flowBeads: Bead Normalisation in Flow Cytometry

March 13, 2013

Abstract

The flowBeads package is an extension of flowCore for bead data. It provides basic functionality for loading, gating and doing normalisation with bead data. Beads specially manufactured to known fluoresence, defined in terms of standard units of fluorescence, are routinely run in flow cytometry for the purpose of instrument quality control and normalisation. The transformation of measured intensity (Mean Fluorescence Intensity) to standard units of fluorescence, Molecules of Equivalent Fluorochrome, allows for sensible comparison of data acquired on different days and on different instruments. The parameters of the transform also correspond to basic quality control estimates of the detector linearity and the background.

1 Theory

The expected fluorescent signal of a bead is determined by:

- the amount of fluorochrome carried by the bead
- the properties of the excitation source (wavelength of laser)
- the properties of the detector channel (bandpass and voltage)

Given these properties, the **Molecules of Equivalent Fluorochrome (MEF)** or Molecules of Equivalent Soluble Fluorochrome (MESF) standard unit of fluorescence is calculated by the manufacturer. This theoretical value provides an absolute scale for measuring fluorescence to compare samples analysed at different times or under different laser/detector configurations (Schwartz et al., 1996; Dendrou et al., 2009). The transform from relative fluorescence to standard fluorescence is a linear transform which is estimated by linear regression of the **Mean Fluorescence Intensity (MFI)** of beads belonging to a number (usually six) of different populations of increasing brightness against their expected MEF fluorescence. Since the MEF of the bead populations scales multiplicatively, a chosen transform `f` is appropriate to linearise the data. In the case of FCS2 data, `f` is $\log_{10}$, and in the case of FCS3, the default choice is the `logicleTransform` of the flowCore package. On the `f` linearised fluorescence scale the transform is therefore:

$$\mathrm{f(MEF)} = \beta \times \mathrm{f(MFI)} + \alpha$$

$$\mathrm{MEF} = \mathrm{f}^{-1}(\beta \times \mathrm{f(MFI)} + \alpha)$$

In the special case where the transform `f` is $\log_{10}$, this can be further simplified to:

$$\mathrm{MEF} = 10^{\alpha} \times \mathrm{MFI}^{\beta}$$

Provided the linearity of the detector is good, the β parameter, representing the slope, is generally close to one. When the beads are run on different days, the MFIs of the bead populations move little relative to each other but

instead shift together as a whole, thus the intercept α, which can be interpreted as the background fluorescence, varies more than the the slope β.

In order to apply the MEF transform, the MEF of the beads for a given laser/detector setup, as supplied by the manufacturer, needs to be matched to the laser/detector configuration provided in the FCS bead file. However, since not all required laser/detector properties are stored as part of the FCS 2 or 3 file format, we rely instead on the names of the detectors channels in the FCS file matching those in the MEF configuration file. As part of the flowBeads package, there is support for **Dakocytomation FluoroSpheres** beads (see Table 1) and **ThermoFischer Scientific Cytocal** for the standard **LSRII** and **LSRFortessa** laser/detector setup, but any other type of bead can be supported provided the MEF configuration file is specified. And to load the Dakocytomation configuration file into the current workspace:

```
> data(dakomef)
```

	FITC	PE	PE.CY5	APC	PE.TEXAS.RED
1					
2	2500	1500	750	4100	552
3	6500	4400	2100	10300	2014
4	19000	14500	6900	25500	6975
5	55000	43800	22100	67300	20685
6	150000	131200	77100	139100	71888

Table 1: FluoroSpheres from Dakocytomation. The Molecules of Equivalent Fluorochromes (MEF) values for the six bead populations as provided by the manufacturer for the LSRII. The first bead population are blank as they contain contain no flurochrome by design.

To load the Cytocal configuration file into the current workspace:

```
> data(cytocalmef)
```

Note that the underlying assumption in using beads as a reference is that the physical MEF property of these beads is more stable than the detected MFI of the bead population as reported by the instrument. For this to be true, the quality of the beads must not be compromised by age or poor storage. Also it is important to keep in that in mind that if any properties of the laser/detector change, for example the voltage of the detector, then the beads need to be run again to recompute the correct transform.

2 Loading Bead Files

Two example FCS 3 bead files, Dakocytomation beads ran on two different days, are included as part of the flowBeads package. These files may be loaded like so:

```
> beads1 <- BeadFlowFrame(fcs.filename=system.file('extdata', 'beads1.fcs', package='flowBeads'))
> beads2 <- BeadFlowFrame(fcs.filename=system.file('extdata', 'beads2.fcs', package='flowBeads'))
```

As no MEF configuration file has been specified as an argument to `BeadFlowFrame`, they are assumed to be the default Dakocytomation beads. `beads1` and `beads2` are also saved as R objects as part of the flowBeads package and so can be loaded directly:

```
> data(beads1)
> data(beads2)
```

Here are a few ways of extracting information from *BeadFlowFrame* objects:

```
> print(beads1)

BeadFlowFrame object '9de89faa-c6fe-4d82-ad9b-ae64a02c3122'
from 2008-01-21
with 5144 beads and 8 observables:
            name desc  range minRange maxRange
$P1          FSC <NA> 262144     0.00   262143
$P2          SSC <NA> 262144     0.00   262143
$P3    ALEXA.488 <NA> 262144   -82.08   262143
$P4       PE.CY7 <NA> 262144   -87.78   262143
$P5          APC <NA> 262144   -26.60   262143
$P6           PE <NA> 262144  -111.00   262143
$P7    ALEXA.700 <NA> 262144   -77.90   262143
$P8 PACIFIC.BLUE <NA> 262144     0.00   262143
146 keywords are stored in the 'description' slot
Beads MEF
    FITC    RPE RPE.CY5    APC PE.TEXAS.RED
1   2500   1500     750   4100          552
2   6500   4400    2100  10300         2014
3  19000  14500    6900  25500         6975
4  55000  43800   22100  67300        20685
5 150000 131200   77100 139100        71888

> print(length(beads1))

[1] 5144

> print(getDate(beads1))

[1] "2008-01-21"

> print(getParams(beads1))

[1] "ALEXA.488"    "PE.CY7"       "APC"          "PE"           "ALEXA.700"
[6] "PACIFIC.BLUE"

> print(getMEFparams(beads1))

[1] "FITC"         "RPE"          "RPE.CY5"      "APC"          "PE.TEXAS.RED"
```

Once the bead files are loaded we can then gate them to identify the distinct populations and compute the MEF transform.

3 Gating Bead Data

Gating of bead data is straightforward as the number of bead populations is known a priori. In the forward and side scatter detector channels, a single population is expected since all beads are known to be of identical shape

and size. Using the `norm2Filter` from flowCore (Ellis et al.), we fit a bivariate Gaussian to the data. Events lying more than one standard deviation away from the mean of the main bead population are excluded.

Once we have gated on the main bead population in the scatter channels, we know from the MEF file that the beads belong to six populations of increasing brightness. All channels are gated with the number of expected clusters set to the number of bead population reported in the bead type file (which is six in the case of Dakocytomation beads). The gating is done separately on each fluorescent channel using the `pam` function in the cluster (Maechler et al., 2012) which is an implementation of the K-medoids algorithm:

```
> gbeads1 <- gateBeads(beads1, verbose=T)
> gbeads2 <- gateBeads(beads2, verbose=F)
```

Note `gateBeads` is quite a slow function as the `pam` function is quite computationally intensive. `gbeads1` and `gbeads2` are *GatedBeadFlowFrame* objects which contain the results of the gating. They can also be loaded directly as saved R objects:

```
> data(gbeads1)
> data(gbeads2)
```

To visualise the results of the gating (see Figure 1 for `gbeads1` and Figure 2 for `gbeads2`):

```
> plot(gbeads1)
```

Individual channels can be plotted like so:

```
> plot(gbeads1, 'APC')
```

Clustering statistics are also calculated and stored in the `clustering.stats` slot as a three way array indexed by statistic (count, mean, standard deviation, coefficient of variation), channel (`ALEXA.488, PEC.Y7, APC, PE, ALEXA.700` and `PACIFIC.BLUE`) and bead population (one to six). For example, the clustering stats of bead population one (the blank beads):

```
> gbeads1@clustering.stats[,,1]
```

```
        ALEXA.488     PE.CY7       APC        PE ALEXA.700 PACIFIC.BLUE
count   156.00000  359.00000 156.00000 156.00000 156.00000    156.00000
mean.fi  14.30846  -11.04752 116.98397  11.06385  34.69327    171.85102
sd.fi    33.59705   23.40036  58.69615  44.54962  60.68674     35.99458
cv      234.80550 -211.81550  50.17452 402.65946 174.92368     20.94522
```

The *GatedBeadFlowFrame* defines `mef.transform` slot which contains a list indexed by channel name, where each element is a list containing the transformation function to apply as well as the coefficients of the transform. As we do not have MEF values for all detector channels, we only define an MEF transform for ones with matching names in the bead configuration file (in this case APC). See Figure 3 for absolute normalisation of the APC channel.

```
> mef.transform <- gbeads1@mef.transform
> names(mef.transform)
```

```
[1] "APC"
```

```
> names(mef.transform$APC)
```

```
[1] "alpha" "beta"  "m"     "rse"   "fun"
```

The `toMEF` function takes a *GatedBeadFlowFrame* and a *flowFrame* and normalises the channels for which we have an MEF transform:

```
> toMEF(bead.data, flow.data)
```

4 Relative Normalisation

The MEF provides an absolute reference but we can still normalise in the absence of MEF provided we can align the MFIs across days. An advantage of relative normalisation is that we can also align the blank bead population as we do not need the MEF. Let MFI_1 be the MFI obtained from the beads on day one, and MFI_2 be the MFI obtained from the beads on day two, then the relative normalisation to compare samples from day one to day two is:

$$\mathrm{f}(\mathrm{MFI}_2) = \beta \times \mathrm{f}(\mathrm{MFI}_1) + \alpha$$

$$\mathrm{MFI}_2 = \mathrm{f}^{-1}(\beta \times \mathrm{f}(\mathrm{MFI}_1) + \alpha)$$

To compute the transform:

```
> relative.transforms <- relativeNormalise(gbeads1, gbeads2)
> names(relative.transforms)
```

```
[1] "ALEXA.488"    "PE.CY7"       "APC"          "PE"           "ALEXA.700"
[6] "PACIFIC.BLUE"
```

We can then apply the transform, see Figure 4 for result of applying relative normalisation to bead data.

```
> fun <- relative.transforms$APC$fun
> mfi1 <- gbeads1@trans(gbeads1@clustering.stats['mean.fi','APC',])
> mfi2 <- gbeads2@trans(gbeads2@clustering.stats['mean.fi','APC',])
> fun.mfi1 <- fun(mfi1)
```

5 Generating a Report

Once the bead data has been gated it is possible to generate an HTML report from a template written using Markdown. These reports can then be viewed as web pages and linked to from a summary page which shows timeline data. This function is not strictly necessary as one may easily implement his own template.

```
> generateReport(gbeads1, output.file='report.html')
```

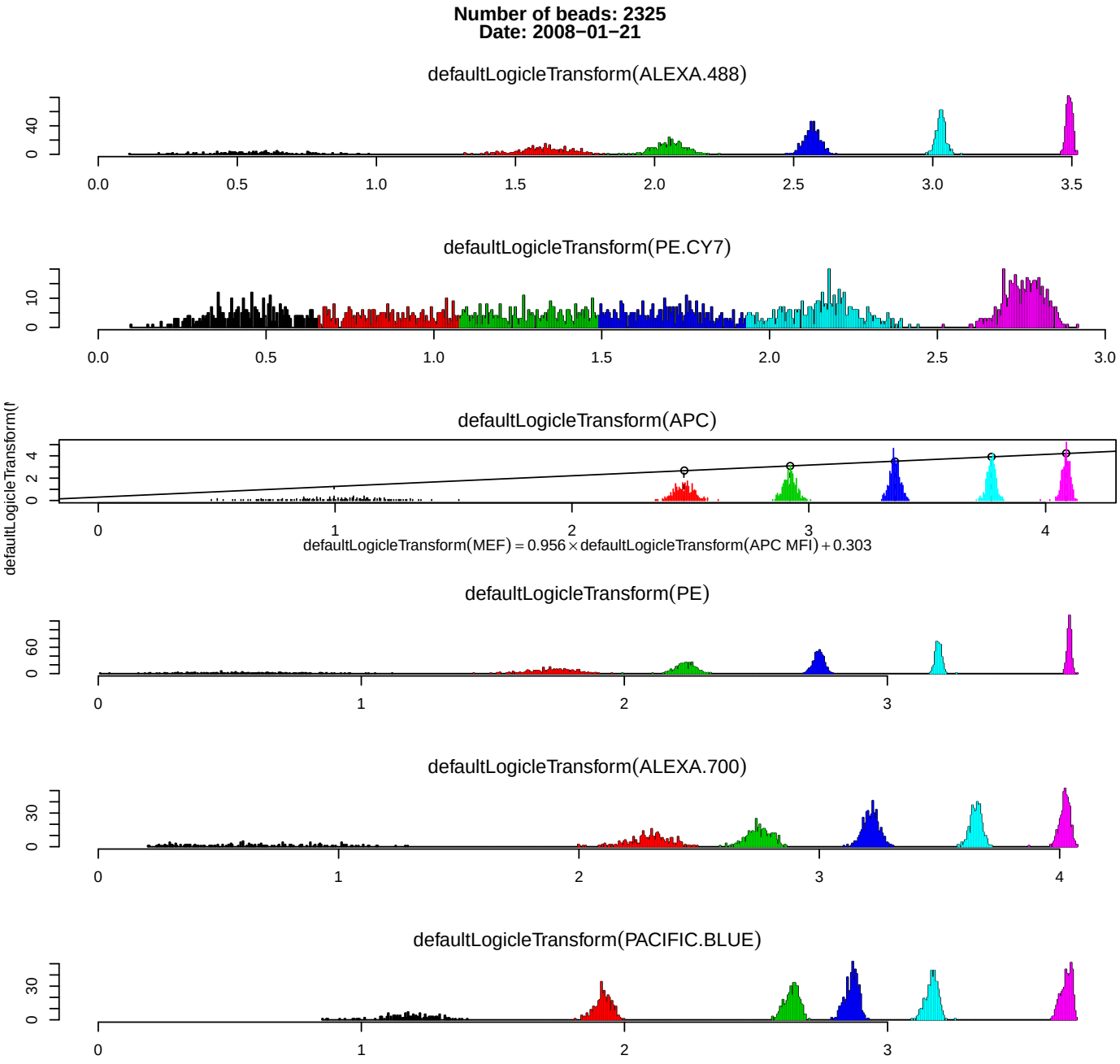

Figure 1: Plot of `gbeads1`. APC gated beads day 1. We see tight clusters. The channel for `PE.CY7` is not tuned to pick up the signal hence in the noise.

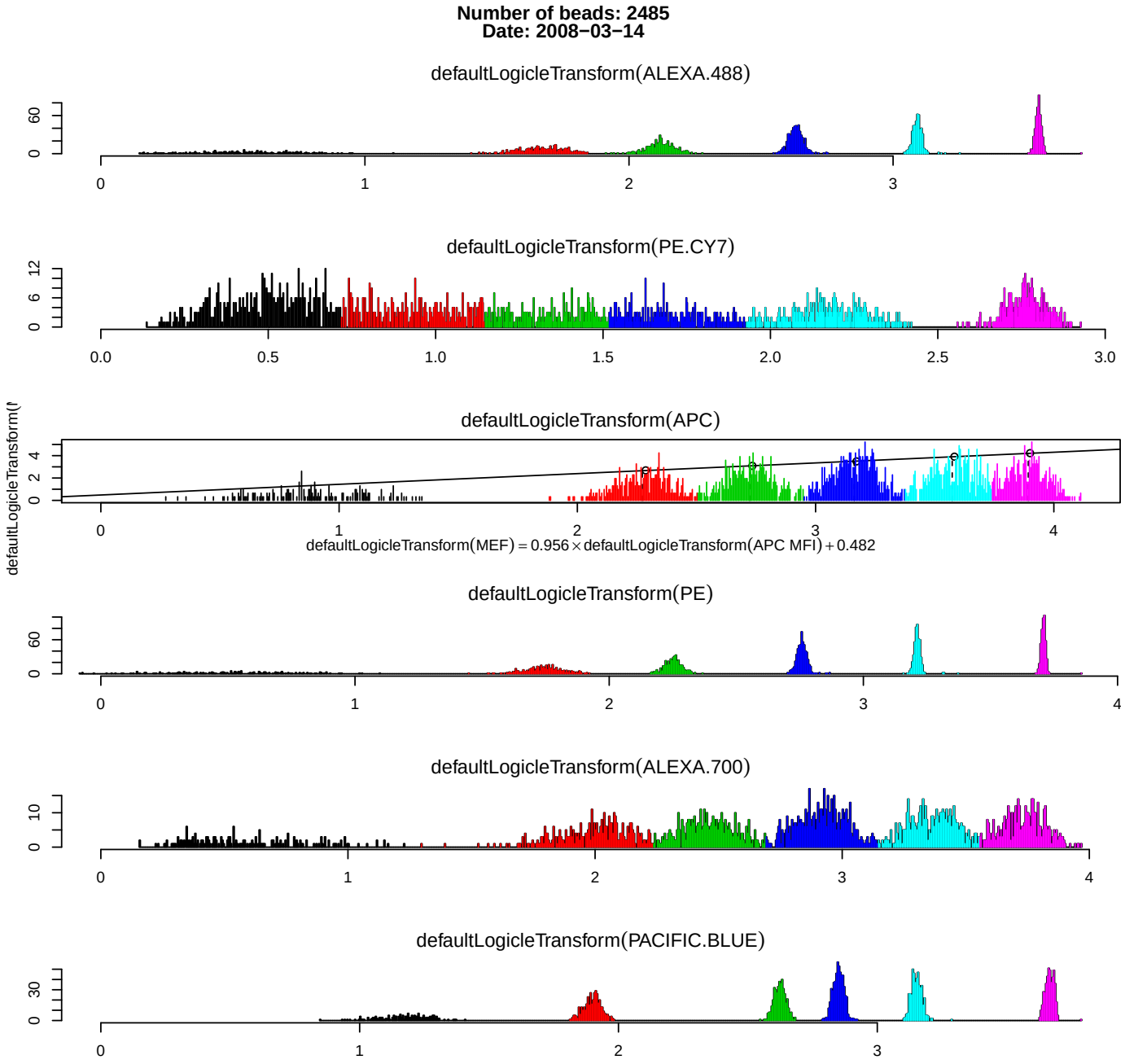

Figure 2: Plot of `gbeads2`. APC gated beads day 2 the α is higher than on day 1 which implies higher background. The clusters are also far more noisy on this day than on the previous day. The channel for `PE.CY7` is not tuned to pick up the signal hence in the noise.

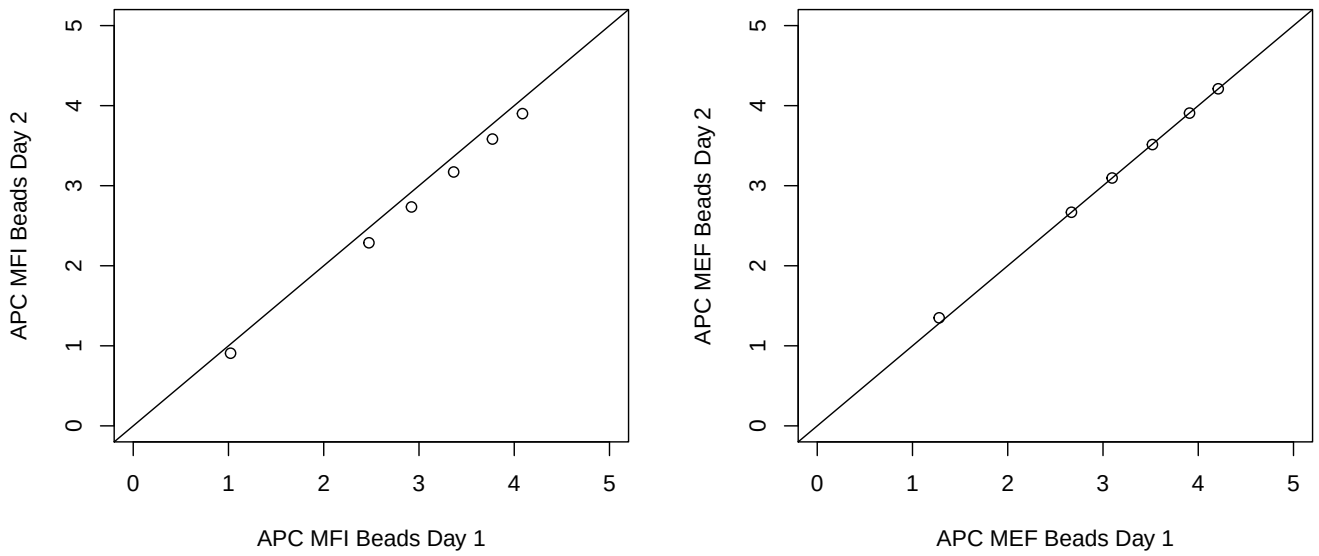

Figure 3: The result of the MEF transform is to align the MFI of the five (non-blank) bead populations across days. Notice that that the alignment of the blank bead population is not perfect since it is not used in estimating the normalisation parameters.

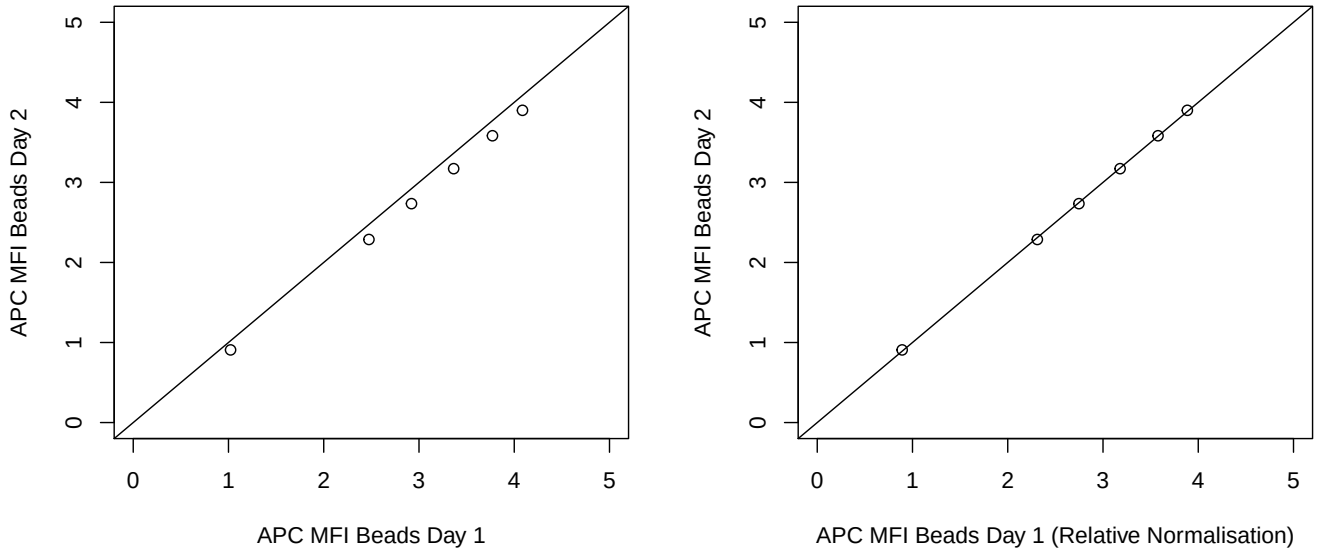

Figure 4: The result of the relative MFI transform is to align the MFI of the six bead populations across both days. Note that after relative normalisation, the MFIs from all bead populations are perfectly aligned.

References

Calliope A Dendrou, Erik Fung, Laura Esposito, et al. Fluorescence Intensity Normalisation: Correcting for Time Effects in Large-Scale Flow Cytometric Analysis. *Advances in Bioinformatics*, 2009:1–6, 2009.

B. Ellis, P. Haaland, F. Hahne, N. Le Meur, and N. Gopalakrishnan. *flowCore: flowCore: Basic structures for flow cytometry data*. R package version 1.20.0.

Martin Maechler, Peter Rousseeuw, Anja Struyf, Mia Hubert, and Kurt Hornik. *cluster: Cluster Analysis Basics and Extensions*, 2012. R package version 1.14.2 — For new features, see the 'Changelog' file (in the package source).

A Schwartz, EF Repollet, R Vogt, and JW Gratama. Standardizing flow cytometry: Construction of a standardized fluorescence calibration plot using matching spectral calibrators. *Cytometry Part A*, 26(1):22–31, 1996.