

Visual documentation of Barbouille

Benjamin Leblanc

21.05.2018

Contents

SideBySide and its derived functions	2
Data	3
Functions	3
Colors	6
Orientation	6
Axes	7
Separators	10
Binning	11
Quality	13
Shaping	14
Populations	15
Debug	17
Movie	17

My collection of plotting and color mapping functions, home cooked with base R graphics, ash and colorspace.

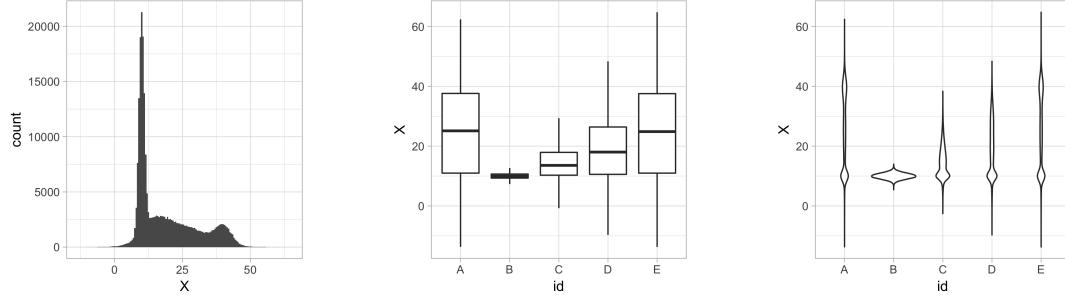
SideBySide and its derived functions

Here include a table of parameters with brief description and availability for SideBySideRepresentation/Distributions/Variations (BivariateRerepresentation?)

Category	Parameter	Type	Description	Default
Orientation	layout	char	“horizontal” or “vertical”	“h”
Axes	rng	num	represented range of X values	NULL
Axes	label	char	name of the X values axis	NULL
Axes	names	logic	show column names of X if available	TRUE
Axes	las	int		1
Separators	box	logic	show frame surrounding visualization slots	TRUE
Separators	grid	color	lines delimiting visualization slots	grey(0.5, alpha = 0.5)
Separators	spacing	int	blank space between visualization slots	0
Binning	bins	int	number of bins for the values of X	200
Binning	uv	int	bins in each distribution slot	50
Binning	bv	int	bins in each variation slot	50
Quality	sampling	num		5,00E+03
Quality	smoothing	int		5
Casting	spray	char	“uniform”, “triangle”, “normal” or “cosine”	“uniform”
Casting	stencil	char	“linear”, “cosine” or “sigmoid”	“linear”

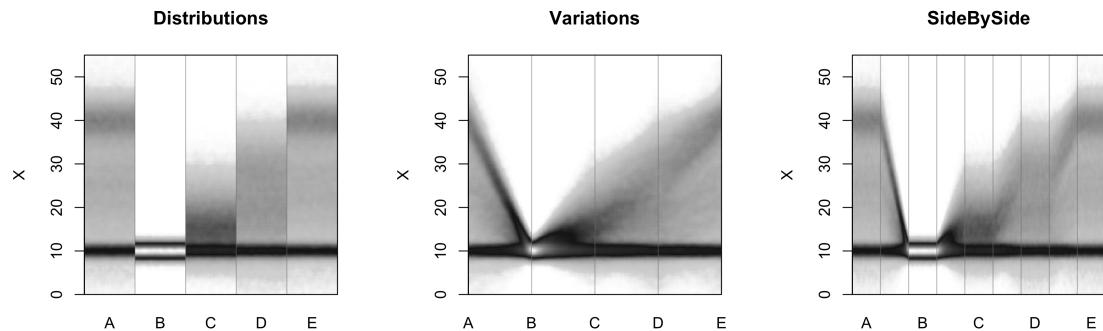
Data

```
ggplot(tbl, aes(x = X)) + geom_histogram(binwidth = 0.5) + theme_light()
ggplot(tbl, aes(x = id, y = X)) + geom_boxplot(outlier.color = NA) + theme_light()
ggplot(tbl, aes(x = id, y = X)) + geom_violin() + theme_light()
```

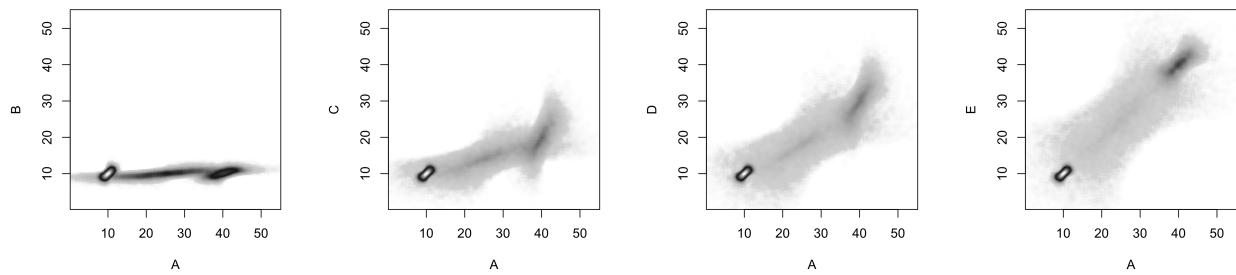


Functions

```
# To visualize distributions and variations separately
Distributions(X)
Variations(X)
# To visualize both distributions and variations
SideBySide(X)
```



```
# To visualize joint distributions
ScatterMaps(X, rng, x = "A", y = c("B", "C", "D", "E"), f = "split", colors = "grey")
```



```

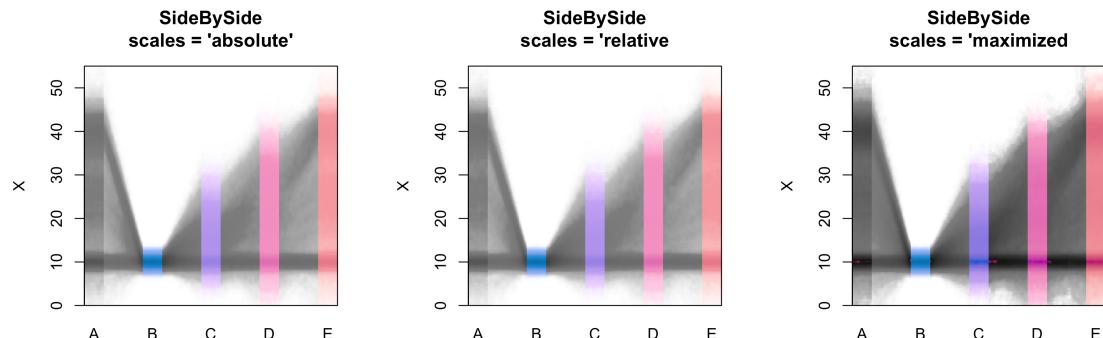
clr <- list(
  d = rgb(0, 0.5, 1),
  v = rgb(1, 0.5, 0),
  p = hsv(c(60, 90, 30) / 360),
  m = c("grey", rgb(0:3/3, 0, 3:0/3))
)
maps <- list(
  list(x = "A", y = "B"),
  list(x = "A", y = "C"),
  list(x = "A", y = "D"),
  list(x = "A", y = "E")
)

```

```

SideBySide(
  X, rng, db = 25, colors = list(d = clr[["m"]]), grid = NA
)

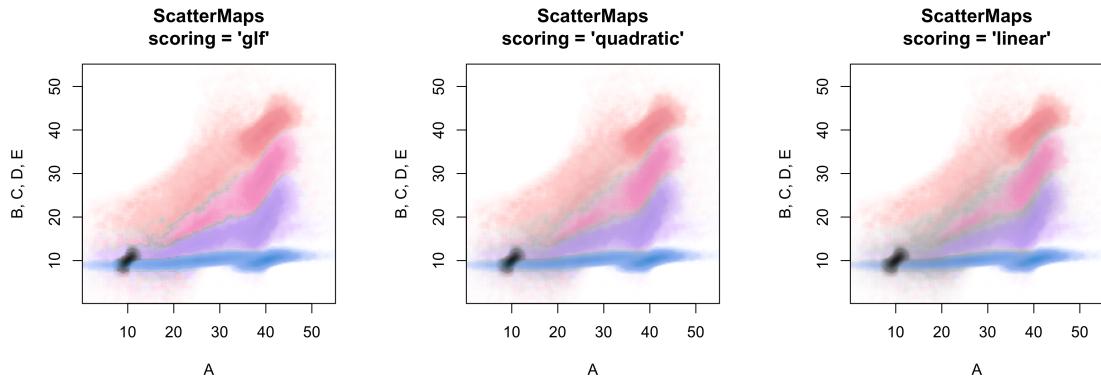
```



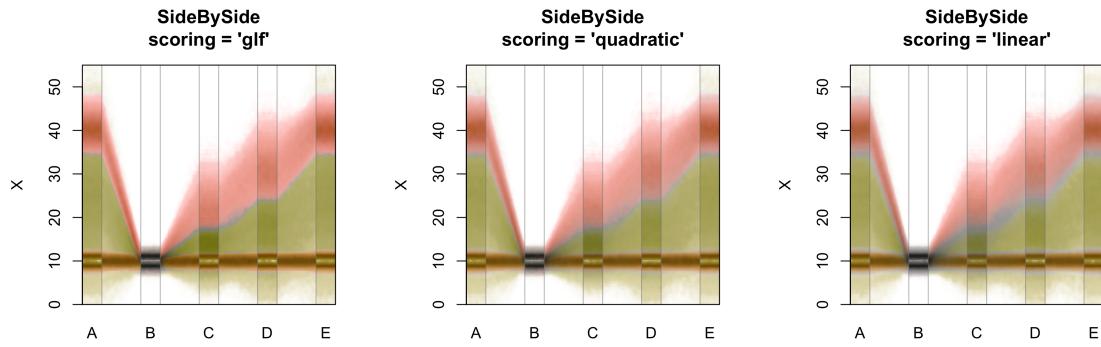
```

ScatterMaps(
  X, rng, maps = maps, layers = "maps",
  colors = list(m = clr[["m"]][-1])
)

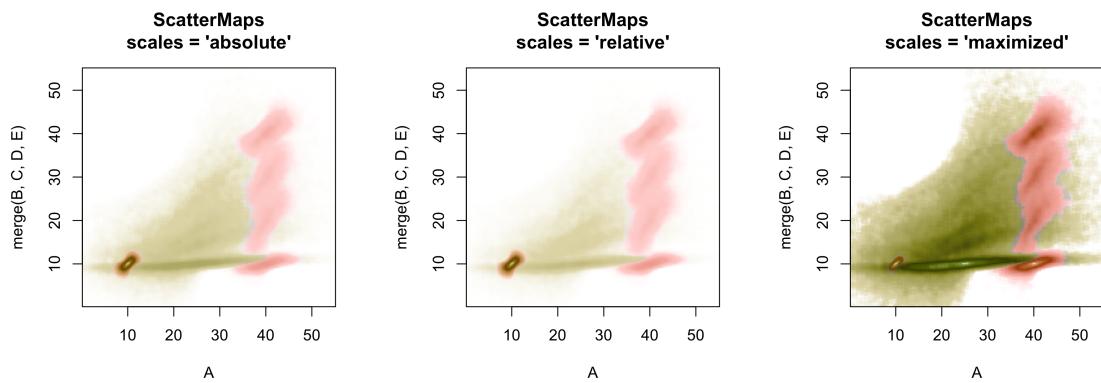
```



```
SideBySide(
  X, rng, db = 25, pops = grp, colors = list(p = clr[["p"]])
)
```

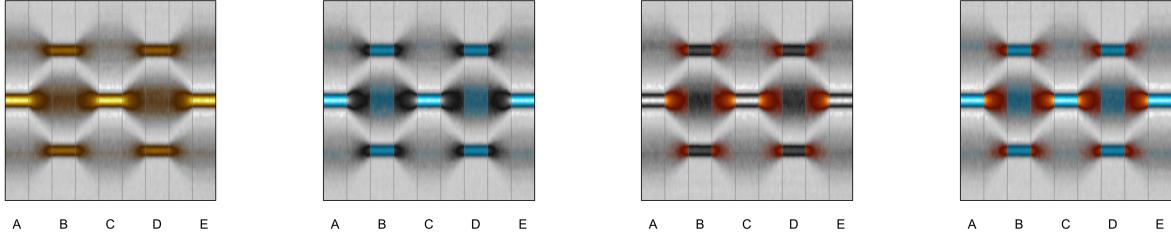


```
ScatterMaps(
  X, rng, pops = grp, x = "A", y = colnames(X)[-1],
  colors = list(p = clr[["p"]])
)
```



Colors

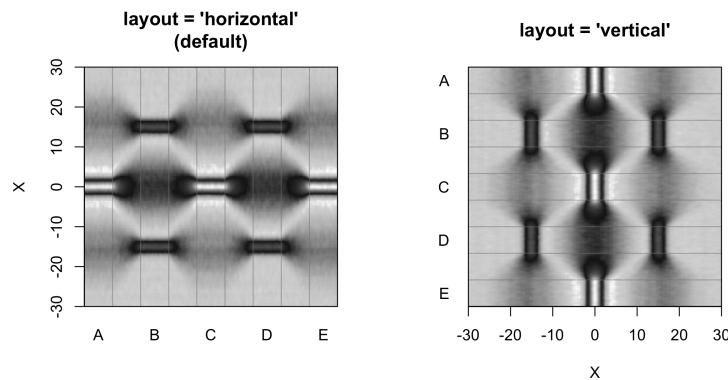
```
SideBySide(X, colors = rgb(1, 1, 0))
SideBySide(X, colors = c(d = rgb(0, 0.5, 1)))
SideBySide(X, colors = c(v = rgb(1, 0.5, 0)))
SideBySide(X, colors = c(d = rgb(0, 0.5, 1), v = rgb(1, 0.5, 0)))
```



Orientation

layout

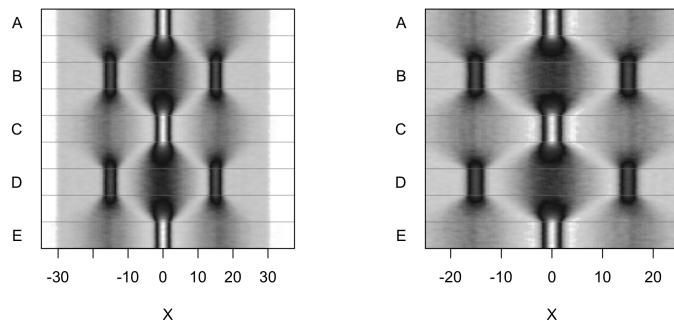
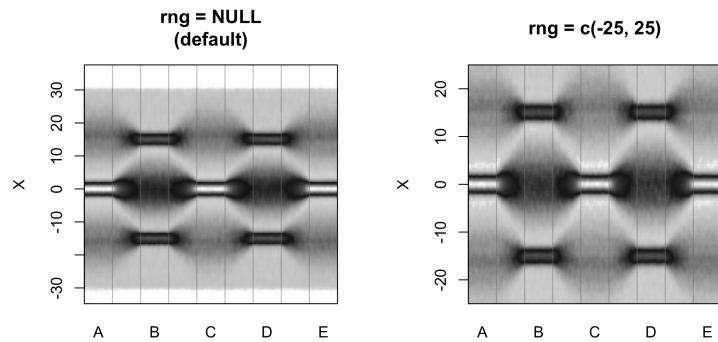
```
# Default value is layout = "horizontal"
SideBySide(X)
# Alternative layout values is "vertical"
# Unambiguous abbreviations can be used
SideBySide(X, layout = "v")
```



Axes

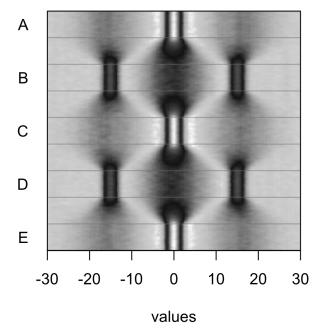
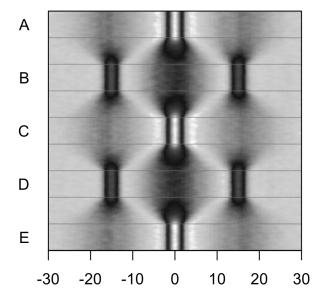
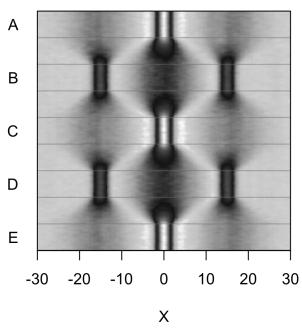
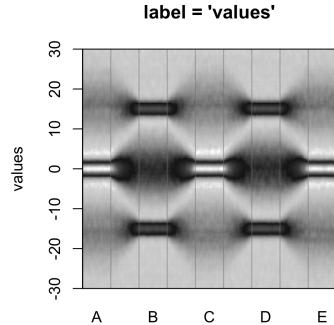
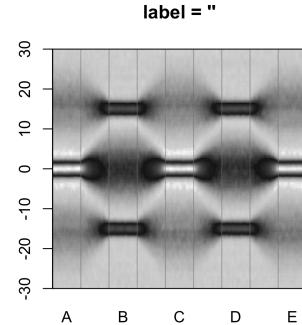
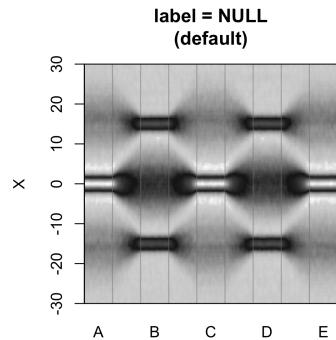
rng

```
# Default value is rng = NULL
SideBySide(X)
SideBySide(X, rng = c(-25, 25))
```



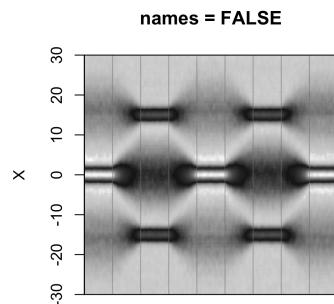
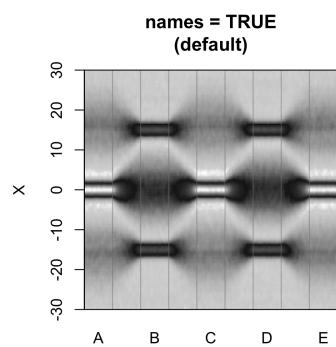
label

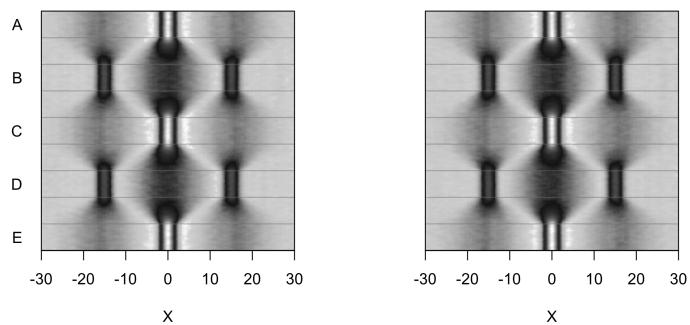
```
# Default value is label = NULL
SideBySide(X)
SideBySide(X, label = "")
SideBySide(X, label = "values")
```



names

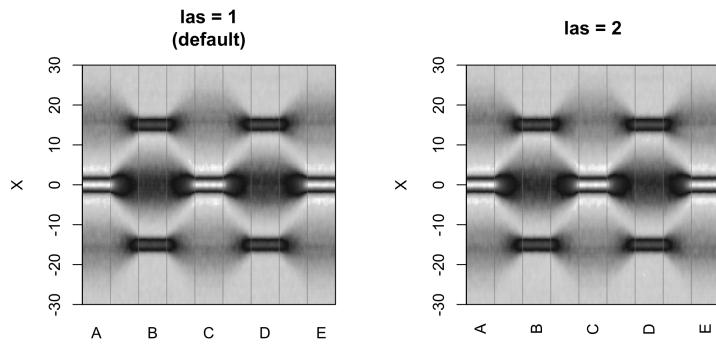
```
# Default value is names = TRUE
SideBySide(X)
SideBySide(X, names = F)
```





las

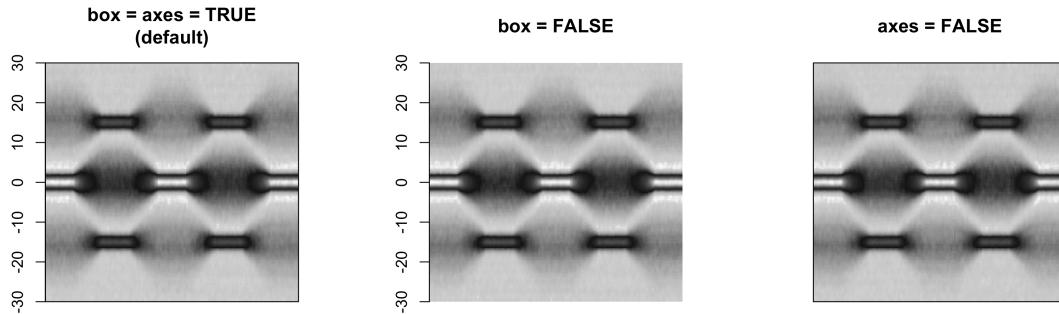
```
# Default value is las = 1
SideBySide(X)
SideBySide(X, las = 2)
```



Separators

box & axes

```
# Default value is box = TRUE
SideBySide(X)
SideBySide(X, box = F)
SideBySide(X, axes = F)
```



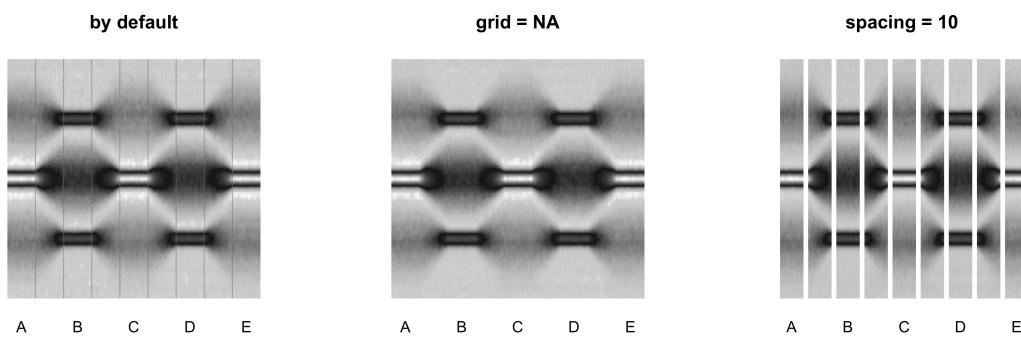
grid & spacing

Line delimiters and blank spaces between columns of X are mutually exclusive.

```
# By default
# grid      = grey(0.5, alpha = 0.5)
# spacing   = 0
SideBySide(X)

# grid = NA disables line delimiters
SideBySide(X, grid = NA)

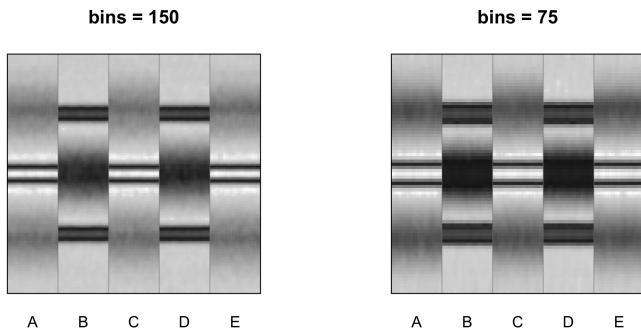
# non zero spacing disables line delimiters
SideBySide(X, spacing = 10)
```



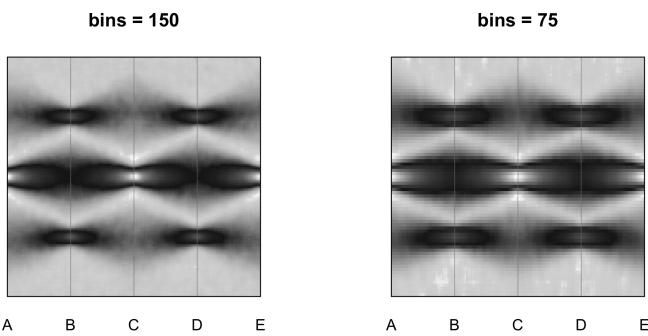
Binning

bins

```
# Default value is bins = 200
Distributions(X, bins = 150)
Distributions(X, bins = 75)
```

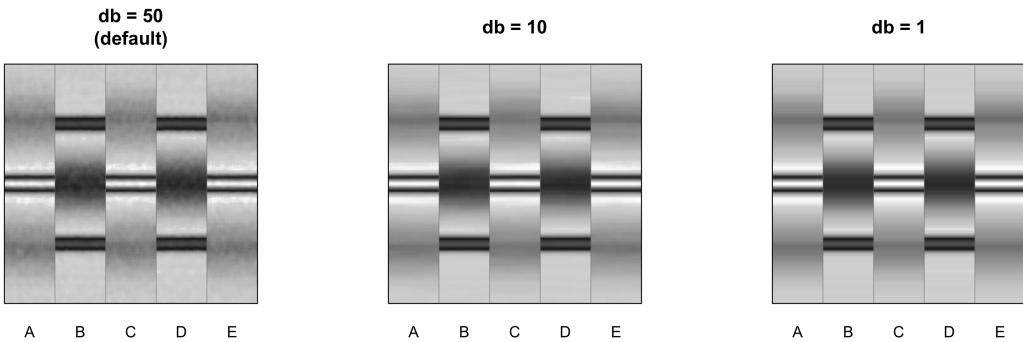


```
# Default value is bins = 200
Variations(X, bins = 150)
Variations(X, bins = 75)
```

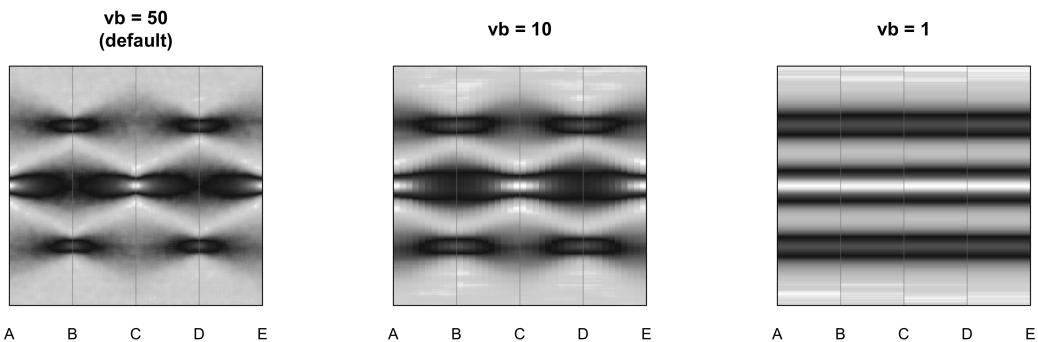


db & vb

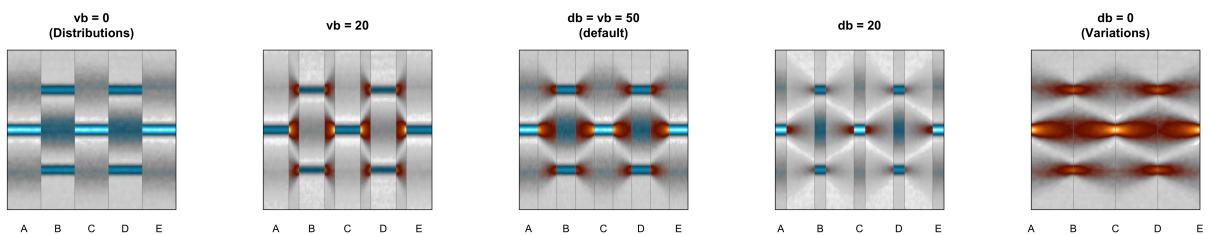
```
# Default value is db = 50
Distributions(X)
Distributions(X, db = 10)
Distributions(X, db = 1)
```



```
# Default value is vb = 50
Variations(X)
Variations(X, vb = 10)
Variations(X, vb = 1)
```



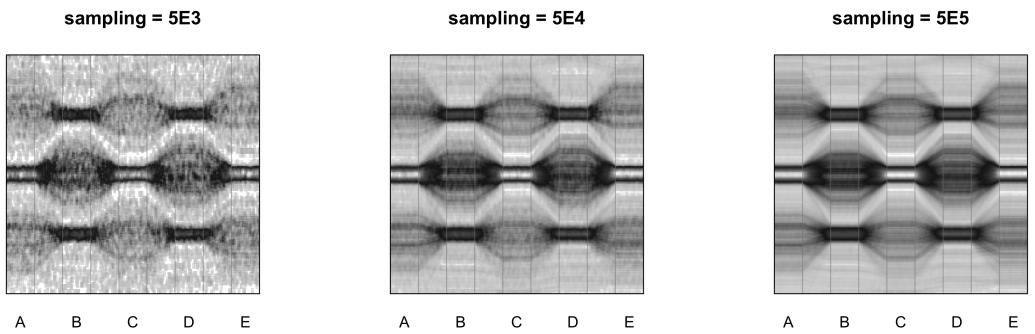
```
# By default
# db = 50
# vb = 50
SideBySide(X, vb = 0) # identical to Distributions(X)
SideBySide(X, vb = 20)
SideBySide(X)
SideBySide(X, db = 20)
SideBySide(X, db = 0) # identical to Variations(X)
```



Quality

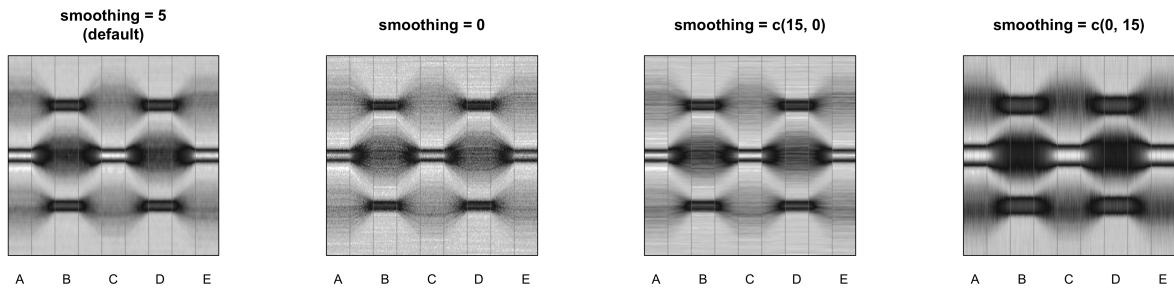
sampling

```
# Default value is sampling = 5E5
SideBySide(X, sampling = 5E3)
SideBySide(X, sampling = 5E4)
```



smoothing

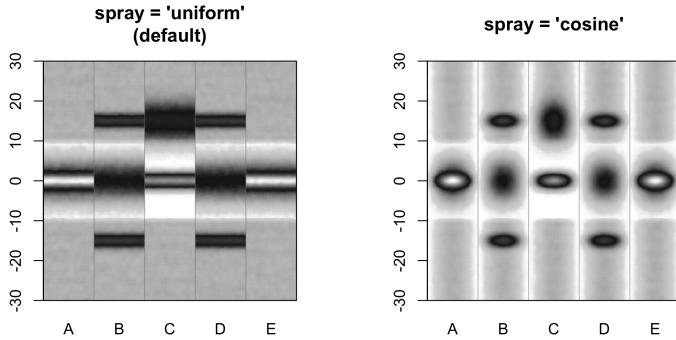
```
# Default value is smoothing = c(5, 5)
SideBySide(X)
SideBySide(X, smoothing = 0)
SideBySide(X, smoothing = c(15, 0))
SideBySide(X, smoothing = c(0, 15))
```



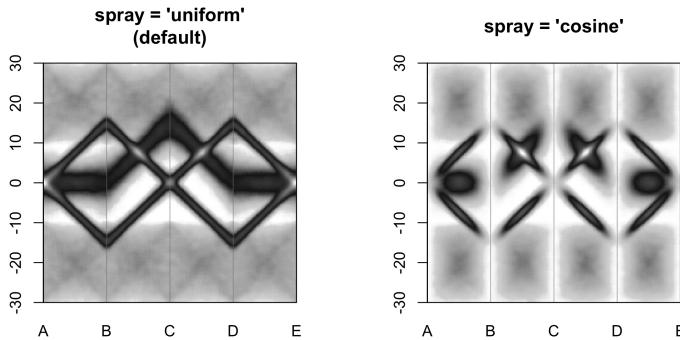
Shaping

spray

```
# Default value is spray = "uniform"
Distributions(X)
# Alternative spray values are "cosine", "normal" and "triangle"
# Unambiguous abbreviations can be used
Distributions(X, spray = "cos")
```



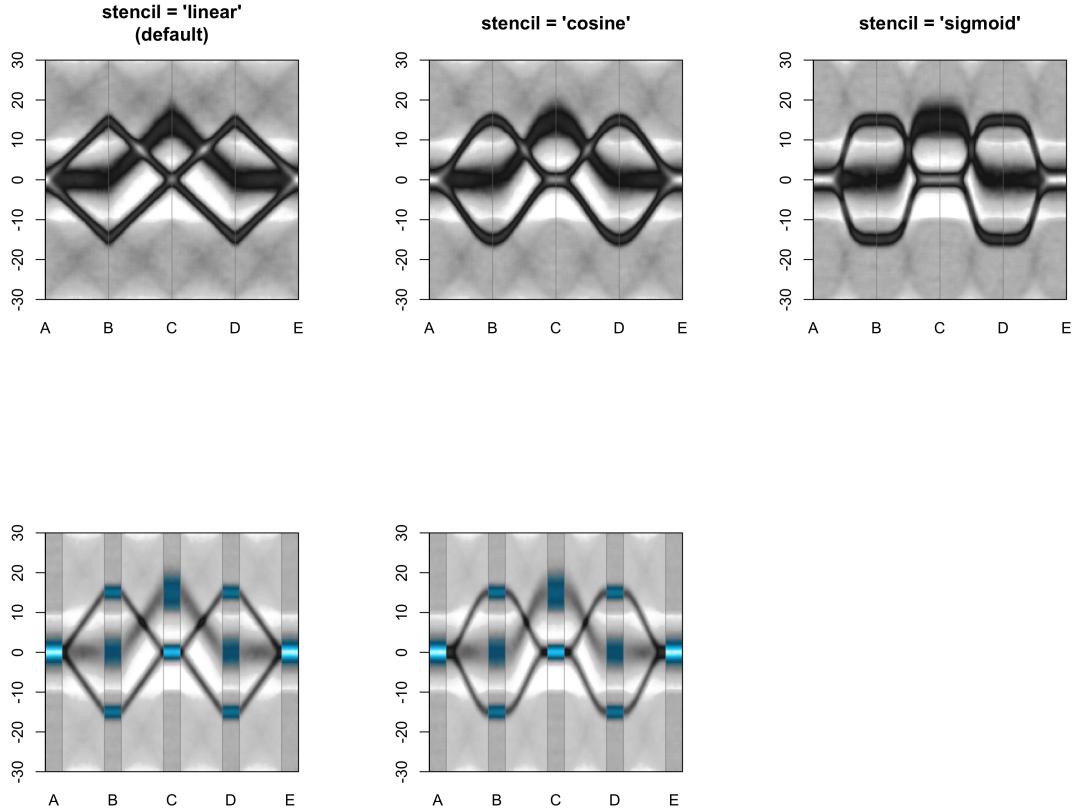
```
# Default value is spray = "uniform"
Variations(X)
# Alternative spray values are "cosine", "normal" and "triangle"
# Unambiguous abbreviations can be used
Variations(X, spray = "cos")
```



stencil

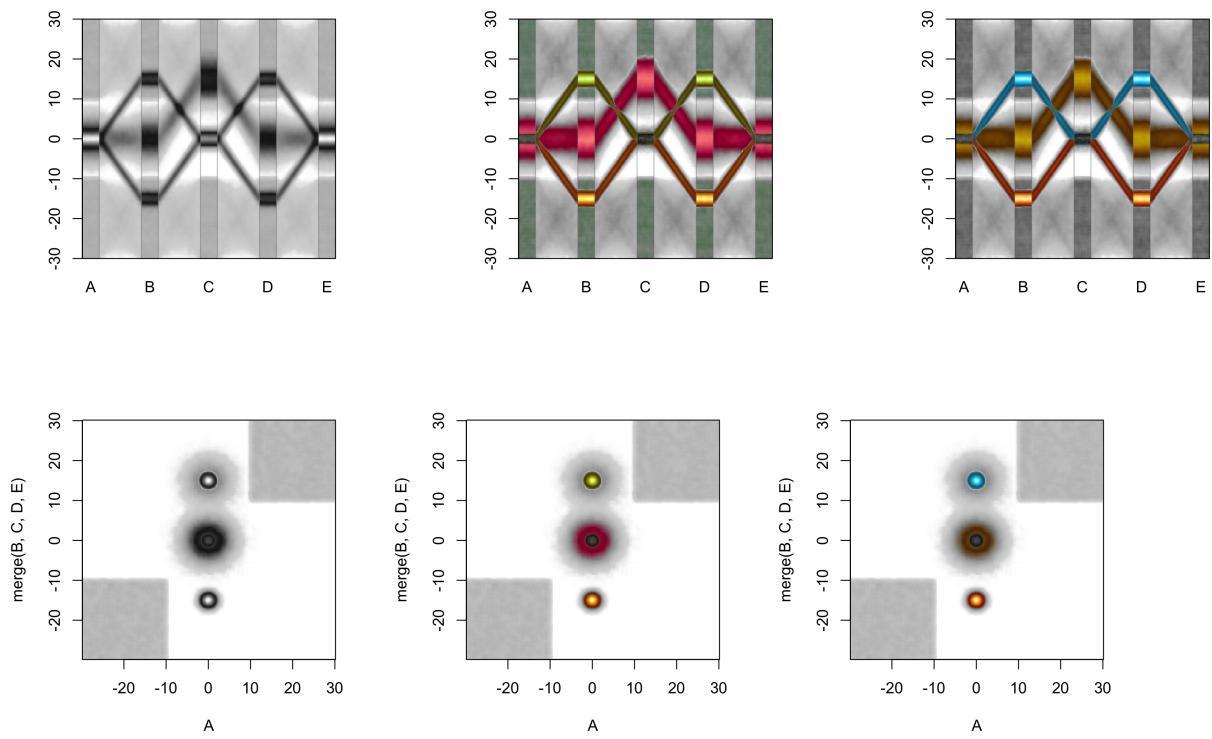
```
# Default value is stencil = "linear"
Variations(X)
# Alternative stencil values are "cosine" and "sigmoid"
# Unambiguous abbreviations can be used
Variations(X, stencil = "cos")
```

```
SideBySide(X, stencil = "lin", db = 20, colors = c(d = rgb(0, 0.5, 1)))
SideBySide(X, stencil = "cos", db = 20, colors = c(d = rgb(0, 0.5, 1)))
```

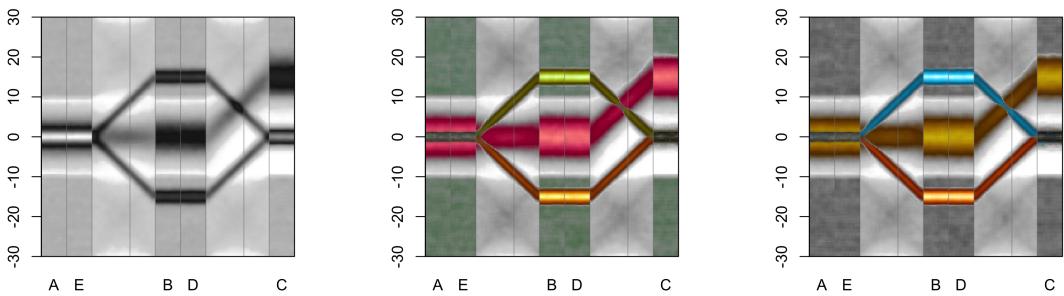


Populations

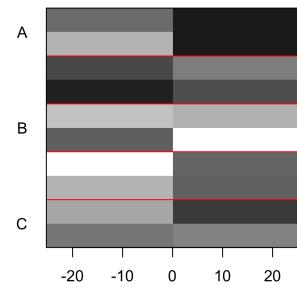
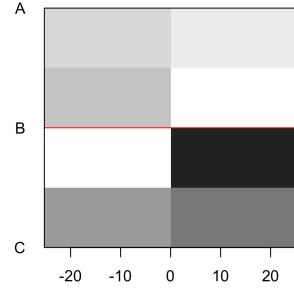
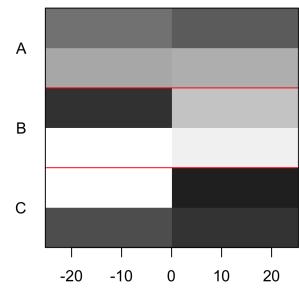
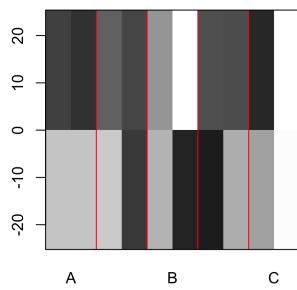
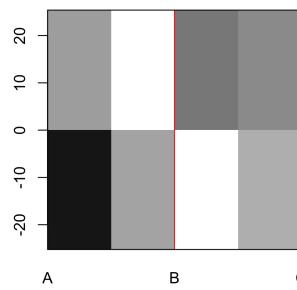
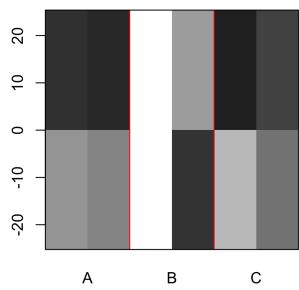
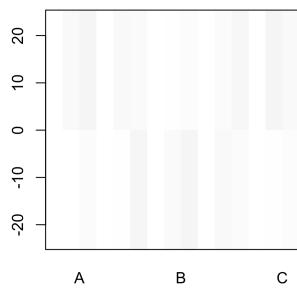
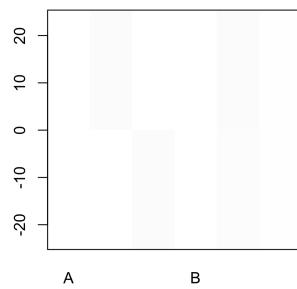
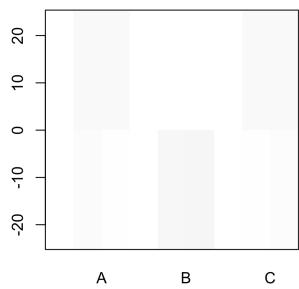
```
SideBySide(X, rng, db = 20)
SideBySide(X, rng, db = 20, pops = grp)
SideBySide(X, rng, db = 20, pops = grp, colors = clr)
ScatterMaps(X, x = "A", y = colnames(X)[-1], pops = grp, colors = "grey")
ScatterMaps(X, x = "A", y = colnames(X)[-1], pops = grp)
ScatterMaps(X, x = "A", y = colnames(X)[-1], pops = grp, colors = clr)
```



```
SideBySide(X[, c(1, 5, 2, 4, 3)], rng, db = 20, skip = c(1, 3))
SideBySide(X[, c(1, 5, 2, 4, 3)], rng, db = 20, skip = c(1, 3), pops = grp)
SideBySide(X[, c(1, 5, 2, 4, 3)], rng, db = 20, skip = c(1, 3), pops = grp, colors = clr)
```



Debug



Movie