curvHDR: filtering of flow cytometry data via significant curvature and highest density regions

G. Luta, U. Naumann and M.P. Wand

20th April, 2010

Flow cytometry is a laser-based biotechnology that produces large multivariate samples. Typically, each member of the sample corresponds to the physical properties of a biological cell – known as *forward scatter* and *side scatter* – and antibody binding activity, through fluorescence intensity measurements. The latter measurements arise from the cells being exposed to several fluorescently conjugated antibodies during the flow cytometry procedure. Shapiro (2003) is a comprehensive reference on flow cytometry.

Filtering, also known as *gating*, is an integral component of flow cytometric data analysis is where cells are subsetted according to physical and fluorescence measurements. The curvHDR method aims to mimic human perception of what subsets might be of interest, using the notions of significant curvature and highest density regions (HDR). Full details of curvHDR are given in Naumann, Luta & Wand (2010). curvHDR filters may be combined with others (e.g. those corresponding to rectangular constraints) to aid automatic processing of flow cytometry data. Naumann & Wand (2009) use such a strategy for a large longitudinal flow cytometry data-set.

The main parameter of curvHDR is the *HDR level* parameter, which we denote here by τ. This parameter controls the probability mass contained in curvHDR sub-regions. For a d-variate density function f and $\tau \in [0,1]$ the τ HDR is

$$R_\tau \equiv \{\boldsymbol{x} \in \mathbb{R}^d : f(\boldsymbol{x}) \geq f_\tau\} \text{ where } f_\tau \text{ is the greatest number for which } \int_{R_\tau} f(\boldsymbol{x})\,d\boldsymbol{x} \geq 1-\tau$$

(e.g. Hyndman, 1996). We can think of the R_τ as corresponding ‘meaningful’ contours of the density function f. For example, $R_{0.9}$ is the region inside that contour of f for which the probability is 0.1, a relatively small region near the peak of f. The HDR $R_{0.1}$ encompasses to 90% of the probability mass of f. Since, in practice, where f is unknown curvHDR works with estimated HDRs, in which f is replaced by a kernel density estimate.

In this vignette we illustrate curvHDR filtering in R. The central function is `curvHDRfilter()`. It produces an object of class `curvHDRfilter`, which can be visualised using an S3 method `plot()` function.

Example Flow Cytometry Data

Figure 1 shows some example longitudinal flow cytometry data (source: Brinkman *et al.*, 2007). These data are available in the `Bioconductor` package `flowViz`.

Figure 1 was obtained via the following R commands. `GvHDtrans` is a *flowSet* object and, in the following sections, is used for illustration of curvHDR.

```
> library(flowViz)
> data(GvHD)
> GvHDtrans <- transform("FSC-H" = asinh, "SSC-H" = asinh, "FL1-H" = asinh,
+     "FL2-H" = asinh, "FL3-H" = asinh, "FL2-A" = asinh, "FL4-H" = asinh) %on%
+     GvHD
> GvHDtransViz <- xyplot(`SSC-H` ~ `FL2-H` | factor(Days), GvHDtrans,
+     subset = Patient == "9", ylim = c(3.5, 7.7))
> print(GvHDtransViz)
```

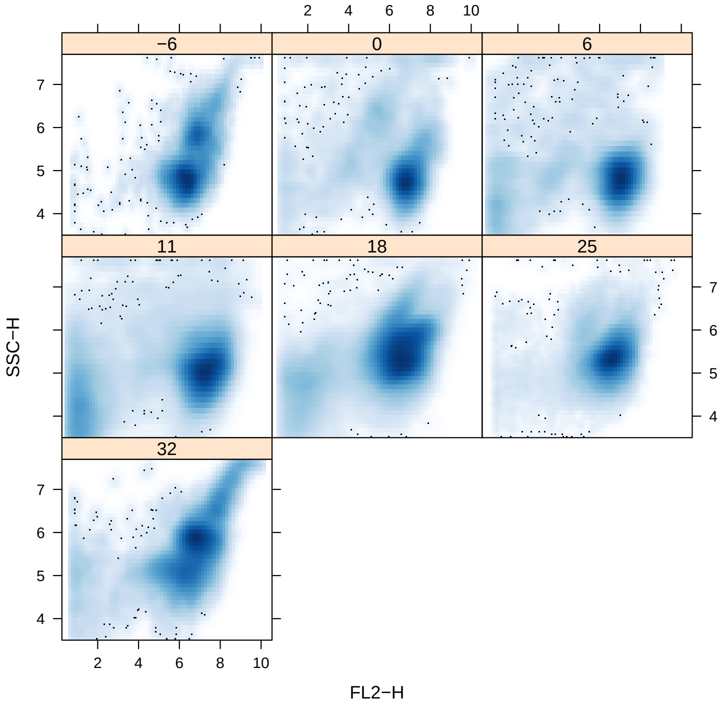

Figure 1: *Some example longitudinal flow cytometry data corresponding to a study on graft-versus-host disease (source: Brinkman et al., 2007). The panels correspond to day number with respect to blood and marrow transplant of a particular patient. The vertical axis is side-scatter, whilst the horizontal axis is the second fluorescence channel. Since the data are large* `flowViz` *defaults to displaying the data as smoothed scatterplots, based on bivariate density estimation.*

Bivariate curvHDR

Figure 2 shows the `HDRlevel=0.2` curvHDR filter for the data shown in the upper-left panel of Figure 1 (corresponding to 6 days before transplant).

Figure 2 was obtained using the following set of commands.

```
> inputData <- exprs(GvHDtrans$s9a01)[, c(4, 2)]
> library(curvHDR)
> cHfObj1 <- curvHDRfilter(inputData, HDRlevel = 0.2)
> plot(cHfObj1, xlab = "FL2-H", ylab = "SSC-H", xlim = c(5, 8.5),
+     ylim = c(4, 7))
```

Note the specification `HDRlevel = 0.2` used in the call to `curvHDRfilter()`. The resulting object, `cHfObj1`, is of class `curvHDRfilter` and is recognised by `plot()`.

Figure 3 shows the result of dropping the `HDRlevel` parameter to 0.1. This results in larger filters since each sub-region now corresponds to about 90% of the data within that region. The commands that led to Figure 3 are as follows:

```
> cHfObj2 <- curvHDRfilter(inputData, HDRlevel = 0.1)
> plot(cHfObj2, xlab = "FL2-H", ylab = "SSC-H", xlim = c(5, 8.5),
+     ylim = c(4, 7))
```

Univariate curvHDR

The `curvHDRfilter()` function may also be applied to univariate flow cytometry data as illustrated by the following code:

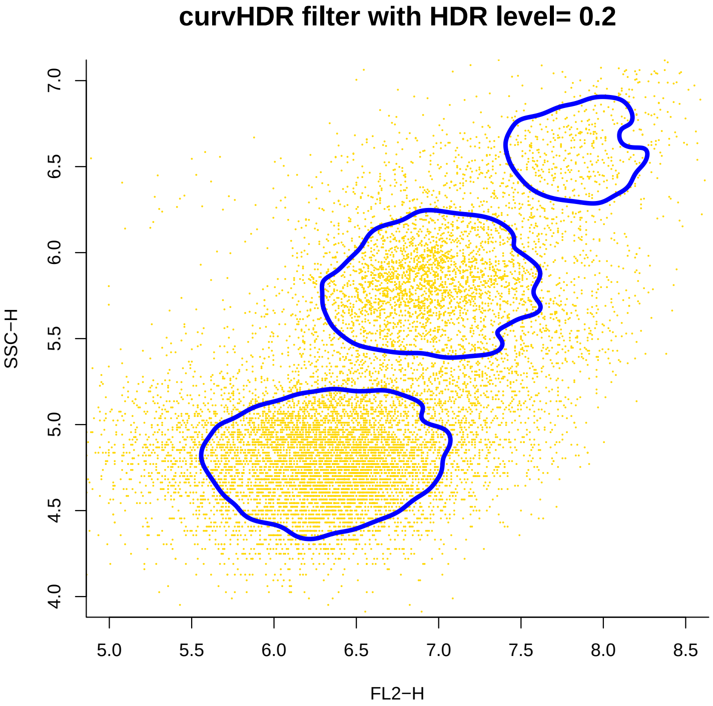

Figure 2: curvHDR *filter of data in the upper-left panel of Figure 1 (corresponding to 6 days before transplant). The HDR level parameter is equal to 0.2.*

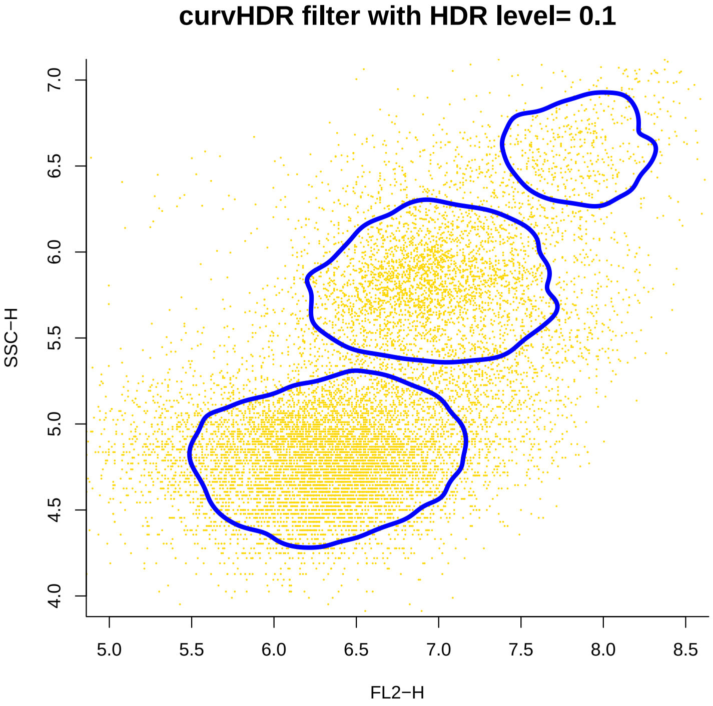

Figure 3: curvHDR *filter of data in the upper-left panel of Figure 1 (corresponding to 6 days before transplant). The HDR level parameter is equal to 0.1.*

```
> inputData <- exprs(GvHDtrans$s9a01)[, 2]
> cHfObj3 <- curvHDRfilter(inputData, HDRlevel = 0.01)
> cHfObj4 <- curvHDRfilter(inputData, HDRlevel = 0.8)
> par(mfrow = c(2, 1))
> plot(cHfObj3, xlab = "SSC-H", xlim = c(4, 7))
> plot(cHfObj4, xlab = "SSC-H", xlim = c(4, 7))
```

The resulting plot is shown in Figure 4. In the univariate case, the filters correspond to intervals, and are shown as blue bars at the base of the data histogram. Note that the sub-regions with `HDRlevel = 0.01` contain those with `HDRlevel = 0.8`.

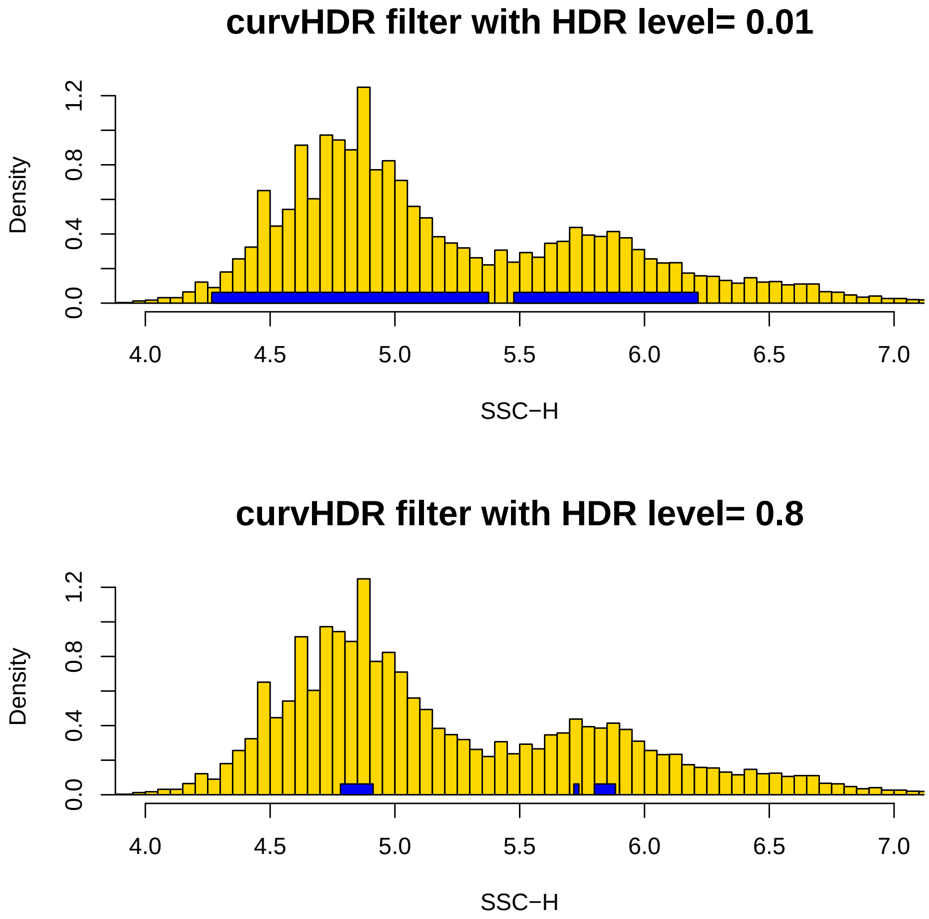

Figure 4: *Examples of univariate* curvHDR *filters with HDR levels equal to 0.01 and 0.8.*

Trivariate curvHDR

A somewhat novel feature of `curvHDRfilter()` is its support of *trivariate* input data. The filters take the form of triangle-faced polyhedra and can be visualised using the three-dimension graphics R packages `misc3d` (Feng & Tierney, 2009) and `rgl` (Adler & Murdoch, 2009) and the RGL device.

However, in the current release of curvHDR we have suppressed the trivariate code, since some aspects still need to be finalised before it is ready for public consumption. It is hoped that this finalisation will take place in mid-2010.

Longitudinal Example

We now return to the full longitudinal data introduced in the section titled 'Example Flow Cytometry Data'. Figure 5 shows the result of applying `HDRlevel=0.1` filters to all 7 scatterplots.

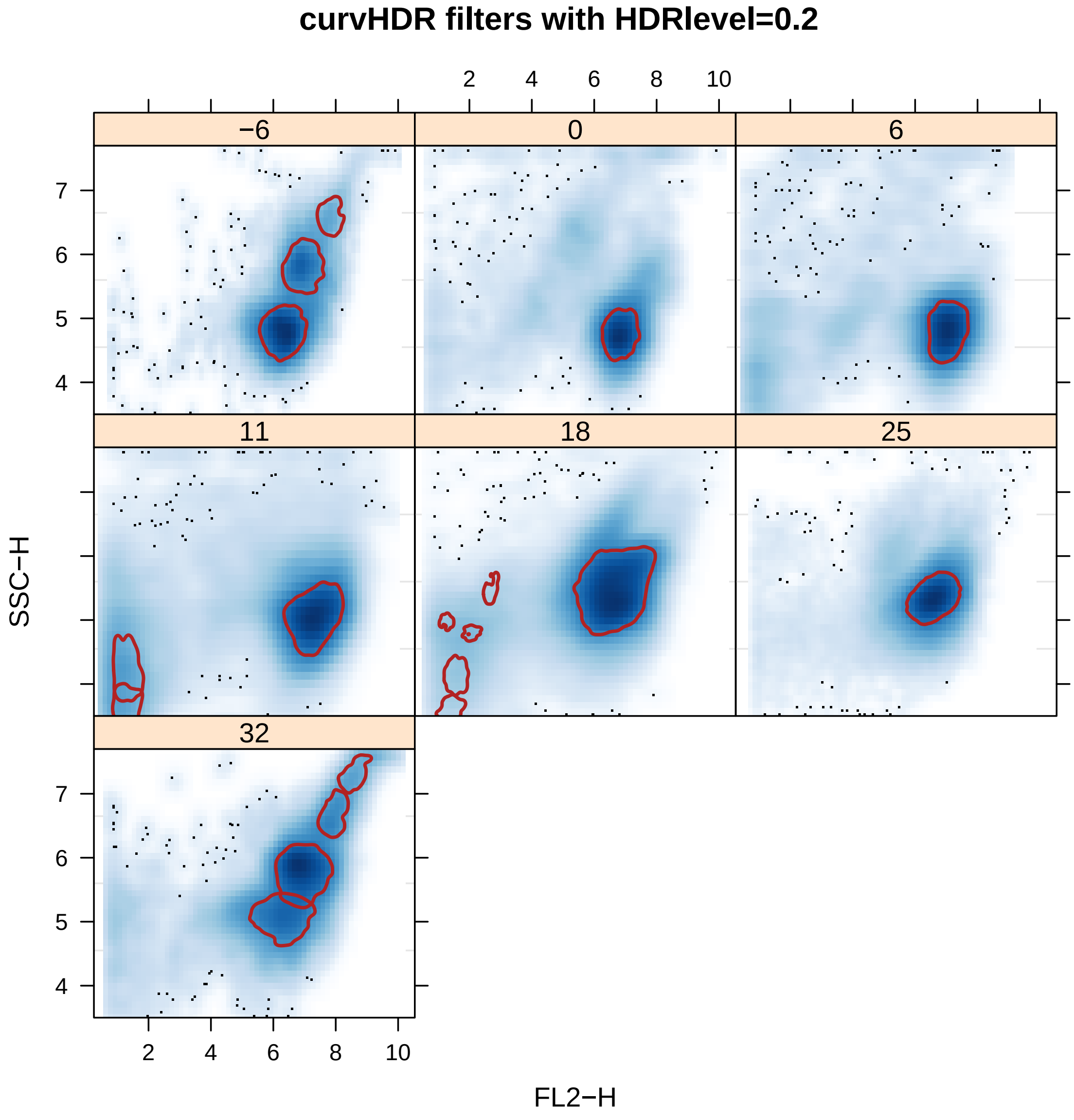

Figure 5: *The result from applying* curvHDR *filters (with* `HDRlevel=0.2`*) to data corresponding to each panel of Figure 1.*

Figure 5 was produced using the following code:

```
> GvHDneat <- list(exprs(GvHDtrans$s9a01)[, c(4, 2)], exprs(GvHDtrans$s9a02)[,
+     c(4, 2)], exprs(GvHDtrans$s9a03)[, c(4, 2)], exprs(GvHDtrans$s9a04)[,
+     c(4, 2)], exprs(GvHDtrans$s9a05)[, c(4, 2)], exprs(GvHDtrans$s9a06)[,
+     c(4, 2)], exprs(GvHDtrans$s9a07)[, c(4, 2)])
> cHfObjAll <- list()
> for (i in 1:7) cHfObjAll[[i]] <- curvHDRfilter(GvHDneat[[i]],
+     HDRlevel = 0.1)
> DaysVals <- c(-6, 0, 6, 11, 18, 25, 32)
> GvHDneatDf <- NULL
> for (i in 1:7) GvHDneatDf <- rbind(GvHDneatDf, cbind(GvHDneat[[i]],
+     rep(DaysVals[i], nrow(GvHDneat[[i]]))))
> GvHDneatDf <- as.data.frame(GvHDneatDf)
> dimnames(GvHDneatDf)[[2]] <- c("FL2.H", "SSC.H", "Days")
> GvHDcH <- xyplot(SSC.H ~ FL2.H | factor(Days), data = GvHDneatDf,
+     ylim = c(3.5, 7.7), xlab = "FL2-H", ylab = "SSC-H", layout = c(3,
+         3), as.table = TRUE, main = "curvHDR filters with HDRlevel=0.2",
+     panel = function(x, y) {
+         dayNum <- panel.number()
+         panel.grid()
+         panel.smoothScatter(x, y)
+         for (k in 1:length(cHfObjAll[[dayNum]]$polys))
+                 panel.polygon(cHfObjAll[[dayNum]]$polys[[k]],
+                 border = "firebrick", lwd = 2)
+     })
> print(GvHDcH)
```

References

Adler, D. & Murdoch, D. (2009). rgl 0.71: 3D visualization device system (OpenGL). R package `http://cran.r-projet.org`.

Brinkman, R.R, Gasparetto, M., Lee, S.-J.J., Ribickas, A.J., Perkins, J., Janssen, W., Smiley, R. and Smith, C. (2007). High-content flow cytometry and temporal data analysis for defining a cellular signature of graft-versus-host disease. *Biology of Blood and Marrow Transplantation*, **13**, 691–700.

Feng, D. & Tierney, L. (2009). misc3d 0.6-1: Miscellaneous 3D plots. R package `http://cran.r-projet.org`.

Hyndman, R.J. (1996). Computing and graphing highest density regions. *The American Statistician*, **50**, 120–126.

Naumann, U. & Wand, M.P. (2009). Automation in high-content flow cytometry screening. *Cytometry A*, **75A**, 789–797.

Naumann, U., Luta, G. and Wand, M.P. (2009). The curvHDR method for gating flow cytometry samples. *BMC Bioinformatics*, **11:44**, 1–13.

Shapiro, H.M. (2003). *Practical Flow Cytometry, 4th Edition.* John Wiley & Sons, New York.