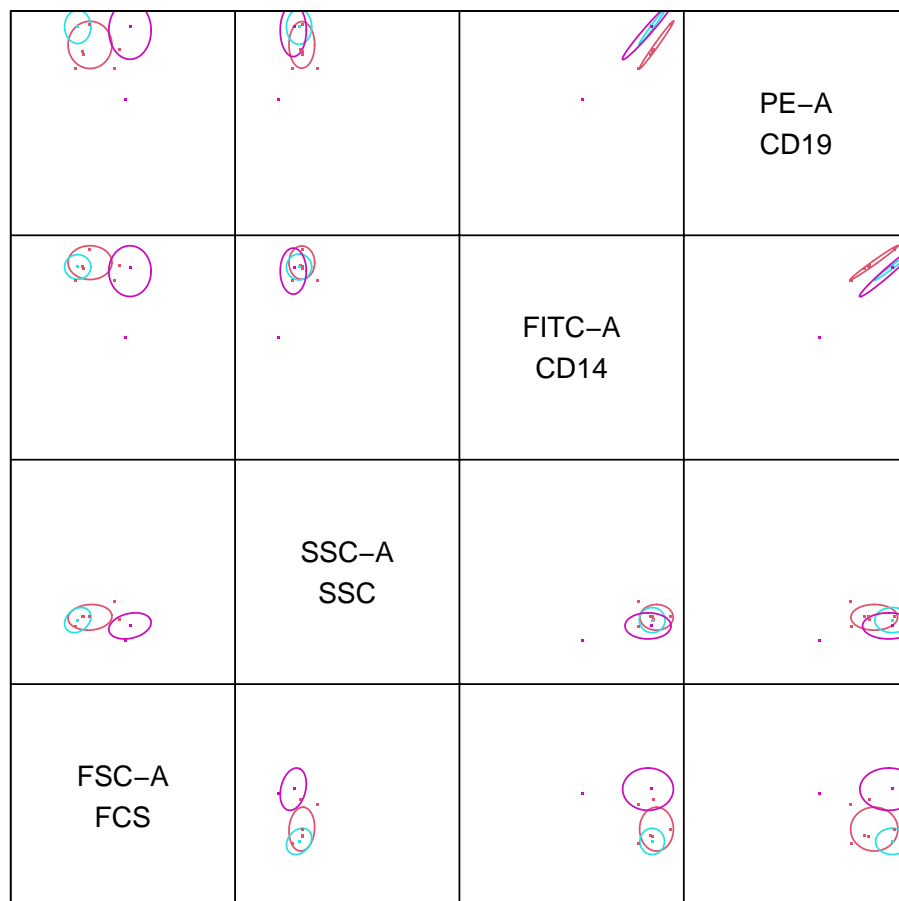
```

Each row denotes an annotated hierarchical level or/and meta-cluster and each column a data sample used in meta-clustering. The row names give the annotated population name. In the last columns additionally the meta-cluster centre values in each parameter are given, which helps to identify the meta-clusters. Further export functions retrieve relative cell event frequencies and sample meta-cluster centre values in a particular parameter.

We see here, that for sample 12546 where the CD15-cells are depleted, the CD14-population is missing. Anyway, this missing cluster could be in the so far unclassified clusters.

```
> plot(meta, c(1,2,1), plot.subset=c(1,2,3,4))
```

1.2.1.all_leucocytes_mo_CD14+



We see the CD14 population of sample 12546 shifted in FSC and CD3 expression levels, probably due to technical variation in the measurement of the CD15-depleted sample, where the granulocytes are missing which constitute about 60% - 70% of the events in the other samples.

5 Session Info

The documentation and example output was compiled and obtained on the system:

```
> toLatex(sessionInfo())
```

- R version 4.4.0 beta (2024-04-15 r86425 ucrt), x86_64-w64-mingw32
- Locale: LC_COLLATE=C, LC_CTYPE=English_United States.utf8, LC_MONETARY=English_United States.utf8, LC_NUMERIC=C, LC_TIME=English_United States.utf8

immunoClust

- Time zone: America/New_York
- TZcode source: internal
- Running under: Windows Server 2022 x64 (build 20348)
- Matrix products: default
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: flowCore 2.16.0, immunoClust 1.36.0
- Loaded via a namespace (and not attached): Biobase 2.64.0, BiocGenerics 0.50.0, BiocManager 1.30.22, BiocStyle 2.32.0, RProtoBufLib 2.16.0, S4Vectors 0.42.0, cli 3.6.2, compiler 4.4.0, cytolib 2.16.0, digest 0.6.35, evaluate 0.23, fastmap 1.1.1, grid 4.4.0, htmltools 0.5.8.1, knitr 1.46, lattice 0.22-6, matrixStats 1.3.0, rlang 1.1.3, rmarkdown 2.26, stats4 4.4.0, tools 4.4.0, xfun 0.43, yaml 2.3.8