

immunoClust - Automated Pipeline for Population Detection in Flow Cytometry

*Till Sörensen*¹

¹till-
antoni.soerensen@charite.de

April 30, 2018

Contents

1	Licensing	2
2	Overview.	2
3	Getting started	2
4	Example Illustrating the immunoClust Pipeline	3
4.1	Cell Event Clustering	3
4.2	Meta Clustering	7
5	Session Info	11

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 assort 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 experimentally 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    0.00000  262143
$P5      SSC-A <NA> 262144 -111.00000  262143
$P8      FITC-A CD14 262144 -111.00000  262143
$P9      PE-A  CD19 262144 -111.00000  262143
$P12     APC-A CD15 262144 -111.00000  262143
$P13     APC-Cy7-A CD4 262144 -111.00000  262143
$P14 Pacific Blue-A CD3 262144 -98.93999  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: immunoClust Experiment
Data Filename:   fcs/12443.fcs
```

immunoClust

```
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: 12
Cluster      Proportion  Observations
1            0.091851      890
2            0.005914       57
3            0.040373     391
4            0.034162     333
5            0.015834     154
6            0.008024       79
7            0.084060     827
8            0.033546     312
9            0.637713    6166
10           0.028571     281
11           0.012627     122
12           0.007323       70

Min.         0.005914       57
Max.         0.637713    6166

** Information Criteria **
Log likelihood: -254005 -254201.8 -173041.3
BIC: -254005
ICL: -254201.8
```

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: immunoClust Experiment
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: 19

```

Cluster	Proportion	Observations
1	0.035909	360
2	0.034792	333
3	0.021334	198
4	0.006002	59
5	0.018863	176
6	0.011567	119
7	0.009854	95
8	0.030672	294
9	0.007740	77
10	0.014646	145
11	0.004426	43
12	0.090926	889
13	0.637967	6171
14	0.028596	281
15	0.009567	93
16	0.002077	19
17	0.001422	14
18	0.026315	246
19	0.007325	70
Min.	0.001422	14
Max.	0.637967	6171

```

** Information Criteria **
Log likelihood: -254496.1 -254857.9 -172543.8
BIC: -254496.1
ICL: -254857.9

```

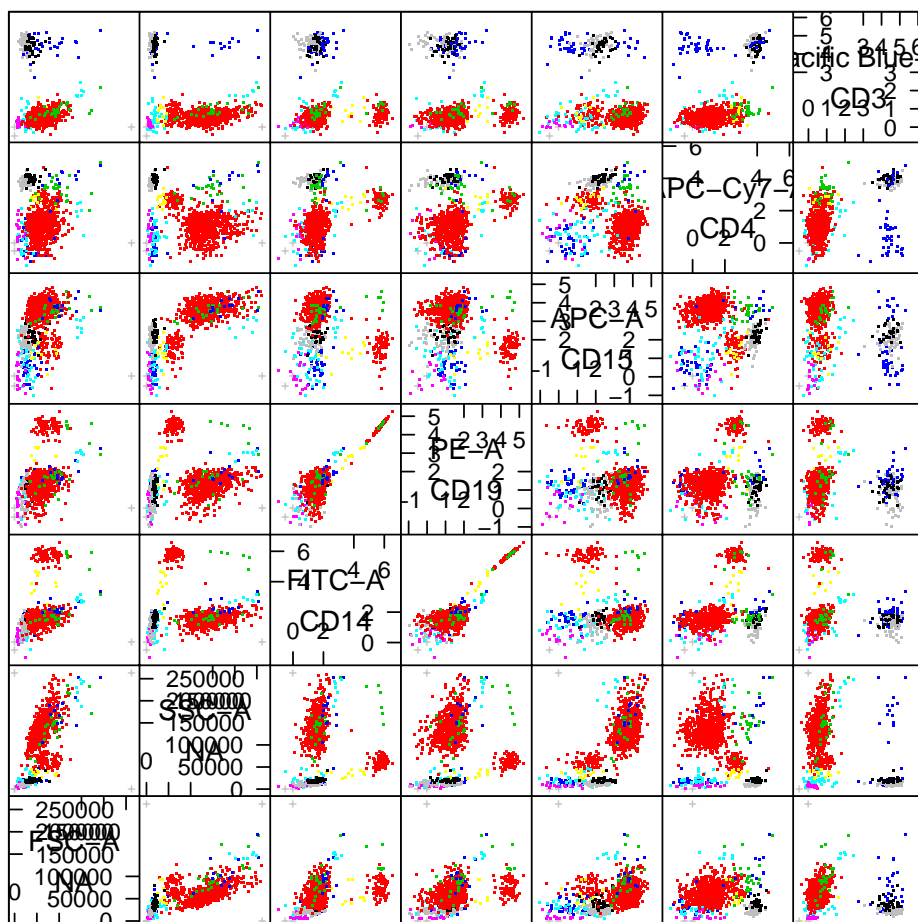
An ICL-bias of 0.3 is reasonable for fluorescence cytometry data based on our experiences, whereas the number of clusters increase dramatically when a `bias` below 0.2 is applied. A principal strategy for the ICL-bias in the whole pipeline is the use of a moderately small `bias` (0.2 - 0.3) for cell-clustering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

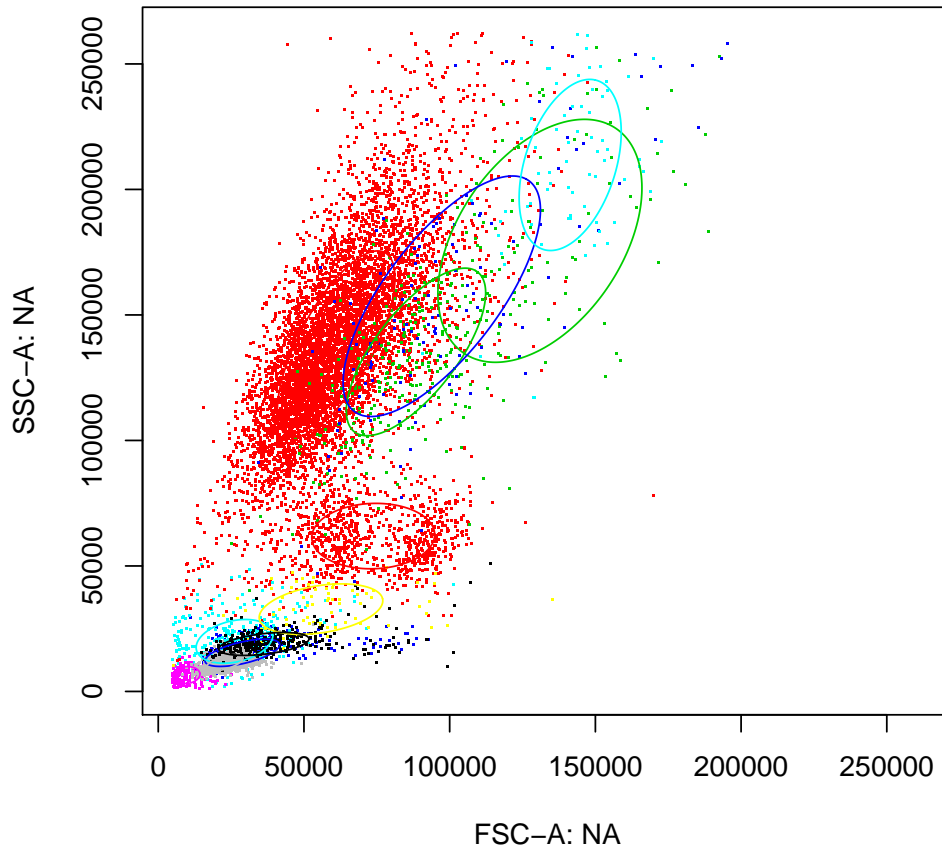
```
> splom(res.fcs, dat.transformed, N=1000)
```



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.

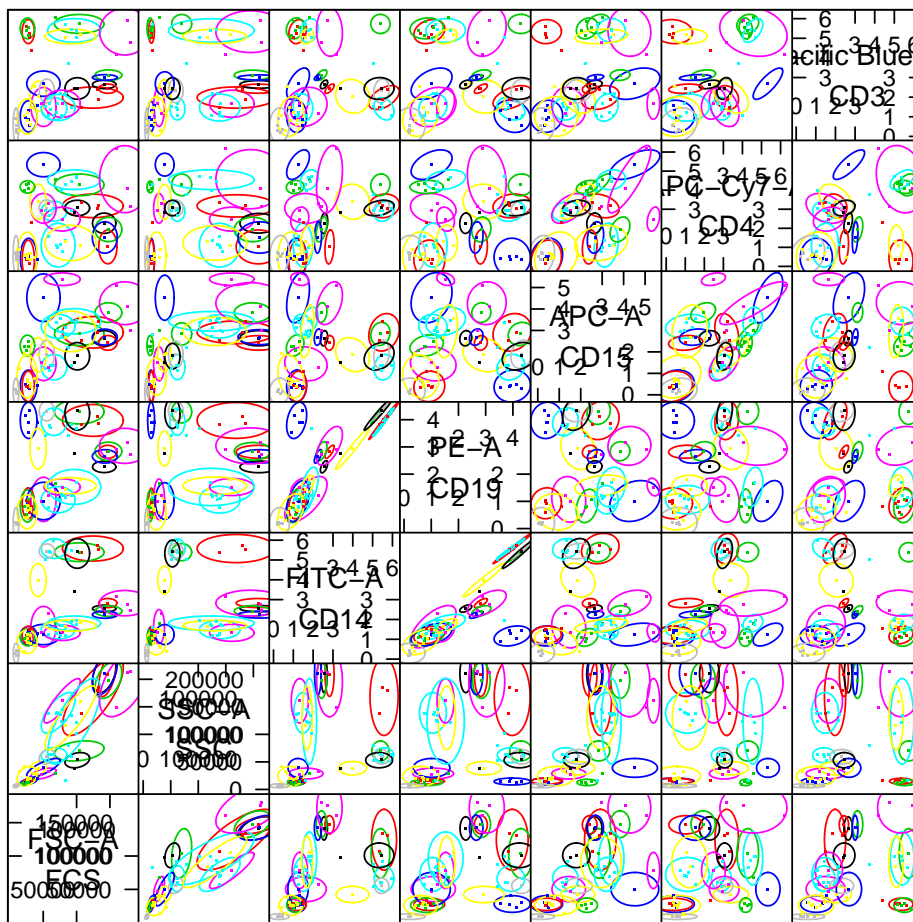
```
> data(dat.exp)
> meta<-meta.process(dat.exp, meta.bias=0.3, scatter.subset=c(1,2))
```

Scatter Gating

The obtained `list`-object contains the meta-clustering result in `$res.clusters`, and the used cell-clusters information in `$dat.clusters`. Additionally a meta-clustering using only the scatter parameter is performed within the `meta.process` function with results in `$res.scatter` and `$dat.scatter`. In a preliminary state of development an automated hierarchical gating on the meta-cluster is performed with results in `$gating`.

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

```
> splom(meta$res.clusters, meta$dat.clusters$M, ellipse=TRUE)
```

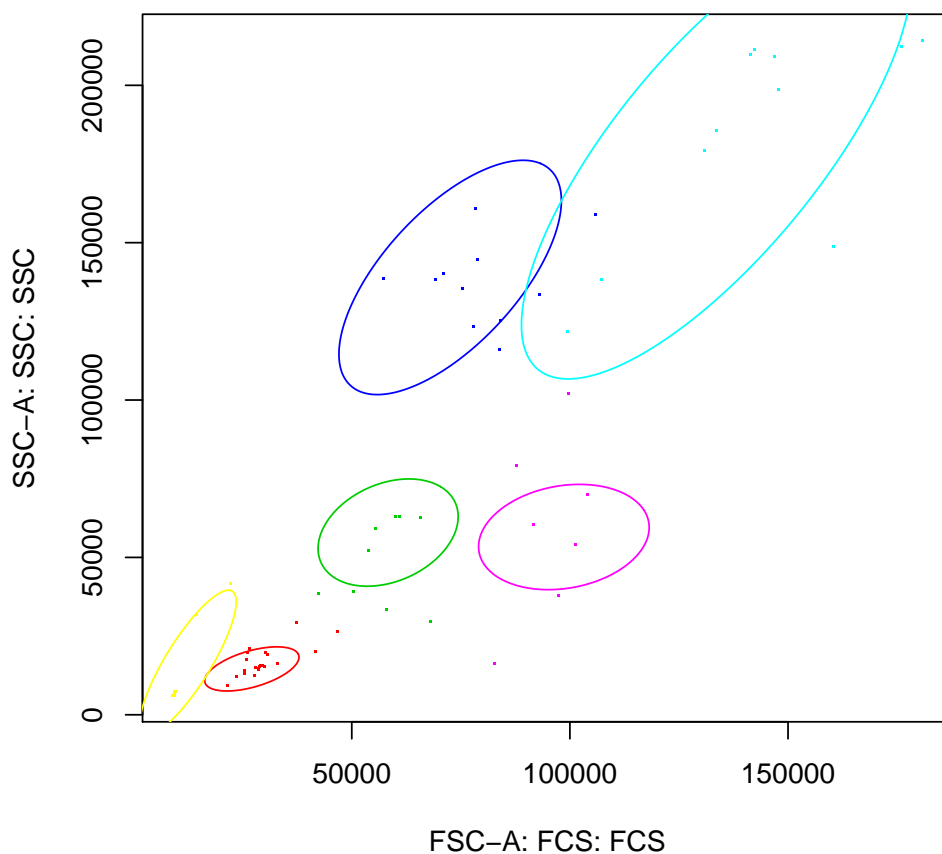


Scatter Plot Matrix

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

A scatter plot of the scatter parameter distribution of the cell-clusters is obtained by the `plot` method using `$res.scatter`.

```
> plot(meta$res.scatter, data=meta$dat.scatter$M)
```

The scatter distribution is divided into 6 major regions P1 - P6 with default colors red, green, blue, cyan, magenta and yellow.

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]
```

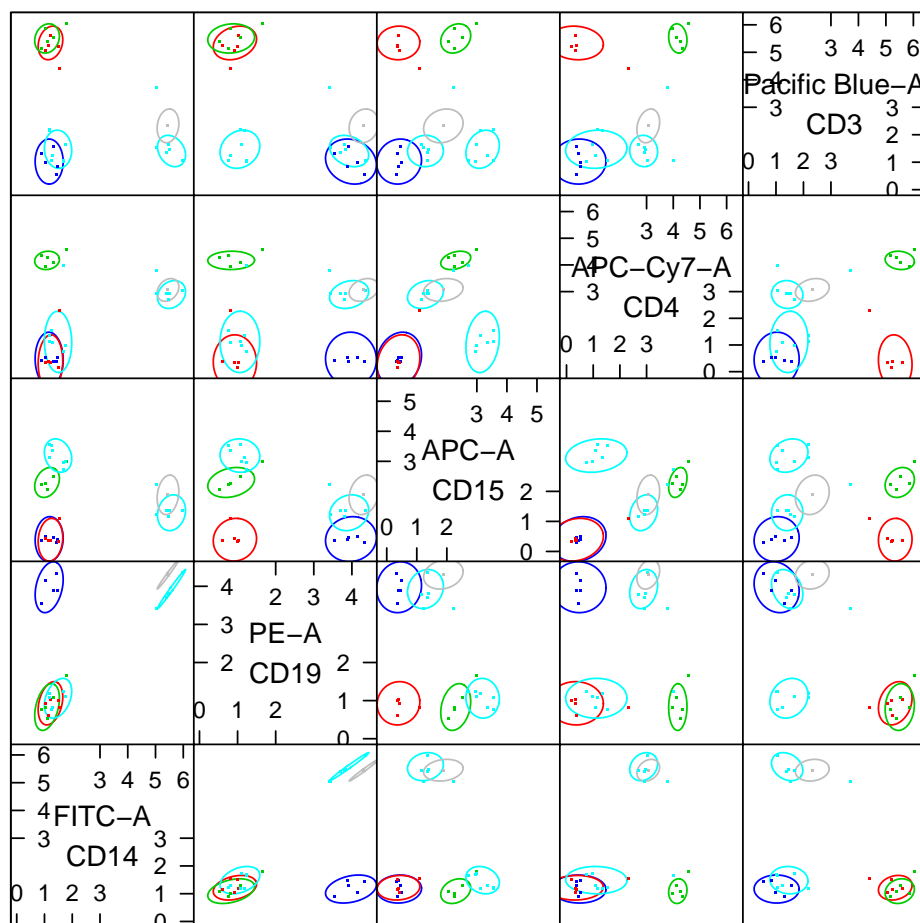
	12543	12546	12549	12552
1.1.1.1.P1_CD3neg_CD19neg_CD4neg.14.yellow	344	695	780	527
1.1.1.2.P1_CD3neg_CD19neg_CD4pos.5.magenta	71	145	0	0
1.1.2.P1_CD3neg_CD19pos.3.blue	0	926	452	331
1.2.1.P1_CD3pos_CD4neg.9.red	389	1079	574	433
1.2.2.P1_CD3pos_CD4pos.10.green3	1107	3425	1585	0
2.1.P2_CD14neg.11.blue	0	220	0	0
2.2.1.P2_CD14pos_CD3neg.4.cyan	898	0	0	761
2.2.2.P2_CD14pos_CD3pos.7.gray	0	1447	0	0
2.3.P2_CD14dim.6.yellow	0	173	0	0
3.1.1.P3_CD3neg_CD15neg.12.cyan	6459	0	5717	7280
3.1.2.P3_CD3neg_CD15pos.13.magenta	0	24	0	0

3.2.1.P3_CD3pos_CD4neg.22.yellow	0	0	40	0
3.2.2.P3_CD3pos_CD4pos.20.cyan	143	8	199	0
4.1.1.1.1.P4_CD14neg_CD3neg_CD15neg_CD19neg.16.black	70	0	0	0
4.1.1.1.2.P4_CD14neg_CD3neg_CD15neg_CD19pos.17.red	0	0	0	95
4.1.1.1.2.P4_CD14neg_CD3neg_CD15neg_CD19pos.19.blue	0	0	132	0
4.1.1.2.P4_CD14neg_CD3neg_CD15pos.18.green3	0	77	0	0
4.1.2.P4_CD14neg_CD3pos.21.magenta	0	103	10	0
4.1.2.1.P4_CD14neg_CD3pos.21.magenta_CD15pos.21.magenta	0	103	10	0
4.2.P4_CD14pos.1.red	50	0	0	62
5.1.P5_CD3neg.8.black	0	923	0	0
5.2.P5_CD3pos.2.green3	0	102	0	0
6.P6.15.gray	151	495	247	247
	12555			
1.1.1.1.P1_CD3neg_CD19neg_CD4neg.14.yellow	400			
1.1.1.2.P1_CD3neg_CD19neg_CD4pos.5.magenta	0			
1.1.2.P1_CD3neg_CD19pos.3.blue	325			
1.2.1.P1_CD3pos_CD4neg.9.red	46			
1.2.2.P1_CD3pos_CD4pos.10.green3	0			
2.1.P2_CD14neg.11.blue	0			
2.2.1.P2_CD14pos_CD3neg.4.cyan	950			
2.2.2.P2_CD14pos_CD3pos.7.gray	0			
2.3.P2_CD14dim.6.yellow	0			
3.1.1.P3_CD3neg_CD15neg.12.cyan	7417			
3.1.2.P3_CD3neg_CD15pos.13.magenta	0			
3.2.1.P3_CD3pos_CD4neg.22.yellow	0			
3.2.2.P3_CD3pos_CD4pos.20.cyan	0			
4.1.1.1.1.P4_CD14neg_CD3neg_CD15neg_CD19neg.16.black	0			
4.1.1.1.2.P4_CD14neg_CD3neg_CD15neg_CD19pos.17.red	0			
4.1.1.1.2.P4_CD14neg_CD3neg_CD15neg_CD19pos.19.blue	0			
4.1.1.2.P4_CD14neg_CD3neg_CD15pos.18.green3	0			
4.1.2.P4_CD14neg_CD3pos.21.magenta	0			
4.1.2.1.P4_CD14neg_CD3pos.21.magenta_CD15pos.21.magenta	0			
4.2.P4_CD14pos.1.red	94			
5.1.P5_CD3neg.8.black	0			
5.2.P5_CD3pos.2.green3	0			
6.P6.15.gray	278			

Each row denotes a meta-cluster and each column a data sample used for meta-clustering. The row names give the automated generated gating name, the meta-cluster index and the default color used in the plot routines for each meta-cluster. With an argument `out.all=TRUE` additionally the event numbers in each gating hierarchy level are extracted. 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.

Picking the meta-clusters of the five commonly found population, with respect to the technical depletion the scatter plot matrix reduces to

```
> splom(meta$res.clusters, meta$dat.clusters$M,
+ include=c(3,9,10,4,7,12), subset=3:7, ellipse=TRUE)
```



Scatter Plot Matrix

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 variation of the CD3 expression in the CD14-population of sample 12546 is caused artificially due to depletion of the granulocytes, which constitute about 60% - 75% of the cells in the other samples.

5 Session Info

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

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

- R version 3.5.0 (2018-04-23), x86_64-w64-mingw32
- Locale: LC_COLLATE=C, LC_CTYPE=English_United States.1252, LC_MONETARY=English_United States.1252, LC_NUMERIC=C, LC_TIME=English_United States.1252
- Running under: Windows Server 2012 R2 x64 (build 9600)
- Matrix products: default

immunoClust

- Base packages: base, datasets, grDevices, graphics, grid, methods, stats, utils
- Other packages: flowCore 1.46.0, immunoClust 1.12.0, lattice 0.20-35
- Loaded via a namespace (and not attached): Biobase 2.40.0, BiocGenerics 0.26.0, BiocStyle 2.8.0, DEoptimR 1.0-8, Rcpp 0.12.16, backports 1.1.2, cluster 2.0.7-1, compiler 3.5.0, corpcor 1.6.9, digest 0.6.15, evaluate 0.10.1, graph 1.58.0, htmltools 0.3.6, knitr 1.20, magrittr 1.5, matrixStats 0.53.1, mvtnorm 1.0-7, parallel 3.5.0, pcaPP 1.9-73, rmarkdown 1.9, robustbase 0.93-0, rprojroot 1.3-2, rrcov 1.4-3, stats4 3.5.0, stringi 1.1.7, stringr 1.3.0, tools 3.5.0, yaml 2.1.18