

PlotSvalbard - User Manual

Mikko Vihtakari

2017-12-11

PlotSvalbard package provides functions to plot research data from Svalbard on high resolution maps in R. The package is based on ggplot2 and the functions can be expanded using ggplot syntax. The package contains also maps from other places in the Arctic, including polar stereographic maps of the Arctic.

Note that the quality of graphics in this document is restricted due to vignette requirements for R. The quality of maps printed on your graphics window should be much higher than here. The maps in PlotSvalbard are designed to be printed on `pdf()` device. If you need the maps for other devices, see the Graphical parameters section.

Basemap

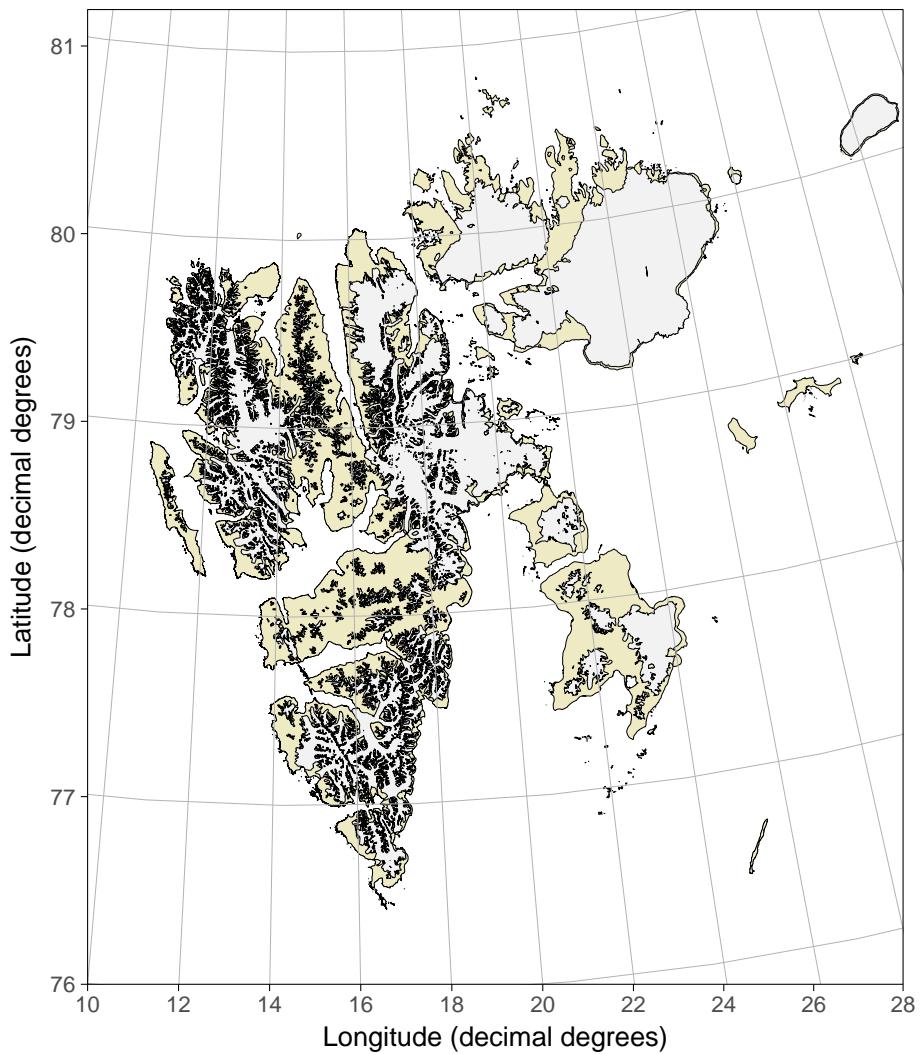
The `basemap` function is the generic plotting command in PlotSvalbard, and is analogous to empty `ggplot()` call. The `basemap()` call plots a map that is specified by the `type` argument. Data that contains geographic information can be plotted on these maps using the ggplot2 layers separated by the `+` operator.

Map types

Map types are specified by the `type` argument in `basemap()` function. The list below shows all currently implemented map types.

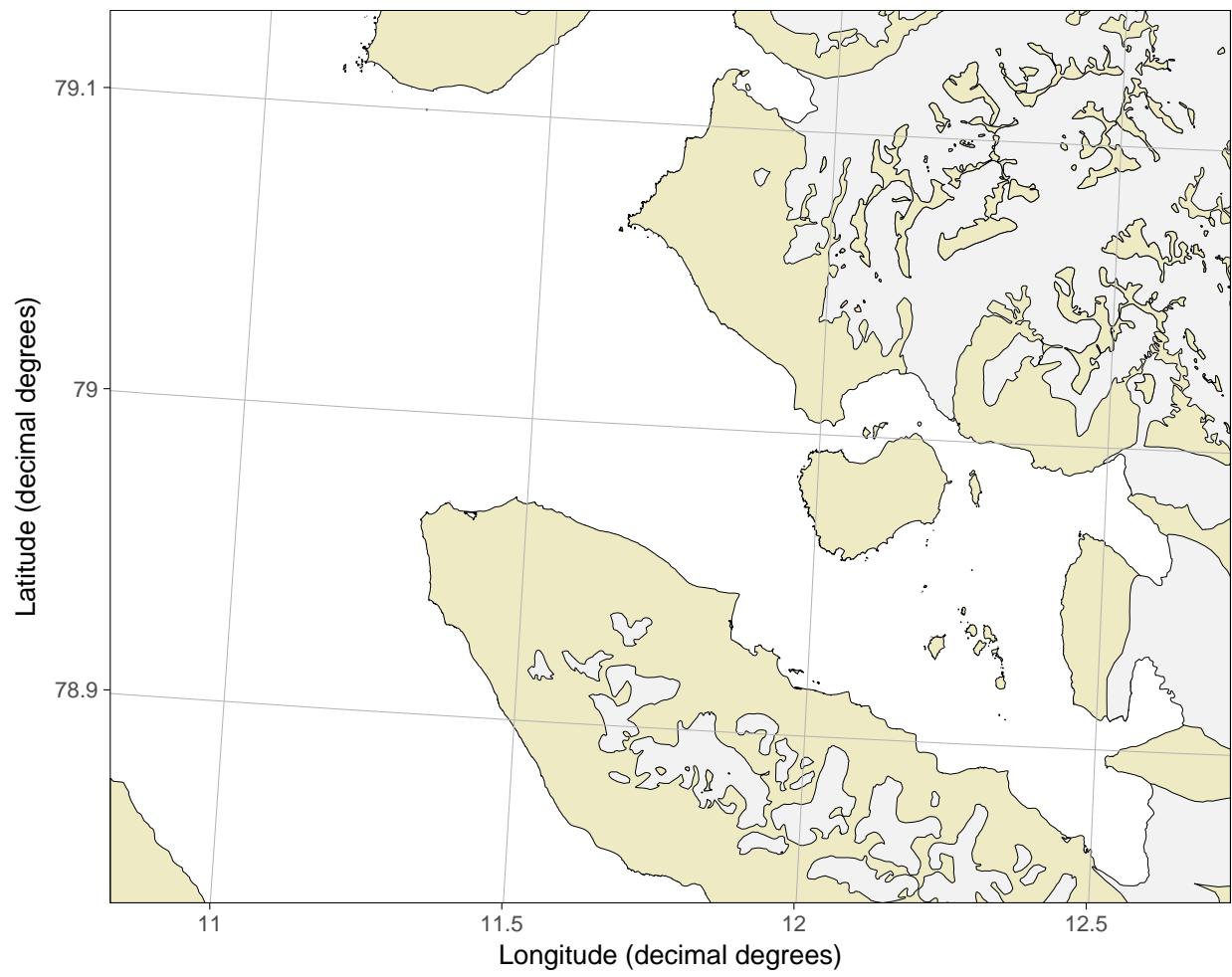
Svalbard

```
library(PlotSvalbard)
basemap("svalbard")
```

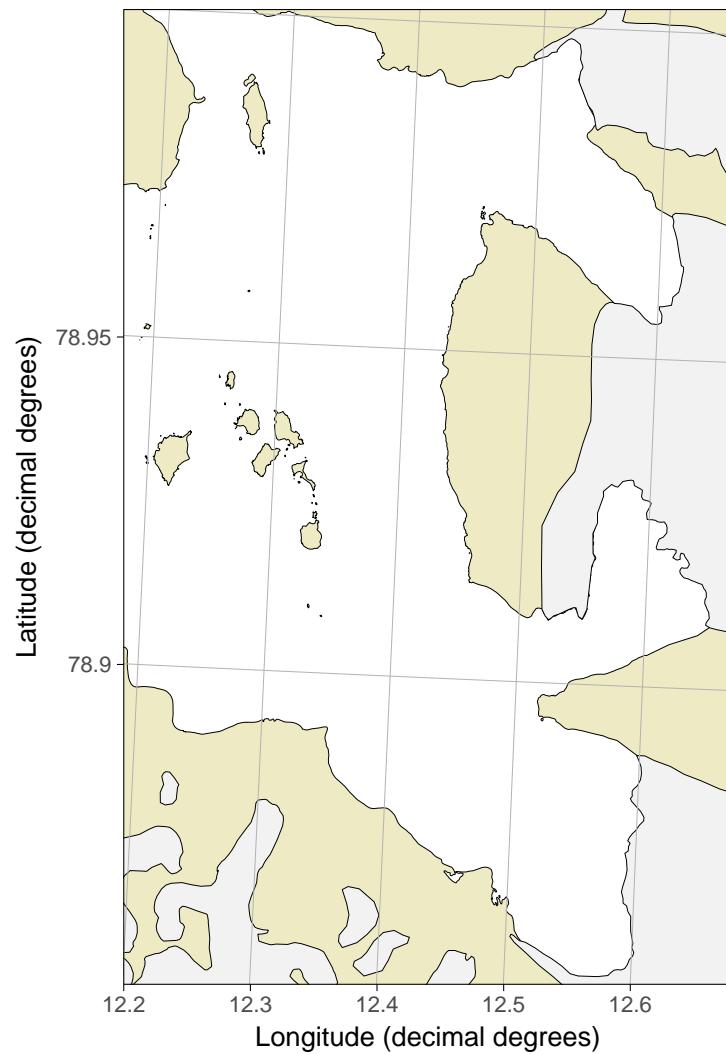


Kongsfjorden

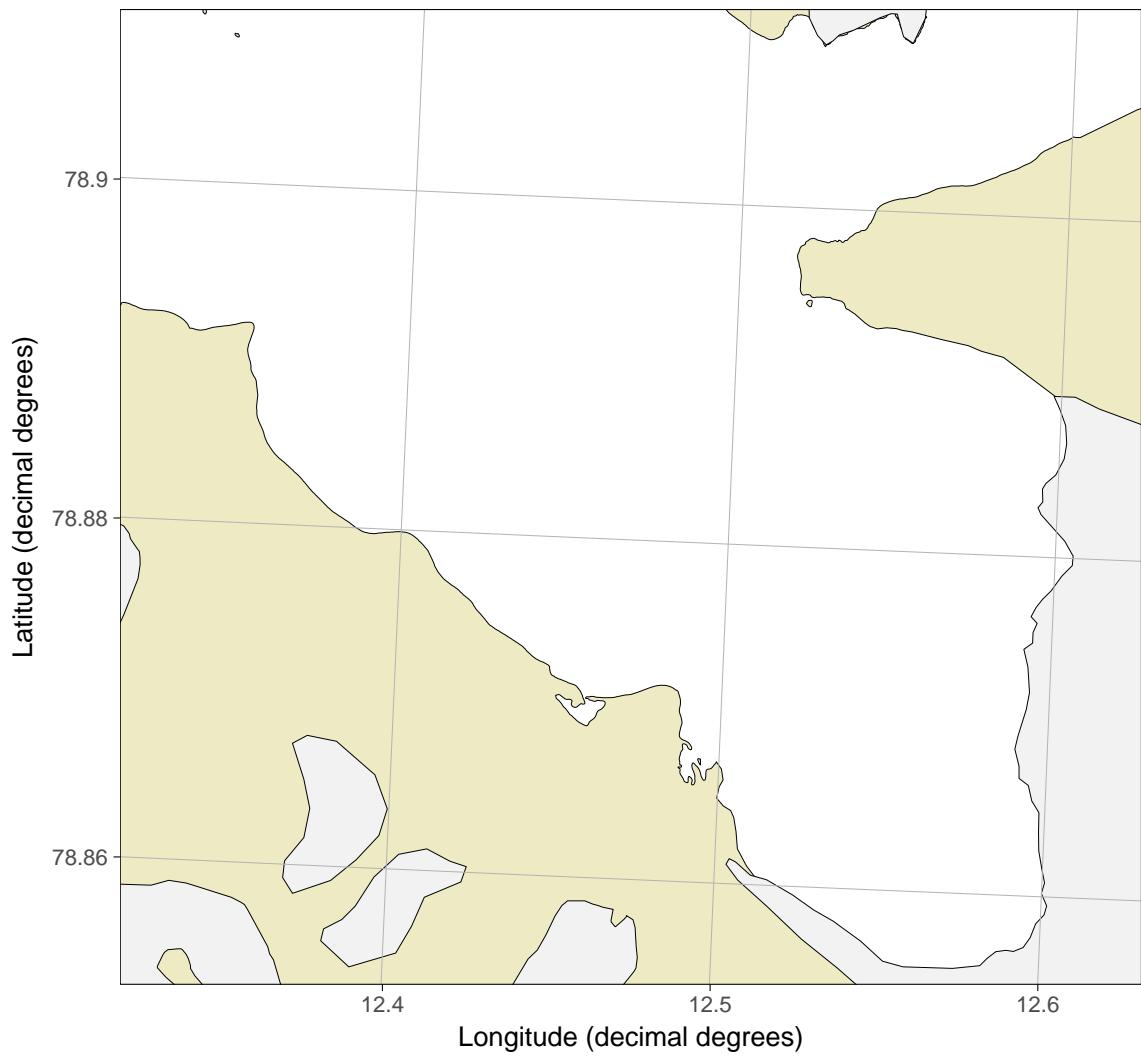
```
basemap("kongsfjorden") # or just basemap()
```



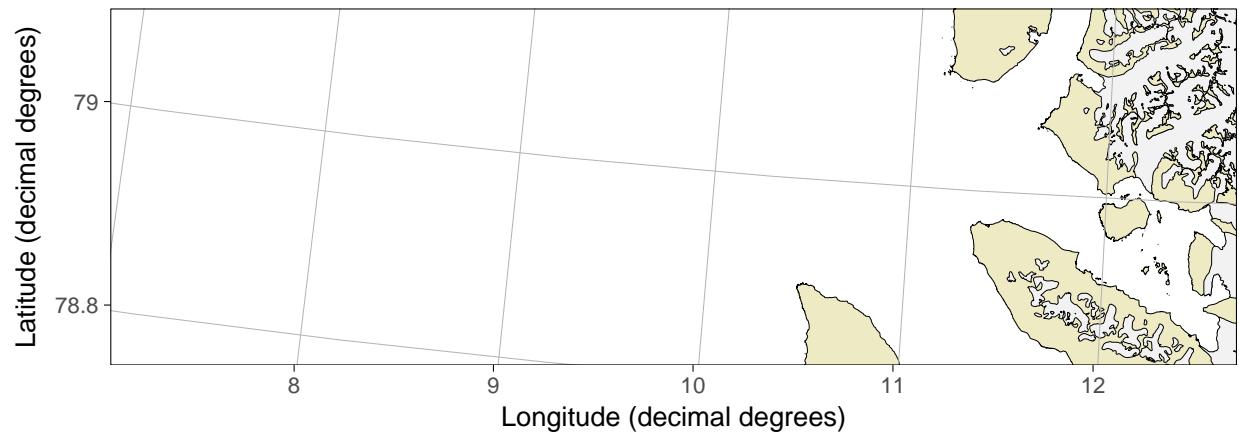
```
basemap("kongsfjordbotn")
```



```
basemap("kronebreen")
```

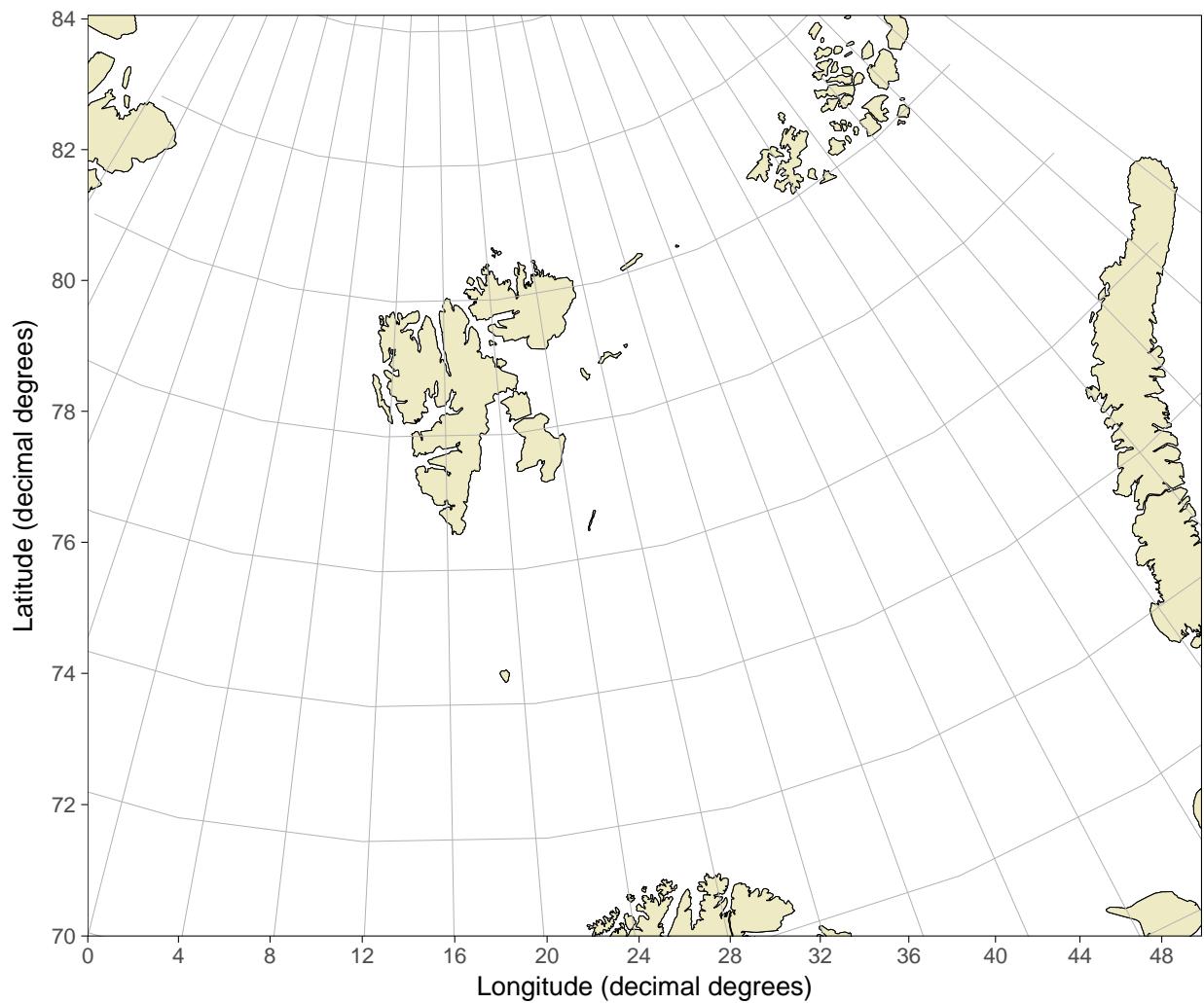


```
basemap("mosj")
```



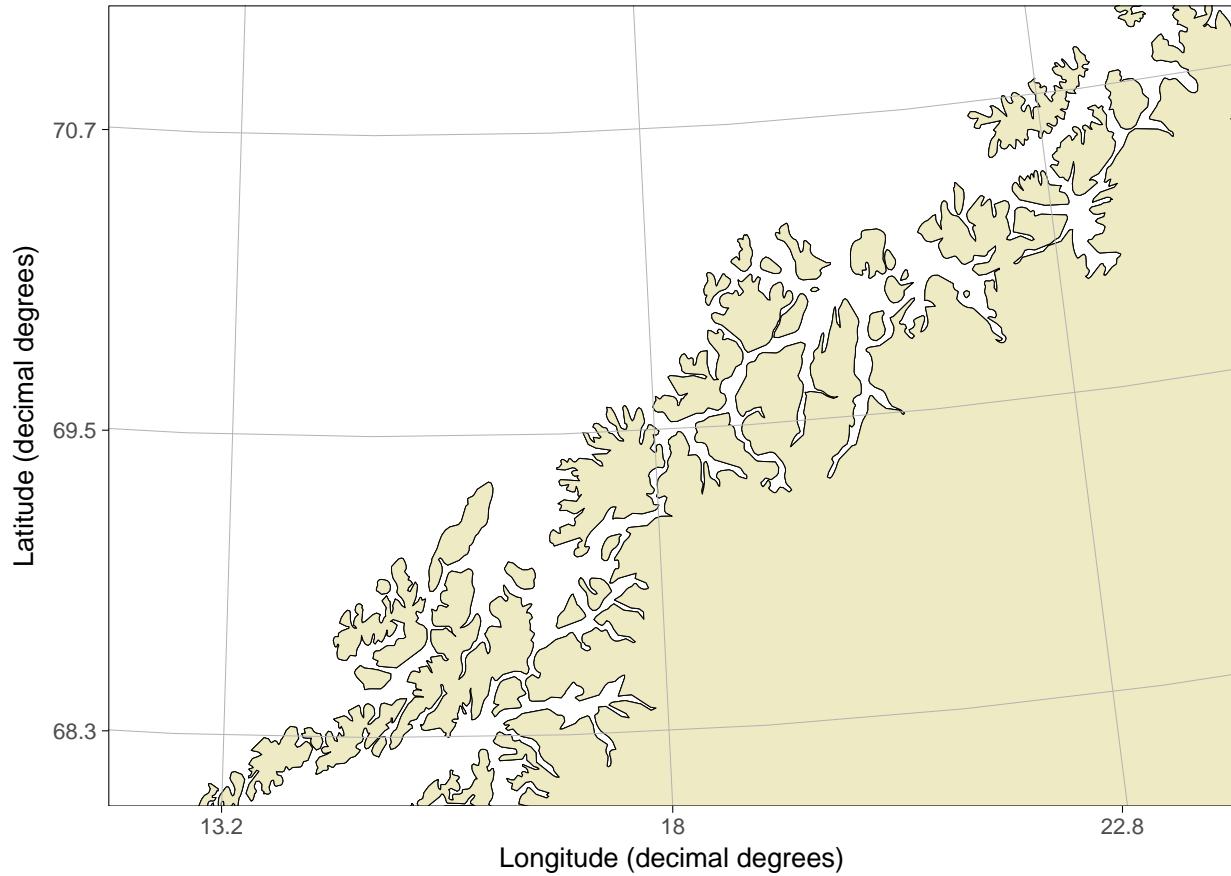
Barents Sea

```
basemap("barentssea")
```



Barents Sea map also prints mainland Norway, but the projection is not optimal, and the resolution is quite low.

```
basemap("barentssea", limits = c(12, 24, 68, 71))
```



Pan-Arctic

Not implemented yet, but the feature is planned.

Limiting (zooming) the maps

Any `basemap` can be limited (or zoomed in) using the `limits` argument. The `limits` argument has to be a numeric vector specifying the minimum longitude, maximum longitude, minimum latitude and maximum latitude of the limiting area in decimal degrees. The order and length (4) of the vector is rigid.

Note that some map types are already limited, so if you are looking for a map of Kongsfjorden, using `basemap("kongsfjorden")` might be just what you need.

Adding data to basemaps

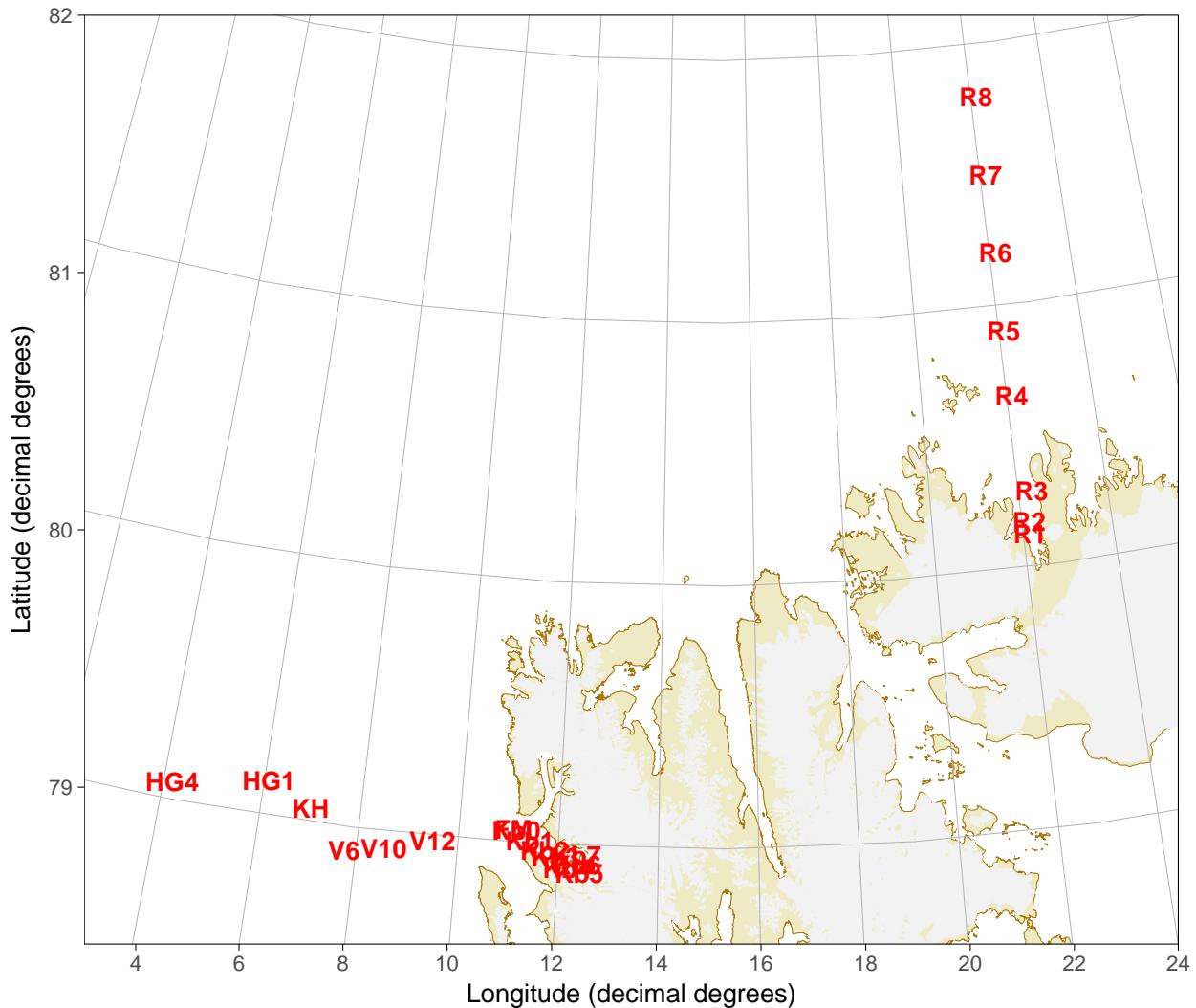
Data can be added to basemaps using the `+` operator and layers for `ggplot2`. Below you find some examples on how to add your research data on basemaps.

Adding station labels (text)

Text can be added to basemaps using the `geom_text()` function:

```
data("npi_stations")

basemap("svalbard", limits = c(3,24,78.5,82), round.lat = 1, round.lon = 2,
        border.col.land = "#a9750d", border.col.glacier = "grey95") +
  geom_text(data = npi_stations, aes(x = lon.utm, y = lat.utm,
    label = Station), color = "red", fontface = 2)
```



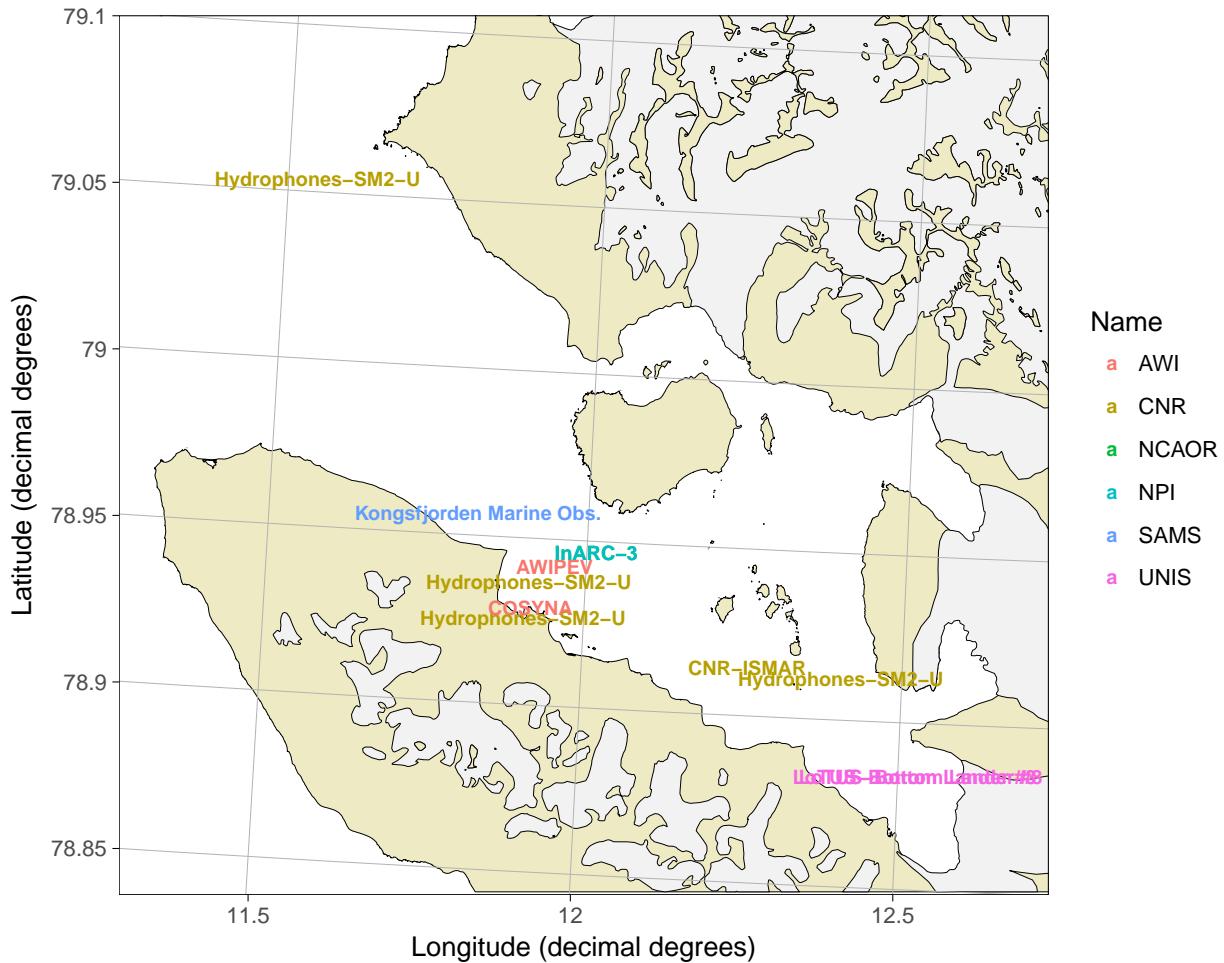
Note the issue with y-axis labels. This is because UTM coordinates make the bounding box of maps “tilted” compared to decimal degrees. In the current version of PlotSvalbard, the labels are bound to decimal degrees, but finding a solution to this should be relatively easy. This error will be corrected in the future versions of the package, so update your PlotSvalbard before you start making maps.

Text size can be mapped to variables using the standard ggplot2 syntax:

```
data("kongsfjord_mooring")
```

```
basemap("kongsfjorden", limits = c(11.3, 12.69, 78.85, 79.1), round.lat = 0.05, round.lon = 0.5) +
```

```
geom_text(data = kongsfjord_moorings, aes(x = lon.utm, y = lat.utm,
label = Mooring.name, color = Name), fontface = 2,
size = 25.4/72.27*8) # font size = 8, see Graphical parameters
```



Adding piecharts

The `scatterpie` package allows relatively easy plotting of piecharts on maps. Extensions for ggplot2 work together with PlotSvalbard.

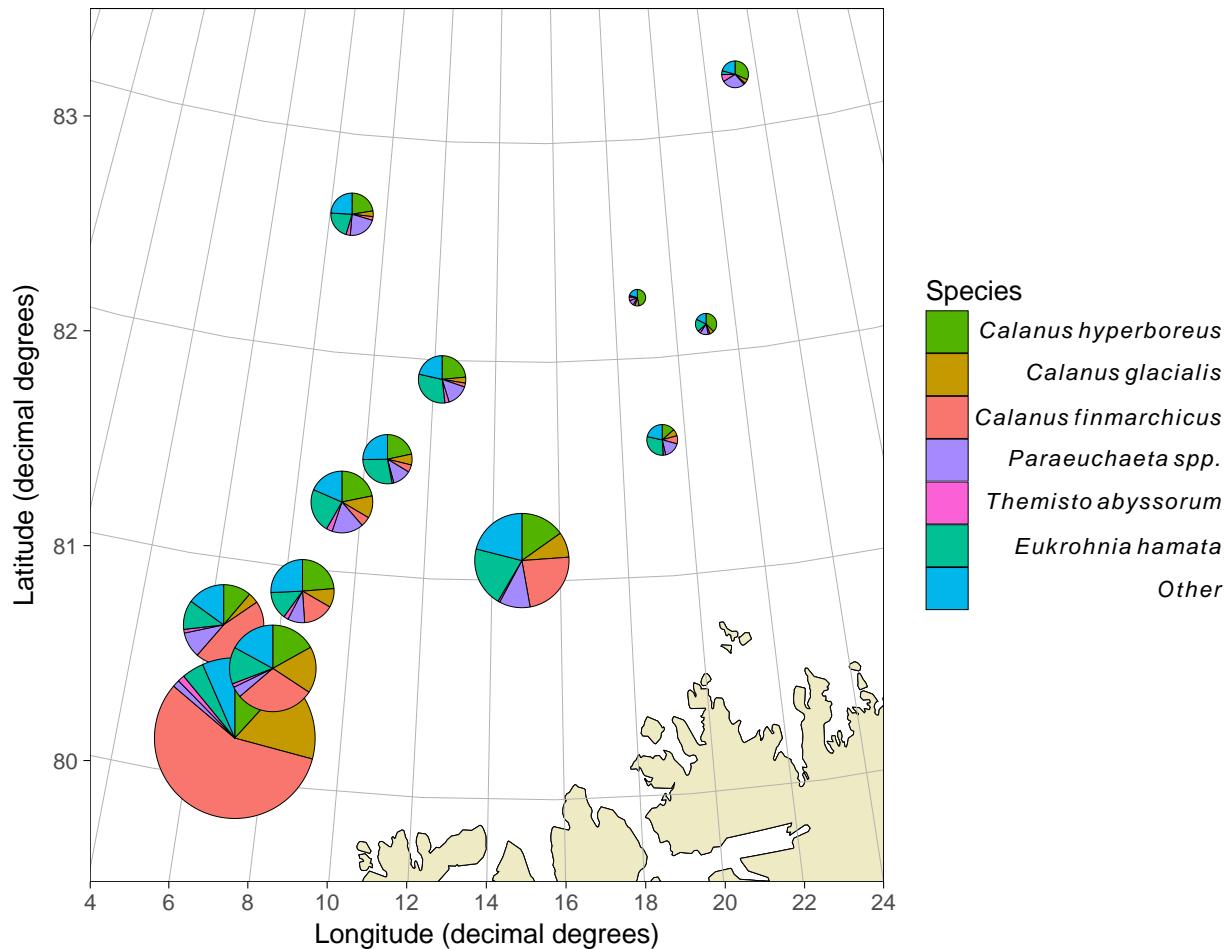
```
data(zooplankton)

x <- cbind(transform_coord(zooplankton, lon = "Longitude", lat = "Latitude"), zooplankton)
#> projection transformed from +proj=longlat +datum=WGS84 to +init=epsg:32633

species <- colnames(x)[!colnames(x) %in% c("lon.utm", "lat.utm", "ID",
"Longitude", "Latitude", "Total")]

library(scatterpie)
```

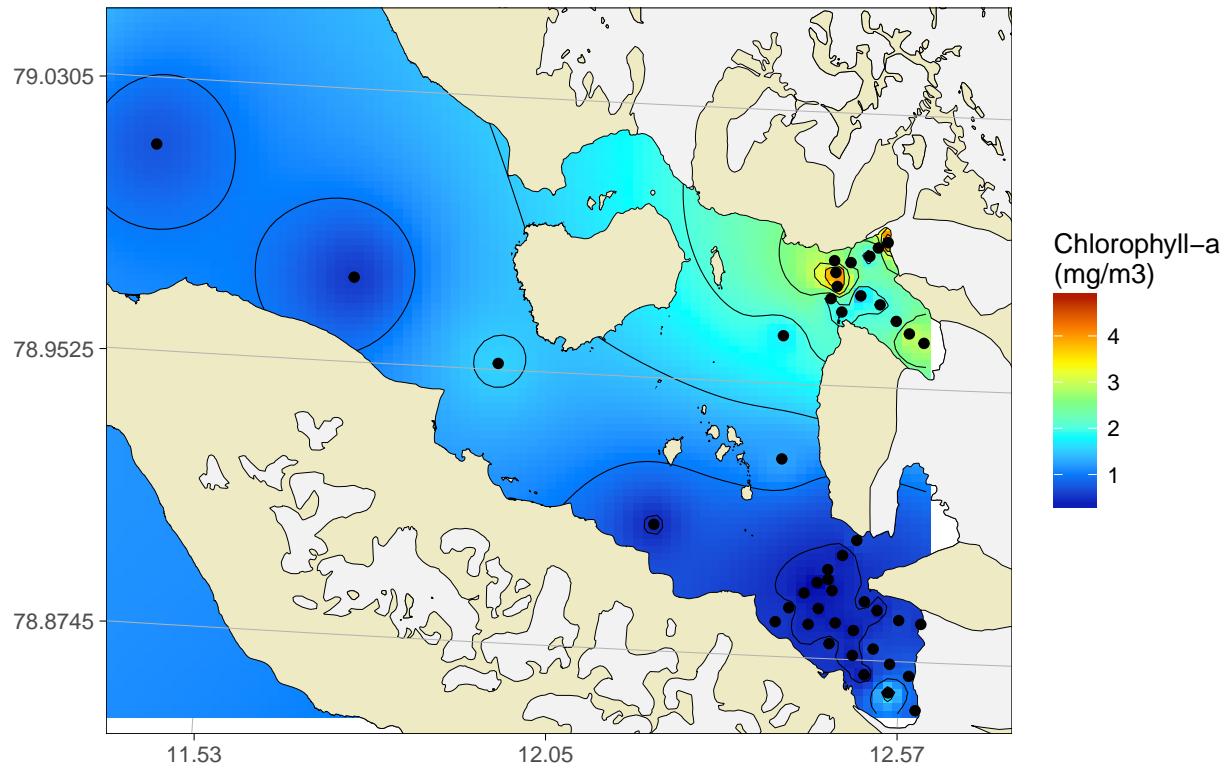
```
basemap("barentssea", limits = c(4, 24, 79.5, 83.5), round.lon = 2, round.lat = 1) +
  geom_scatterpie(aes(x = lon.utm, y = lat.utm, group = ID, r = 100*Total),
  data = x, cols = species, size = 0.1) + scale_fill_discrete(name = "Species",
  breaks = species, labels = parse(text = paste0("italic(`", sub("*\\\. ", "~", species), `)")))
```



Adding interpolated 2D surfaces

PlotSvalbard uses the `krige` function from the `gstat` package to produce interpolated 2D plots:

```
data("chlorophyll")
x <- interpolate(chlorophyll, Subset = "From <= 10", value = "Chla") ## Interpolate
## [inverse distance weighted interpolation]
plot(x, type = "kongsfjorden", limits = c(11.4,12.7,78.855,79.05),
  legend.label = "Chlorophyll-a\n(mg/m3)")
```



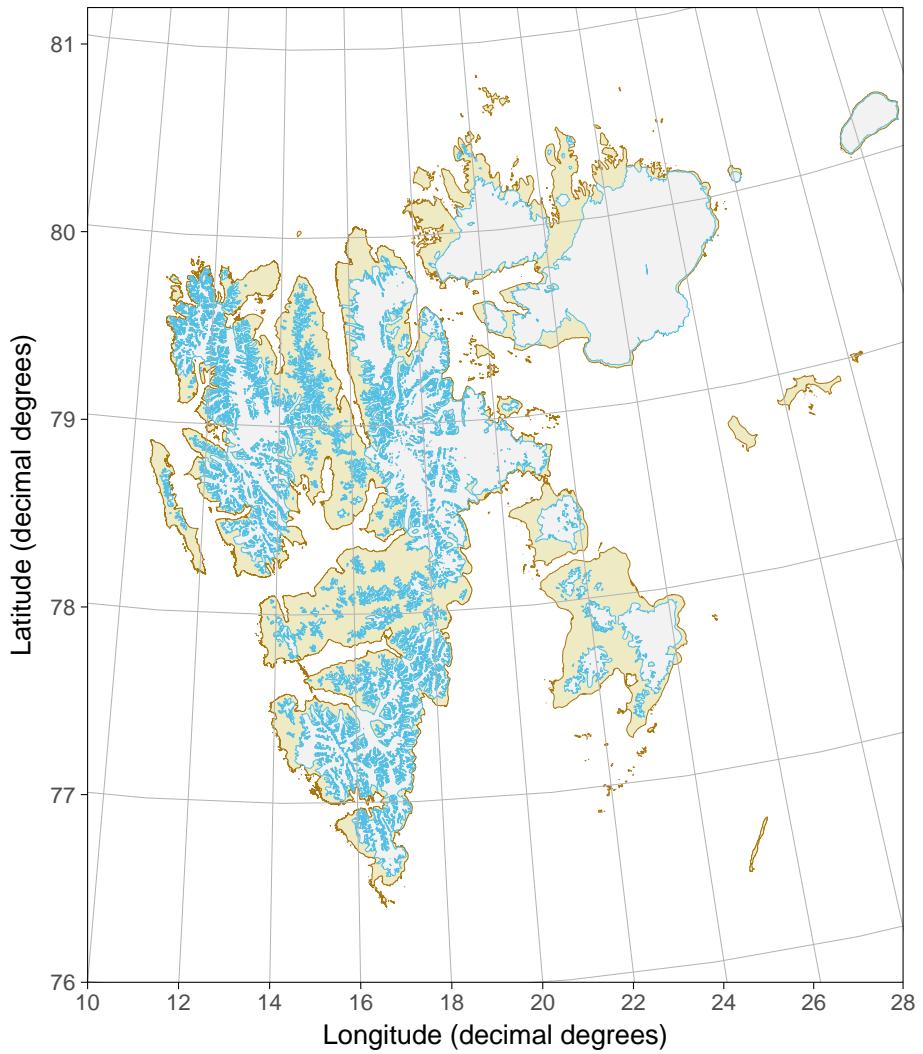
Bathymetry

Not implemented yet, but the feature is planned.

Graphical parameters

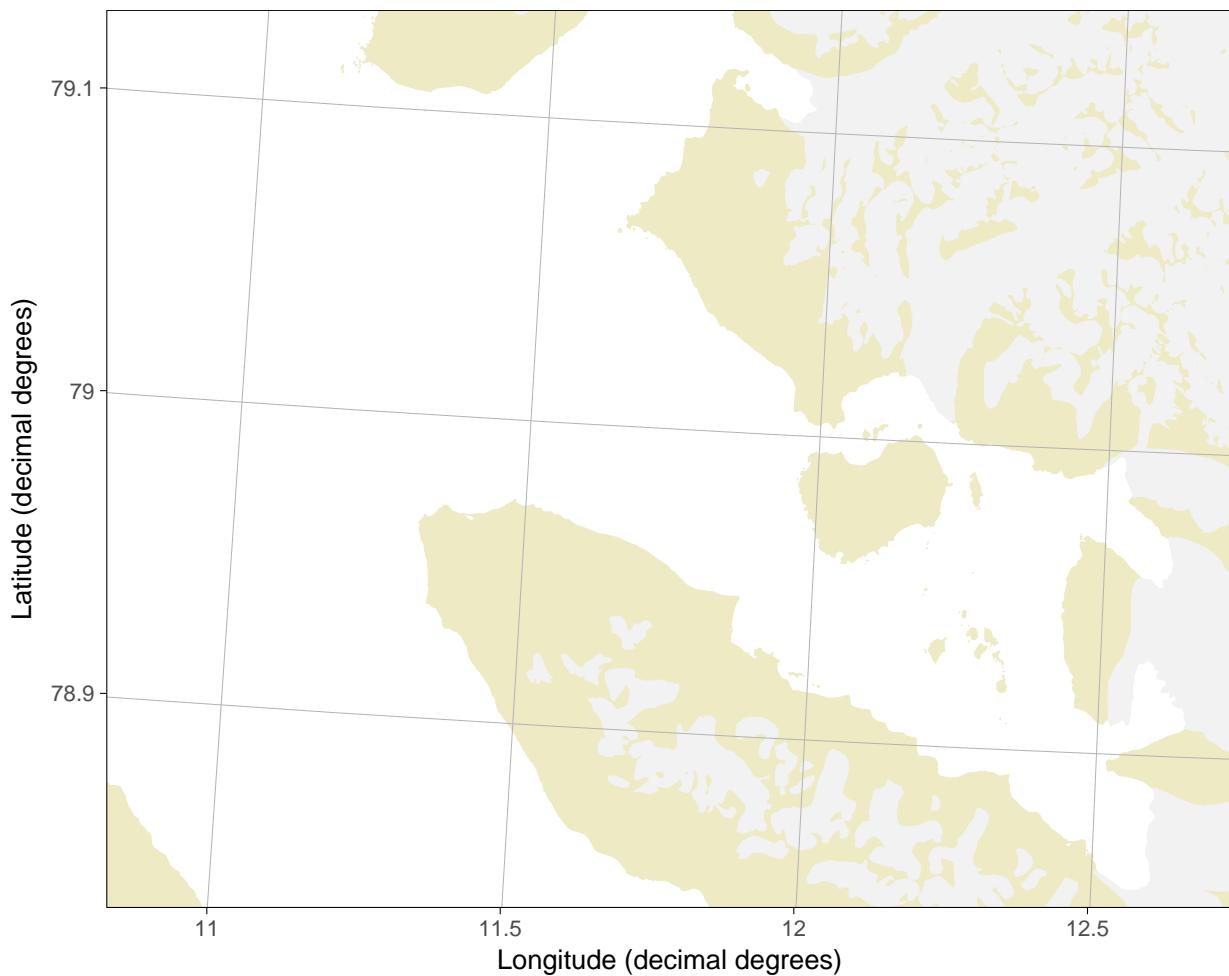
The line widths and general looks of the basemaps are optimized for a half page sized pdf figure in scientific journals (Springer figure dimensions were used to develop the function). The line widths may not look good when printed on other devices. You can modify the line widths and colors using `size.*` and `border.col.*` arguments.

```
basemap("svalbard", size.land = 0.01, size.glacier = 0.05, size.grid = 0.05,
       border.col.glacier = "#52bfe4", border.col.land = "#a9750d")
```



Approach to remove borders of land and glacier shapes:

```
basemap("kongsfjorden", border.col.glacier = "grey95", border.col.land = "#eeeac4")
```



The line width of `ggplot2` is 2.13 wider than the line widths measured as points (pt). This means that if you want a certain line width, multiply the desired line width by $1/2.13$ inside `size.*` arguments. Similar conversion factor for font size is $1/2.845276$.

Performance

The `basemap("svalbard")` is currently fairly slow (approx. 25 seconds) due to the large size of the shapefiles that are used to generate the map. Unfortunately this slowness also applies for zoomed in maps, because I have not found a way to clip the glacier shapefile. The `clip_shapefile` function was developed for this purpose, and worked well for land shapes. However, clipping a glacier shapefile takes longer than directly plotting it. It is possible that there is a way to speed up the `basemap("svalbard")`, but in the current version plotting Svalbard is just slow.

If you are looking for optimal `limits` for your data, you can use the `basemap("barentssea")` option to find these limits and replace "`barentssea`" with "`svalbard`" once you are done:

```
system.time(basemap("barentssea", limits = c(c(19.5,23.5,80,81.7))))
#>   user  system elapsed
#> 0.19    0.00    0.19
system.time(basemap("svalbard", limits = c(c(19.5,23.5,80,81.7))))
```

```
#> Regions defined for each Polygons
#>   user  system elapsed
#> 22.31    0.00   22.48
```

Citing PlotSvalbard

The `citation` function tells how to cite PlotSvalbard package

```
citation("PlotSvalbard")
#>
#> To cite package 'PlotSvalbard' in publications use:
#>
#> Mikko Vihtakari (2017). PlotSvalbard: PlotSvalbard – plot
#> research data from Svalbard on maps. R package version 0.1.8.
#> https://github.com/MikkoVihtakari/PlotSvalbard
#>
#> A BibTeX entry for LaTeX users is
#>
#> @Manual{,
#>   title = {PlotSvalbard: PlotSvalbard – plot research data from Svalbard on maps},
#>   author = {Mikko Vihtakari},
#>   year = {2017},
#>   note = {R package version 0.1.8},
#>   url = {https://github.com/MikkoVihtakari/PlotSvalbard},
#> }
```

However, please note that the maps generated by this package should be cited to their original source. See “Source” in `?basemap` for more details. The example data included in the package are property of the Norwegian Polar Institute and should not be used in other instances. I.e. these data are unpublished at the moment.

PlotSvalbard depends on many packages (see `DESCRIPTION`) for dependencies. If you use these packages in your publications, please cite the respective packages.