Analyzing Flow Cytometry Data with Bioconductor

Nolwenn Le Meur, Deepayan Sarkar, Errol Strain, Byron Ellis, Perry Haaland, Florian Hahne

Fred Hutchinson Cancer Research Center

6 August 2007

Flow Cytometry

- High throughput, high volume, high dimensional data
- Software challenges
  - implementing standard methods
  - developing novel methods
- R + Bioconductor
  - open source
  - supports rapid prototyping
  - large existing base of users

Bioconductor packages for FCM data

- Existing R packages: *prada*, *rflowcyt*
  - underlying data structures different
- *flowCore*: attempt at standardization
  - data input
  - object model
  - standard tools (gates, transformations)
- *flowViz*
  - flexible visualization using *flowCore* infrastructure
- Potentially more packages
  - building on top of the basic infrastructure
  - implementing future developments

Goals of Lab Session

- Introduction to *flowCore*
- Some simple analysis and visualization

File formats

- Standard formats
  - FCS 2, FCS 3
  - LMD (list mode) files
- *flowCore* can read all these formats
  - read.FCS() – single file
  - read.FCSheader() – header only
  - read.flowSet() – collection of files

Reading in a file

read.FCS() reads individual files

```
> ff <- read.FCS("../data/0877408774.F06")
> ff
```

```
flowFrame object with 10000 cells and 8 observables:
<FSC-H> FSC-H <SSC-H> SSC-H <FL1-H> <FL2-H> <FL3-H> <FL1-A> <FL4
slot 'description' has 147 elements
```

```
> exprs(ff)[c(1:5, 9996:10000),
            c("FSC-H", "SSC-H", "FL1-H", "FL2-H", "Time")]
```

	FSC-H	SSC-H	FL1-H	FL2-H	Time
[1,]	528	672	8.446683	6.163591	7
[2,]	519	645	11.785791	5.336699	7
[3,]	449	136	10.022534	15.440480	7
[4,]	269	319	95.168807	35.992104	7
[5,]	361	138	10.113176	1318.969048	7
[6,]	373	122	22.134185	1260.910502	464
[7,]	349	131	24.002372	1.154944	464
[8,]	213	179	43.093186	31.163494	464
[9,]	388	183	15.580123	4.110368	464
[10,]	426	177	13.249199	840.874501	464

```
> pData(parameters(ff))
     name                desc range
$P1 FSC-H               FSC-H  1024
$P2 SSC-H               SSC-H  1024
$P3 FL1-H                      1024
$P4 FL2-H                      1024
$P5 FL3-H                      1024
$P6 FL1-A                <NA>  1024
$P7 FL4-H                      1024
$P8  Time Time (51.20 sec.)    1024
```

```
> str(head(keyword(ff), 15))

List of 15
 $ FCSversion: chr "2"
 $ $BYTEORD  : chr "4,3,2,1"
 $ $DATATYPE : chr "I"
 $ $NEXTDATA : chr "0"
 $ $SYS      : chr [1:4] "Macintosh" "System" "Software" "9.2.2"
 $ CREATOR   : chr [1:2] "CELLQuest<aa>" "3.3"
 $ $TOT      : chr "10000"
 $ $MODE     : chr "L"
 $ $PAR      : chr "8"
 $ $P1N      : chr "FSC-H"
 $ $P1R      : chr "1024"
 $ $P1B      : chr "16"
 $ $P1E      : chr "0,0"
 $ $P2N      : chr "SSC-H"
 $ $P2R      : chr "1024"
```

Reading in multiple files

read.flowSet() reads in a collection of files

```
> fset <- read.flowSet(...)

> fset

A flowSet with 35 experiments.

  rowNames: s5a01, s5a02, ..., s10a07 (35 total)
  varLabels and varMetadata:
    Patient: Patient code
    Visit: Visit number
    ...: ...
    name: NA
    (5 total)

  column names:
  FSC-H SSC-H FL1-H FL2-H FL3-H FL2-A FL4-H Time
```

Individual frames can be accessed using [[and $

```
> fset[[2]]
```

```
flowFrame object with 3405 cells and 8 observables:
<FSC-H> FSC-Height <SSC-H> SSC-Height <FL1-H> CD15 FITC <FL2-H>
slot 'description' has 153 elements
```

```
> fset$"s5a02"
```

```
flowFrame object with 3405 cells and 8 observables:
<FSC-H> FSC-Height <SSC-H> SSC-Height <FL1-H> CD15 FITC <FL2-H>
slot 'description' has 153 elements
```

Subsets of a *"flowSet"* can be extracted using [

```
> fset[1:5]
```

```
A flowSet with 5 experiments.

  rowNames: s5a01, s5a02, ..., s5a05 (5 total)
  varLabels and varMetadata:
    Patient: Patient code
    Visit: Visit number
    ...: ...
    name: NA
    (5 total)

  column names:
FSC-H SSC-H FL1-H FL2-H FL3-H FL2-A FL4-H Time
```

fsApply(): applies function on all frames in a *"flowSet"*

```
> head(fsApply(fset, nrow))
```

```
       [,1]
s5a01  3420
s5a02  3405
s5a03  3435
s5a04  8550
s5a05 10410
s5a06  3750
```

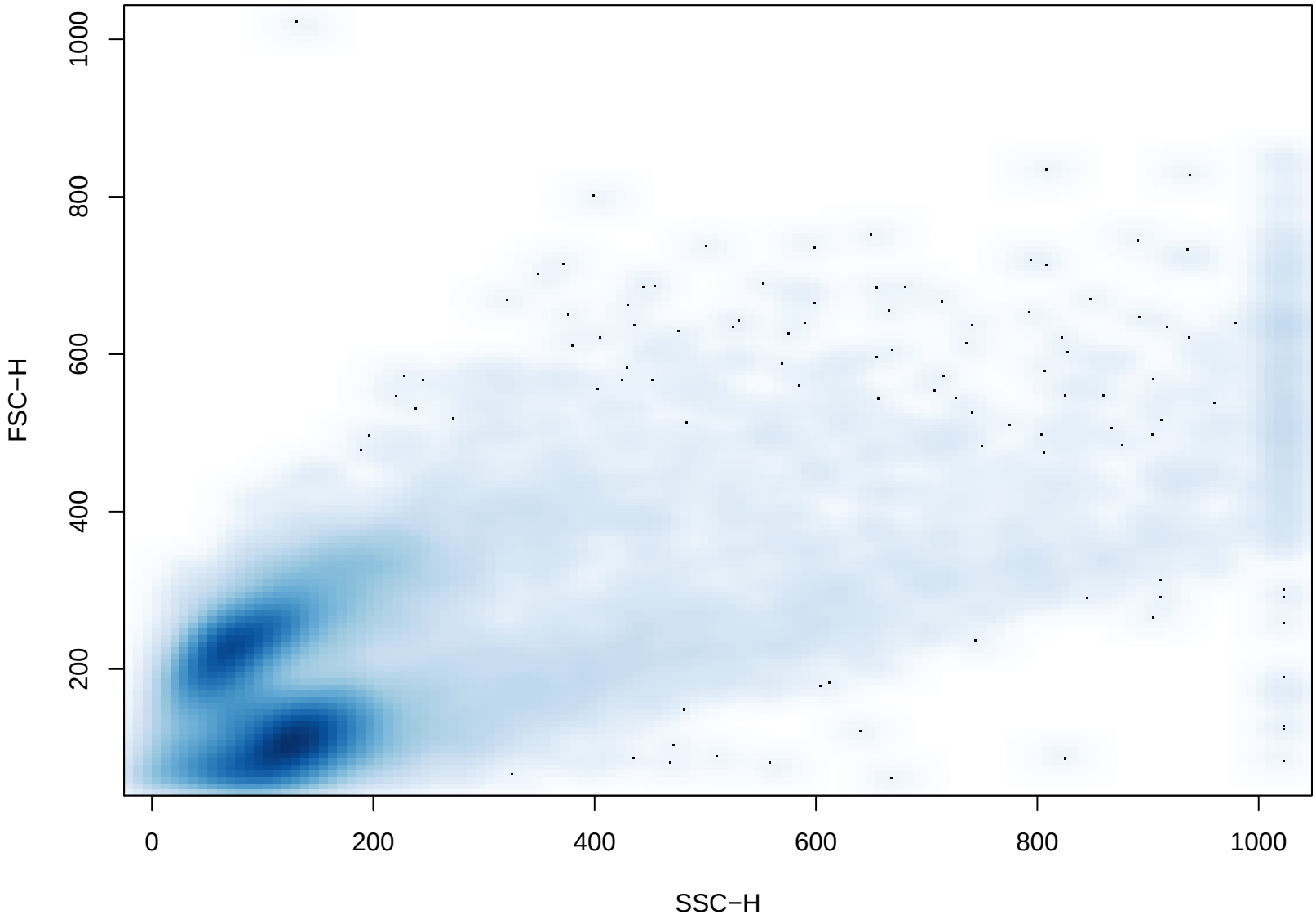
Phenodata manipulation

```
> varMetadata(phenoData(fset))
        labelDescription
Patient     Patient code
Visit       Visit number
Days    Days since graft
Grade   Grade (leukemia)
name                 <NA>
> varMetadata(phenoData(fset))["name", "labelDescription"] <-
      "File name"
> phenoData(fset)
  rowNames: s5a01, s5a02, ..., s10a07 (35 total)
  varLabels and varMetadata:
    Patient: Patient code
    Visit: Visit number
    ...: ...
    name: File name
    (5 total)
```

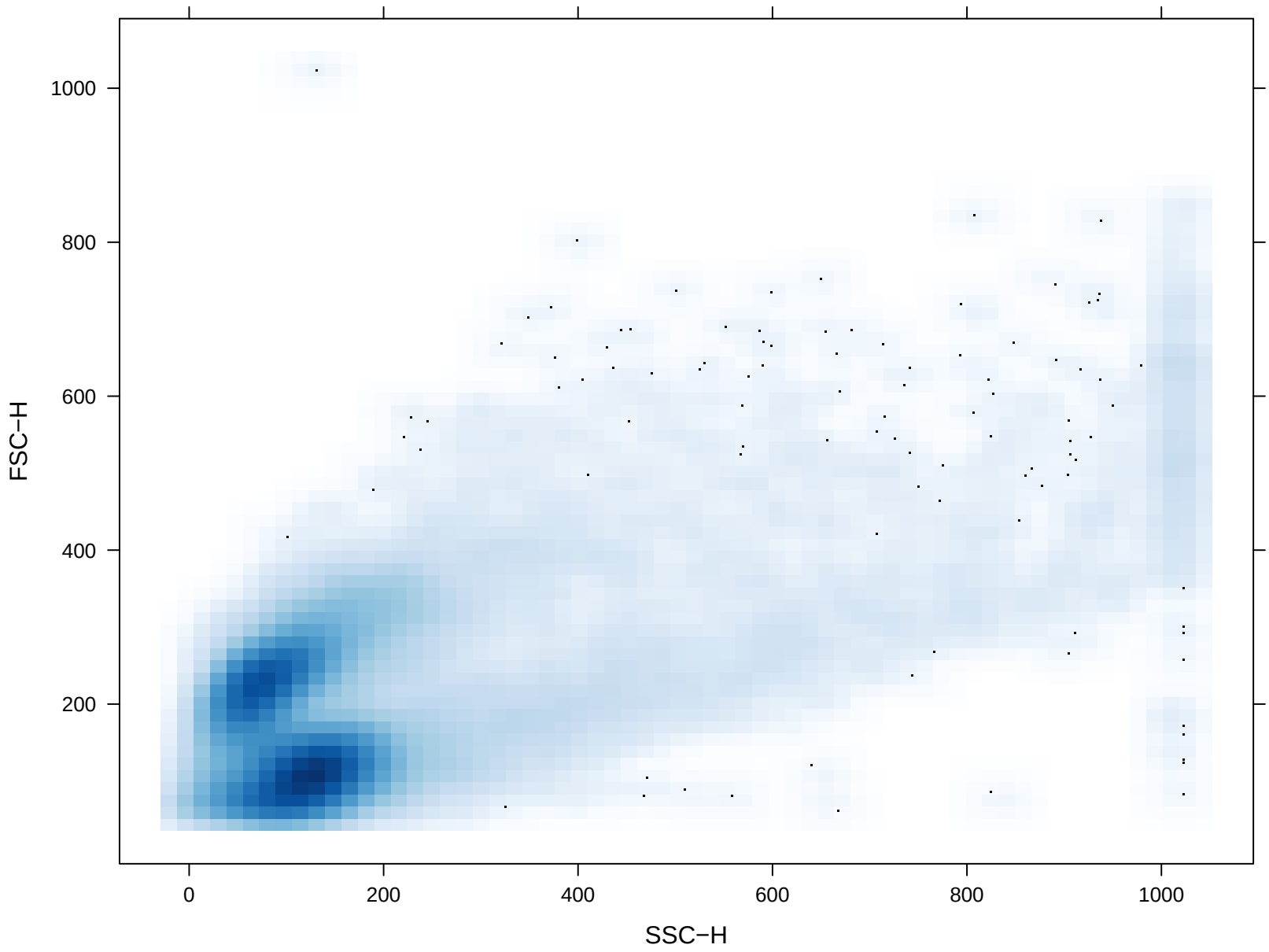
Visualization

- *flowCore* has some basic plots
- *flowViz* has more flexible methods based on *lattice*

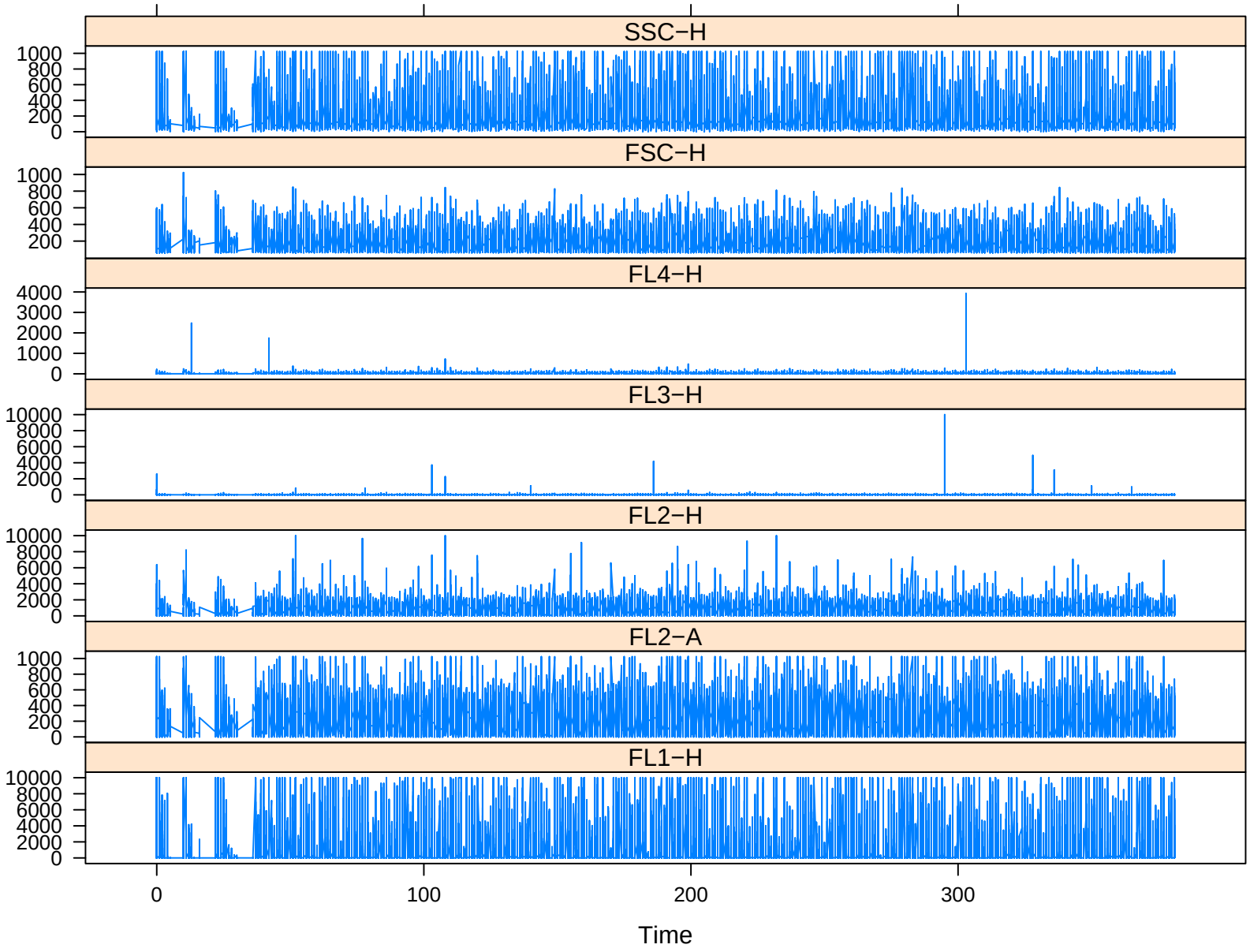
```
> plot(fset$"s10a06", c("SSC-H", "FSC-H"))
```



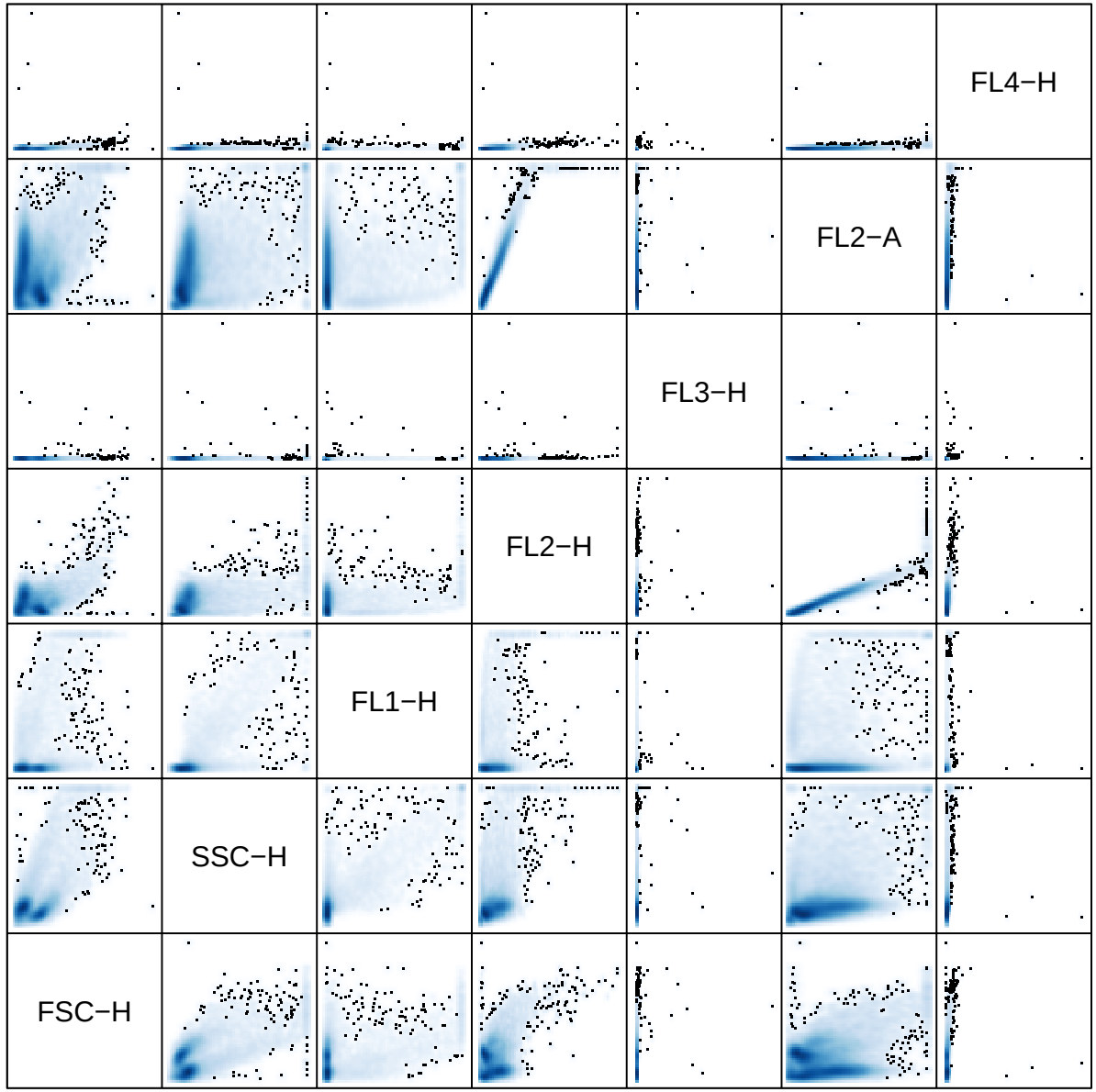
```
> xyplot(`FSC-H` ~ `SSC-H`, data = fset$"s10a06")
```



```
> xyplot(fset$"s10a06")
```



```
> splom(fset$"s10a06")
```



Scatter Plot Matrix

Transformations

- Original scale often not very good for visualization
- The transform() function creates transformed versions

```
> summary(transform(fset$"s10a06",
                    asinh.FSC.H = asinh(`FSC-H`)))
```

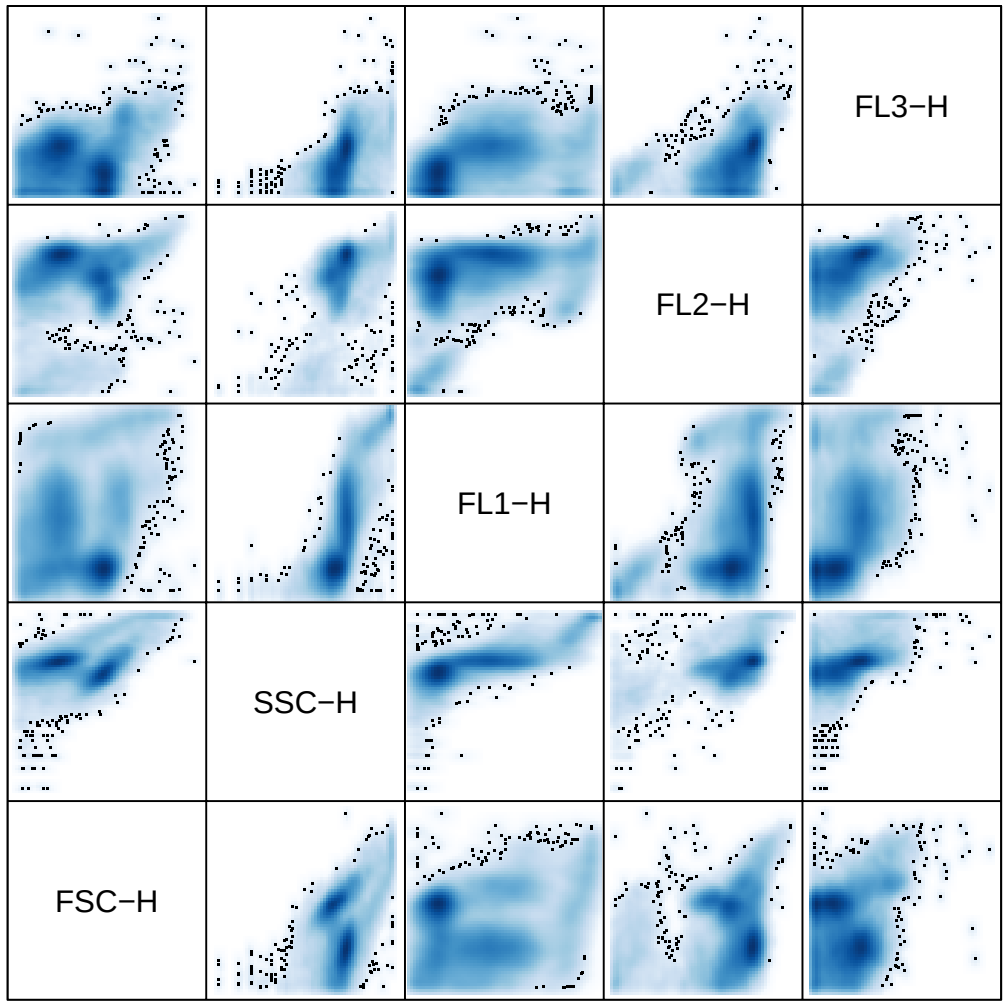
```
          FSC-H   SSC-H     FL1-H   FL2-H     FL3-H  FL2-A     FL4-
Min.       60.0    0.0     1.000     1.0     1.000    0.0     1.00
1st Qu.   103.0   77.0     3.656   338.7     2.460   83.0     1.82
Median    145.0  113.0    11.680   708.7     5.684  175.0    10.77
Mean      175.9  145.1   431.500   871.7    13.270  218.2    27.27
3rd Qu.   232.0  149.0    94.320  1307.0    13.010  325.0    49.32
Max.     1023.0 1023.0 10000.000 10000.0 10000.000 1023.0 3921.00
       asinh.FSC.H
Min.         4.788
1st Qu.      5.328
Median       5.670
Mean         5.729
3rd Qu.      6.140
```

Transformations

```
> summary(transform("FSC-H" = asinh) %on% fset$s10a06)
```

	FSC-H	SSC-H	FL1-H	FL2-H	FL3-H	FL2-A	FL4-H
Min.	4.788	0.0	1.000	1.0	1.000	0.0	1.000
1st Qu.	5.328	77.0	3.656	338.7	2.460	83.0	1.828
Median	5.670	113.0	11.680	708.7	5.684	175.0	10.770
Mean	5.729	145.1	431.500	871.7	13.270	218.2	27.270
3rd Qu.	6.140	149.0	94.320	1307.0	13.010	325.0	49.320
Max.	7.624	1023.0	10000.000	10000.0	10000.000	1023.0	3921.000

```
> s10a06 <- fset$"s10a06"[, 1:5]
> splom(transform("FSC-H" = asinh, "SSC-H" = asinh,
                  "FL1-H" = asinh, "FL2-H" = asinh,
                  "FL3-H" = asinh) %on% s10a06)
```



Scatter Plot Matrix

Standard Transformations

truncateTransform $y = \begin{cases} a & x < a \\ x & x \geq a \end{cases}$

scaleTransform $f(x) = \frac{x-a}{b-a}$

linearTransform $f(x) = a + bx$

quadraticTransform $f(x) = ax^2 + bx + c$

lnTransform $f(x) = \log(x)\frac{r}{d}$

logTransform $f(x) = \log_b(x)\frac{r}{d}$

biexponentialTransform $f^{-1}(x) = ae^{bx} - ce^{dx} + f$

logicleTransform A special form of the biexponential transform with parameters selected by the data.

arcsinhTransform $f(x) = asinh(a + bx) + c$

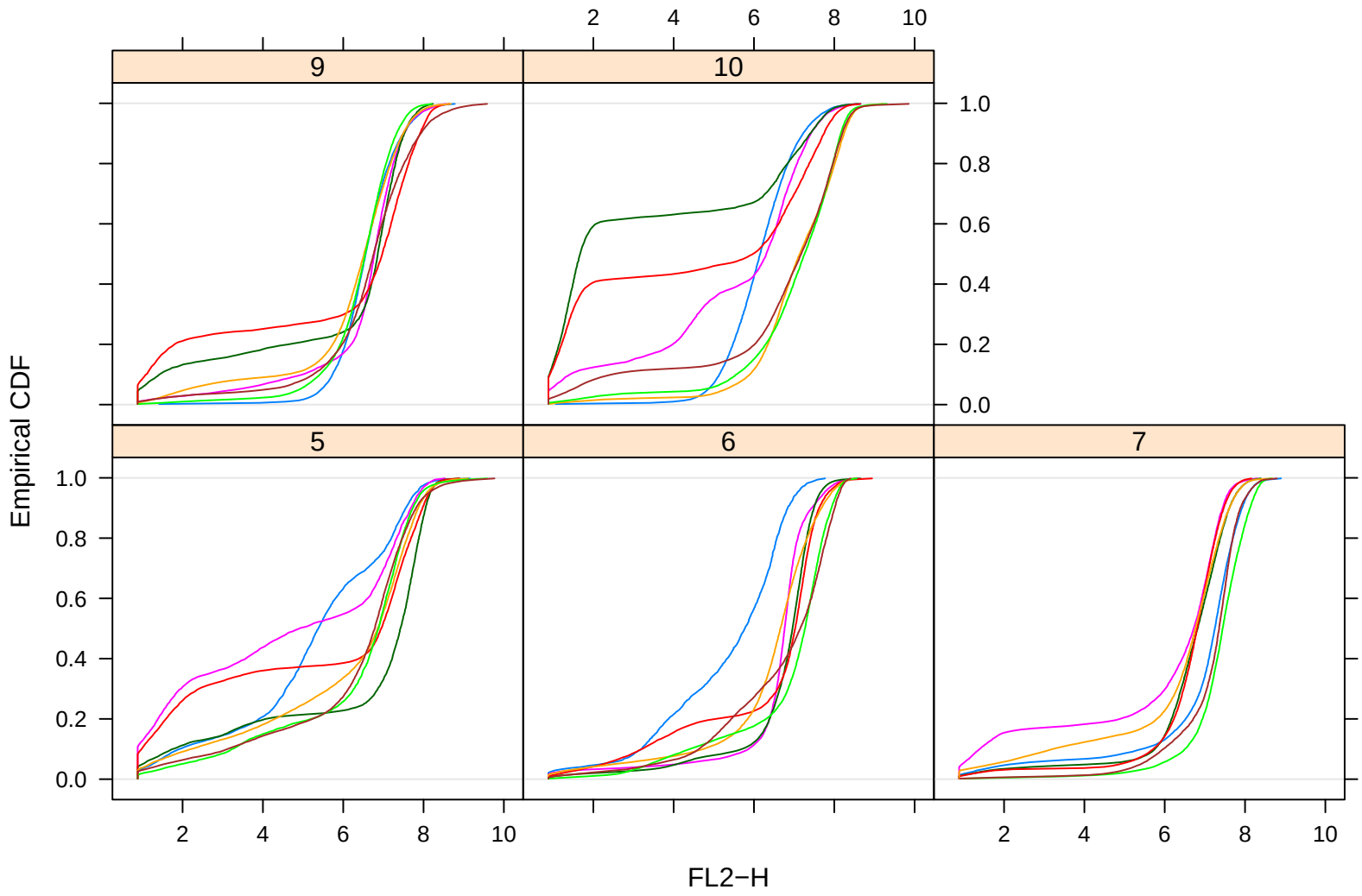
Transforms can be applied on an entire *"flowFrame"*

```
> fset.trans <-
      transform("FSC-H" = asinh, "SSC-H" = asinh,
                "FL1-H" = asinh, "FL2-H" = asinh,
                "FL3-H" = asinh) %on% fset
```

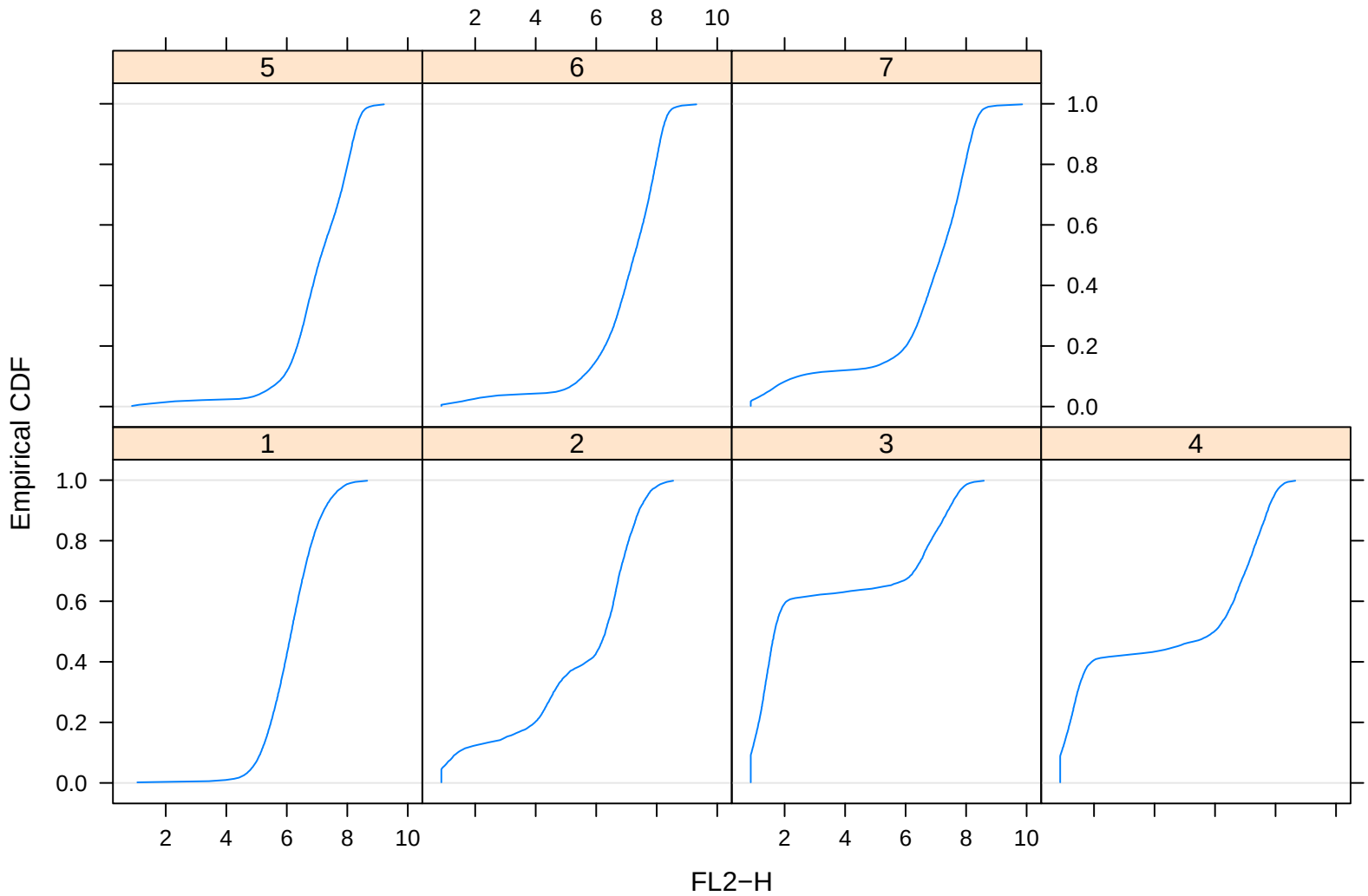
Quality Assessment

- Conditional plots (*lattice* + *flowViz*)
- Numerical summaries

```
> ecdfplot(~ `FL2-H` | Patient, fset.trans, groups = Visit)
```



```
> ecdfplot(~ `FL2-H` | Visit, fset.trans,
          subset = Patient == "10")
```

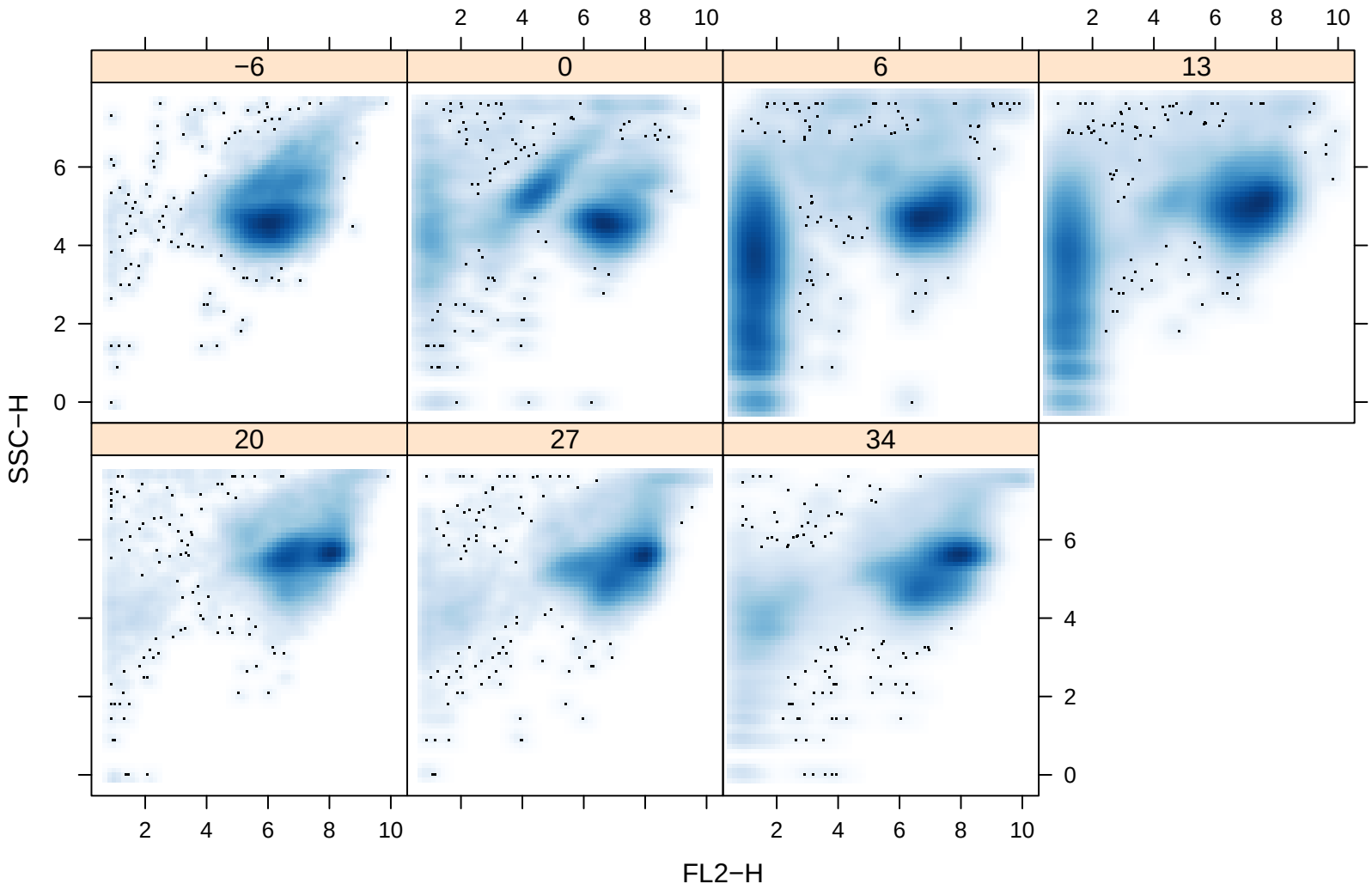


Numerical quality measures

```
> fsApply(fset.trans, each_col, IQR)[29:35, 1:4]
```

	FSC-H	SSC-H	FL1-H	FL2-H
s10a01	0.6044971	0.8265640	1.5110417	1.098386
s10a02	0.8865547	0.9066027	1.1048425	2.608522
s10a03	0.5024268	1.9429009	0.7189971	5.341477
s10a04	0.7710728	1.5766873	1.1472946	5.871970
s10a05	0.6435293	0.4883394	3.3556510	1.404507
s10a06	0.8119895	0.6601100	3.2320107	1.350488
s10a07	0.7519678	0.8953221	2.9799957	1.557562

```
> xyplot(`SSC-H` ~ `FL2-H` | factor(Days), fset.trans,
         subset = Patient == "10")
```



Standard Filters (a.k.a. Gates)

rectangleGate Cubic shape in one or more dimensions (e.g. interval gates)

polygonGate Two dimensional polygonal gate

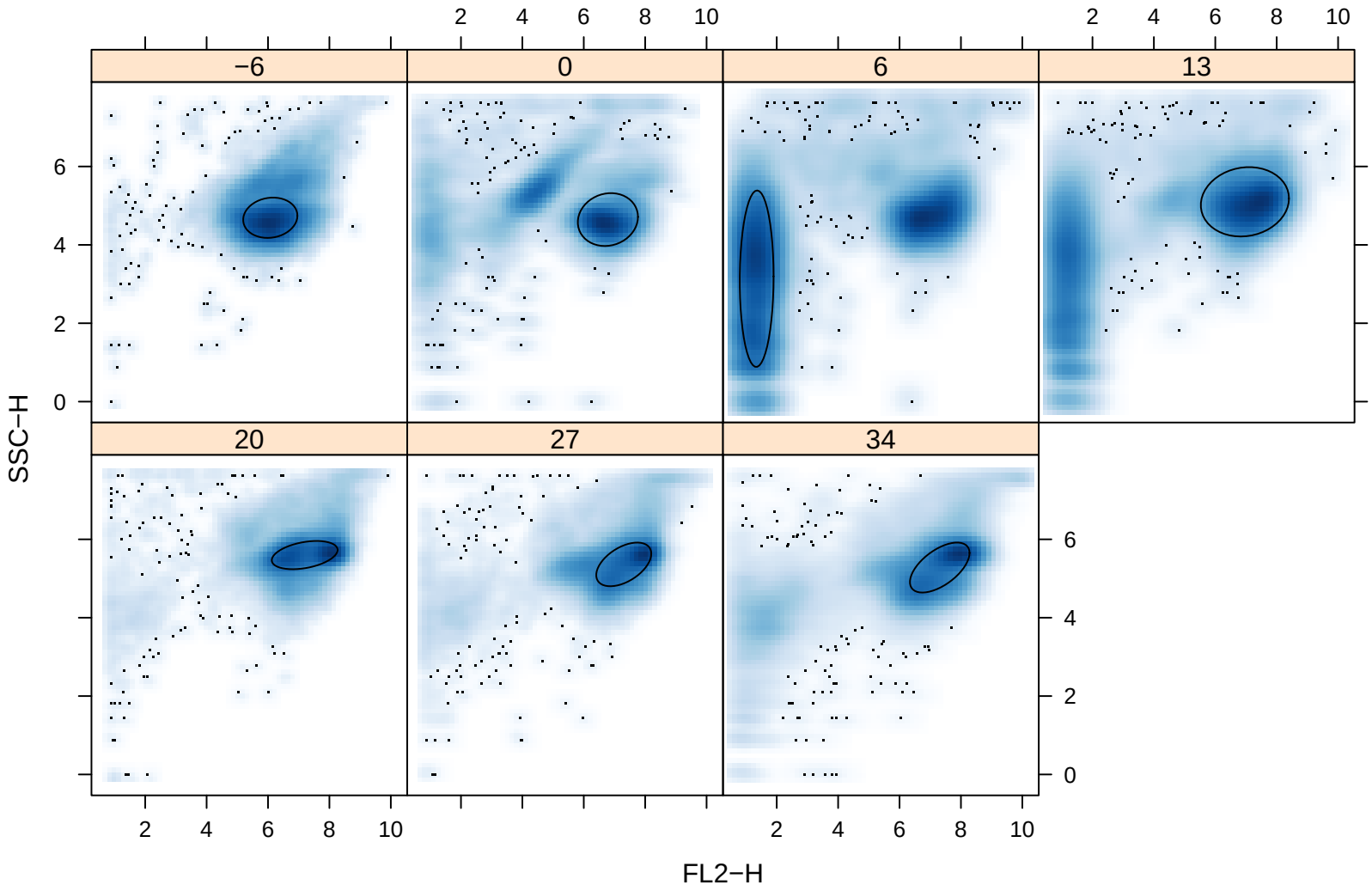
polytopeGate Convex hull of a set of points, possibly in more than two dimensions

ellipsoidGate Ellipsoidal region in two or more dimensions

Data-driven Filters

norm2Filter	Finds a subregion that most resembles a bivariate Gaussian distribution
kmeansFilter	(Multiple) populations using one-dimensional k-means clustering

```
> xyplot(`SSC-H` ~ `FL2-H` | factor(Days), fset.trans,
         subset = Patient == "10",
         filter = norm2Filter("SSC-H", "FL2-H"))
```



```
> rect.gate <- rectangleGate("RangeGate", "FL2-H" = c(4, 10))
> rect.results <- filter(fset.trans, rect.gate)
> lapply(rect.results[29:35], summary)
$s10a01
RangeGate: 17117 of 17289 (99.01%)

$s10a02
RangeGate: 3609 of 4530 (79.67%)

$s10a03
RangeGate: 2496 of 6765 (36.90%)

$s10a04
RangeGate: 5402 of 9540 (56.62%)

$s10a05
RangeGate: 24924 of 25515 (97.68%)

$s10a06
RangeGate: 23597 of 24656 (95.70%)
```

```
> xyplot(`SSC-H` ~ `FL2-H` | factor(Days),
         subset = Patient == "10",
         data = Subset(fset.trans, rect.results),
         filter = norm2Filter("SSC-H", "FL2-H", scale = 2))
```