

# immunoClust - Automated Pipeline for Population Detection in Flow Cytometry

*Till Sörensen*<sup>1</sup>

<sup>1</sup>till-  
antoni.soerensen@charite.de

April 26, 2022

## 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 . . . . .	7
4.3	Meta Annotation . . . . .	8
5	Session Info . . . . .	13

## 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* form 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%)
```

## immunoClust

```
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         0.033716      330
   2         0.016084      157
   3         0.006989       68
   4         0.040362      391
   5         0.035831      354
   6         0.054087      518
   7         0.003976       37
   8         0.005129       50
   9         0.114607     1111
  10         0.637713     6166
  11         0.028584      281
  12         0.015600      149
  13         0.007322       70

   Min.      0.003976       37
   Max.      0.637713     6166

** Information Criteria **
Log likelihood: -254197.6 -254348.8 -173023.4
BIC: -254197.6
ICL: -254348.8
```

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 **
```

## immunoClust

```
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: 18

Cluster	Proportion	Observations
1	0.637747	6167
2	0.015503	149
3	0.034409	337
4	0.007693	74
5	0.009611	92
6	0.001364	13
7	0.004503	43
8	0.001495	15
9	0.028554	281
10	0.007324	70
11	0.019200	177
12	0.009855	95
13	0.011413	119
14	0.055164	527
15	0.036981	368
16	0.005994	58
17	0.075302	740
18	0.037888	357
Min.	0.001364	13
Max.	0.637747	6167

### \*\* Information Criteria \*\*

Log likelihood: -254350 -254655.9 -172506.6

BIC: -254350

ICL: -254655.9

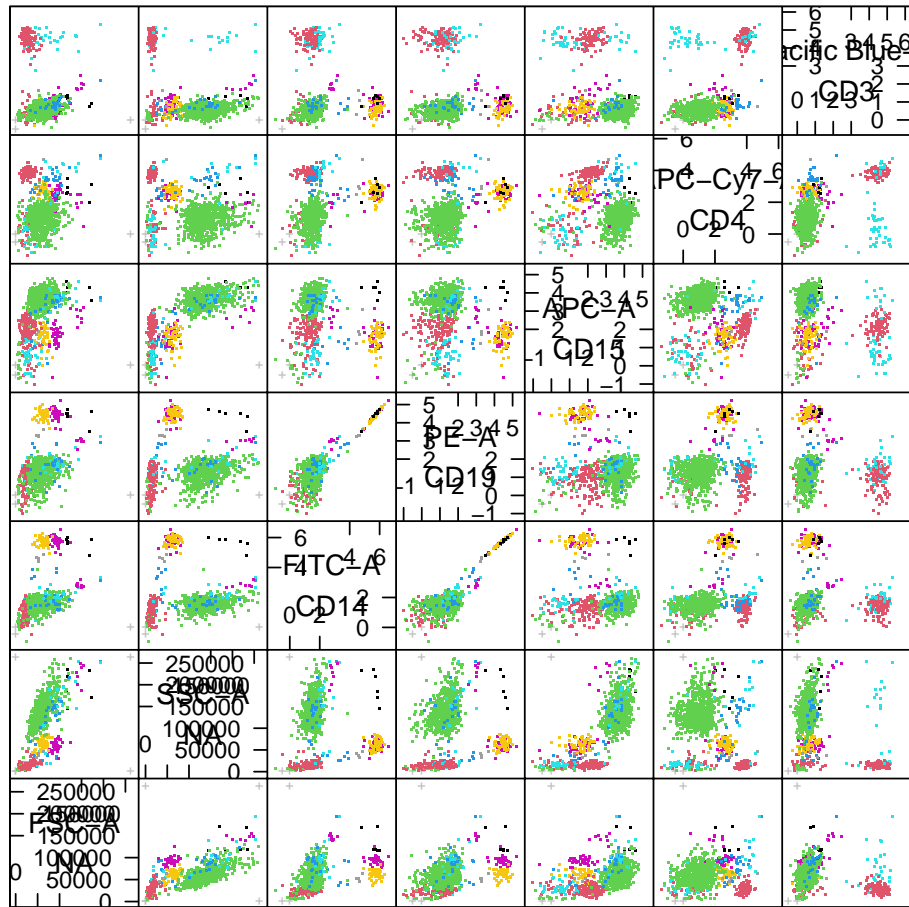
An ICL-bias of 0.3 is reasonable for fluorescence cytometry data based on our experiences, whereas the number of clusters increase dramatically when a `bias` below 0.2 is applied. A principal strategy for the ICL-bias in the whole pipeline is the use of a moderately small `bias` (0.2 - 0.3) for cell-clustering 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.

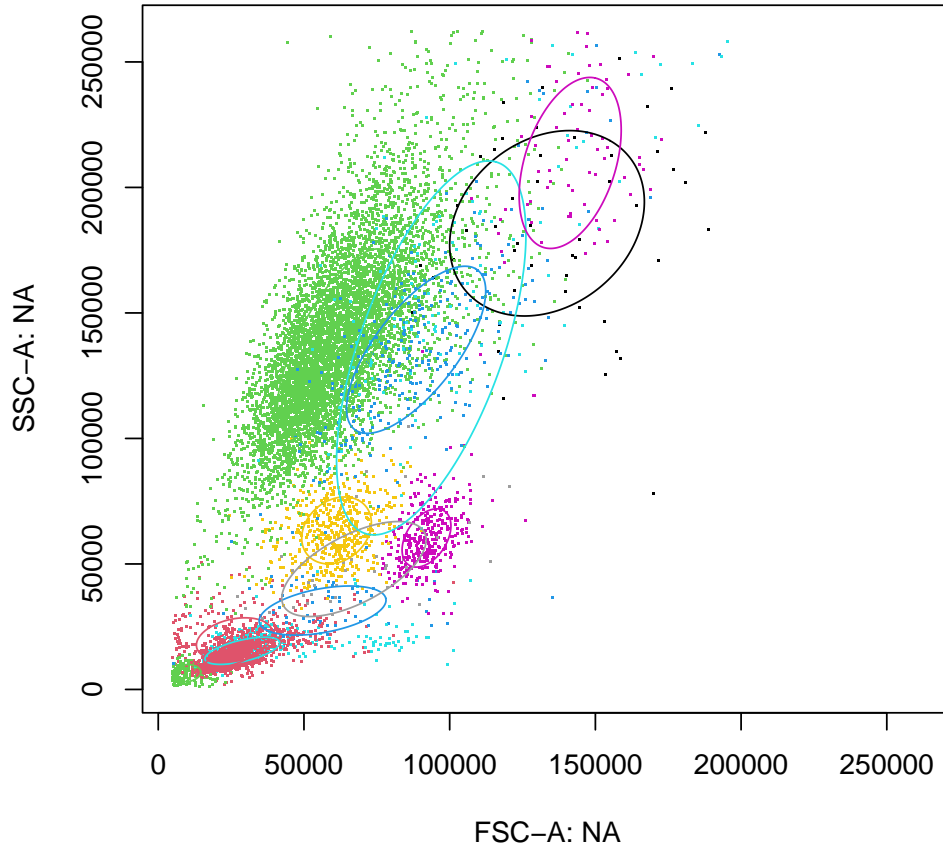
```
> splom(res.fcs, dat.transformed, N=1000)
```



Scatter Plot Matrix

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.

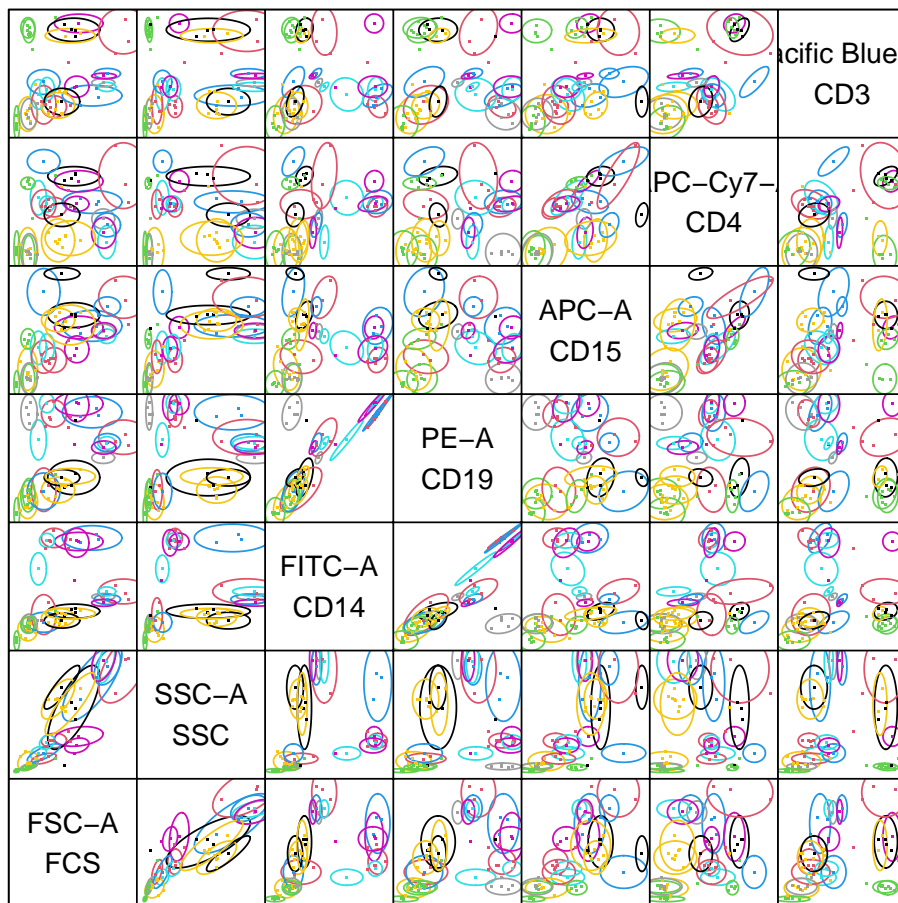
```
> data(dat.exp)
> meta<-meta.process(dat.exp, meta.bias=0.3)
```

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 manner using methods of the `immunoMeta`-object.

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

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

**.all**



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 on the event numbers of all meta-clusters in each sample

```
> cls <- clusters(meta, c())
> events(meta, cls)
```

	cls-1	cls-2	cls-3	cls-4	cls-5	cls-6	cls-7	cls-8	cls-9	cls-10	cls-11
exp-1	898	389	50	0	0	344	0	143	71	1107	0
exp-2	0	1079	0	173	102	695	926	8	145	3425	220
exp-3	0	574	0	0	0	780	452	199	0	1585	0
exp-4	761	433	62	0	0	527	331	0	0	0	0
exp-5	950	46	94	0	0	400	325	0	0	0	0
	cls-12	cls-13	cls-14	cls-15	cls-16	cls-17	cls-18	cls-19	cls-20	cls-21	



## immunoClust

```
exp-1    0    0  6459    70    0    0   151    0    0    0
exp-2  1447  923    0    0    24   103   495   77    0    0
exp-3    0    0  5717    0    0    10   247    0    0   132
exp-4    0    0  7280    0    0    0   247    0   95    0
exp-5    0    0  7417    0    0    0   278    0    0    0
  cls-22
exp-1    0
exp-2    0
exp-3   40
exp-4    0
exp-5    0
```

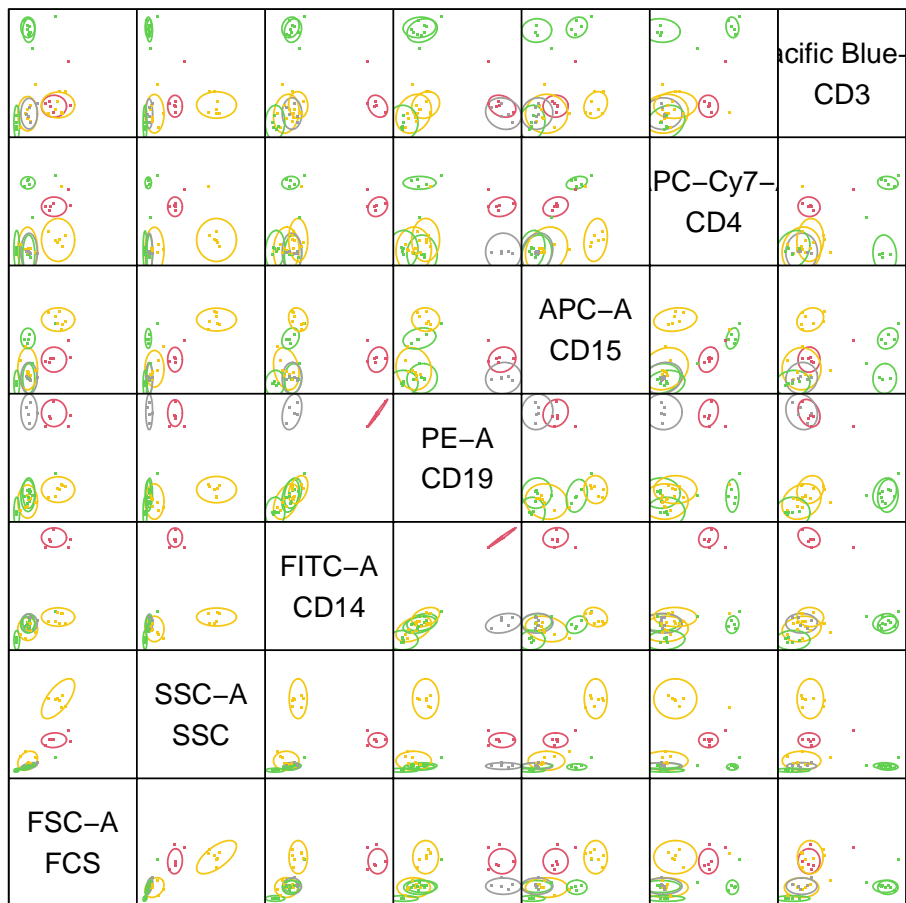
and pick the meta-clusters of the five commonly found population, with respect to the technical depletion to collect them in a first annotation level

```
> addLevel(meta,c(1),"leucocytes") <- c(1,2,6,7,10,14,18)
```

In the plot of this level the five major 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))
> sort(mu(meta,cls,7))      ## CD3 expression
      cls-18   cls-6   cls-7   cls-1   cls-14   cls-2   cls-10
0.5563285 1.0177510 1.0231479 1.4074683 1.4710931 5.3398778 5.5034995
> inc <- mu(meta,cls,7) > 5  ## CD3+ clusters
> cls[inc]
[1] 2 10
> mu(meta,cls[inc],6)      ## CD4 expression
      cls-2   cls-10
0.3526607 4.1704618
> addLevel(meta,c(1,1), "CD3+CD4+") <- 10
> addLevel(meta,c(1,2), "CD3+CD4-") <- 2
> cls <- unclassified(meta,c(1))
> sort(mu(meta,cls,5))      ## CD15 expression

```

## immunoClust

```

      cls-18   cls-7   cls-6   cls-1   cls-14
0.1607839 0.4098828 0.8552890 1.2885715 3.1912791

> inc <- mu(meta,cls,5) > 3
> addLevel(meta,c(1,3), "CD15+") <- cls[inc]
> cls <- unclassified(meta,c(1))
> sort(mu(meta,cls,3))      ## CD14 expression

      cls-18   cls-6   cls-7   cls-1
0.2970245 0.8748380 1.1685025 5.5770927

> inc <- mu(meta,cls,3) > 5
> addLevel(meta,c(1,4), "CD14+") <- cls[inc]
> cls <- unclassified(meta,c(1))
> sort(mu(meta,cls,4))      ## CD19 expression

      cls-18   cls-6   cls-7
0.2053237 0.6140560 3.9759928

> inc <- mu(meta,cls,4) > 3
> addLevel(meta,c(1,5), "CD19+") <- 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.all=FALSE)
> tbl[,1:5]

```

	12543	12546	12549	12552	12555
1.1.all_leucocytes_CD3+CD4+.10.green3	1107	3425	1585	0	0
1.2.all_leucocytes_CD3+CD4-.2.green3	389	1079	574	433	46
1.3.all_leucocytes_CD15+.14.yellow	6459	0	5717	7280	7417
1.4.all_leucocytes_CD14+.1.red	898	0	0	761	950
1.5.all_leucocytes_CD19+.7.gray	0	926	452	331	325
1.all_leucocytes.6.yellow	344	695	780	527	400
1.all_leucocytes.18.green3	151	495	247	247	278
.all.3.blue	50	0	0	62	94
.all.4.cyan	0	173	0	0	0
.all.5.magenta	0	102	0	0	0
.all.8.black	143	8	199	0	0
.all.9.red	71	145	0	0	0
.all.11.blue	0	220	0	0	0
.all.12.cyan	0	1447	0	0	0
.all.13.magenta	0	923	0	0	0
.all.15.gray	70	0	0	0	0
.all.16.black	0	24	0	0	0
.all.17.red	0	103	10	0	0
.all.19.blue	0	77	0	0	0
.all.20.cyan	0	0	0	95	0
.all.21.magenta	0	0	132	0	0
.all.22.yellow	0	0	40	0	0

## immunoClust

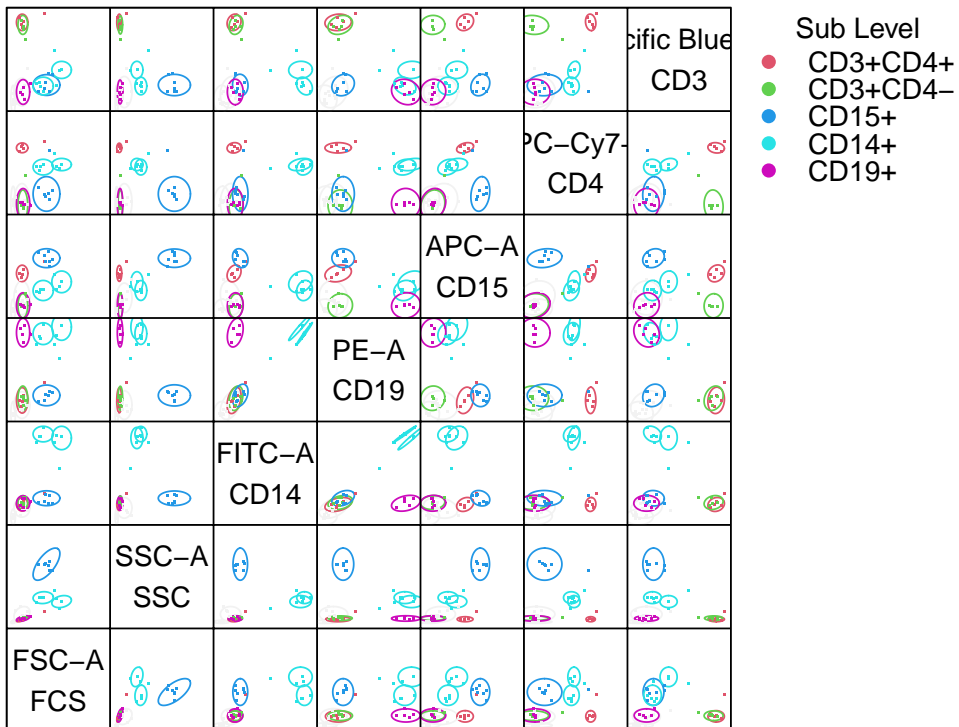
Each row denotes an annotated hierarchical level or/and meta-cluster and each column a data sample used in meta-clustering. The row names give the annotated population name, the meta-cluster index and the default color used in the plot routines for each meta-cluster. In the last columns additionally 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.

```
> move(meta, c(1,4)) <- 13
```

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

### 1.all\_leucocytes



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 4.2.0 RC (2022-04-19 r82224), x86\_64-pc-linux-gnu
- Locale: LC\_CTYPE=en\_US.UTF-8, LC\_NUMERIC=C, LC\_TIME=en\_GB, LC\_COLLATE=C, LC\_MONETARY=en\_US.UTF-8, LC\_MESSAGES=en\_US.UTF-8, LC\_PAPER=en\_US.UTF-8, LC\_NAME=C, LC\_ADDRESS=C, LC\_TELEPHONE=C, LC\_MEASUREMENT=en\_US.UTF-8, LC\_IDENTIFICATION=C
- Running under: Ubuntu 20.04.4 LTS
- Matrix products: default
- BLAS: /home/biocbuild/bbs-3.15-bioc/R/lib/libRblas.so
- LAPACK: /home/biocbuild/bbs-3.15-bioc/R/lib/libRlapack.so
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: flowCore 2.8.0, immunoClust 1.28.0
- Loaded via a namespace (and not attached): Biobase 2.56.0, BiocGenerics 0.42.0, BiocManager 1.30.17, BiocStyle 2.24.0, RProtoBufLib 2.8.0, Rcpp 1.0.8.3, RcppParallel 5.1.5, S4Vectors 0.34.0, cli 3.3.0, compiler 4.2.0, cytolib 2.8.0, digest 0.6.29, evaluate 0.15, fastmap 1.1.0, grid 4.2.0, htmltools 0.5.2, knitr 1.38, lattice 0.20-45, matrixStats 0.62.0, rlang 1.0.2, rmarkdown 2.14, stats4 4.2.0, tools 4.2.0, xfun 0.30, yaml 2.3.5