

# PBSmapx: User's Guide

Rowan Haigh and Jon T. Schnute

April 14, 2014

## 1 What is PBSmapx?

The package **PBSmapx**, which is short-hand notation for *PBS Map Explore*, provides a convenient Graphical User Interface (GUI) tool for interacting with **PBSmapping**. The primary mapping function `createMap` presents a menu system for plotting topographical files as well as output from gridding and summary routines for geo-referenced data such as catch, effort, and catch per unit effort (CPUE). The underlying R code is somewhat convoluted; however, the user can largely ignore the code and simply manipulate the entries in the GUI menu. Though designed for users at the Pacific Biological Station (PBS), the GUI has proven useful for users at other locales.

**PBSmapx** depends on the R package **PBStools**, which in turn depends on four other R packages: **PBSmapping** (Schnute *et al.*, 2004; Boers *et al.*, 2004), **PBSmodelling** (Schnute *et al.*, 2006), **PBSdata**, and **RODBC**. We use **PBSmodelling** to implement the GUI. Originally, **PBSmapx** evolved over time (2007–2012) within the R package **PBSfishery** (found on Google's Project Hosting site: <http://code.google.com/p/pbs-fishery/>), along with useful datasets (regional boundaries, key codes, example data) and handy utility functions. In April 2012, we decided to split **PBSfishery** into three separate libraries – **PBSmapx**, **PBSdata**, and **PBStools** – for distribution on CRAN.

## 2 What is PBS?

The Pacific Biological Station is the oldest fisheries research centre on Canada's Pacific coast and forms part of a network of nine major scientific facilities operated by Fisheries and Oceans Canada. Located in Nanaimo, British Columbia, the Station is home to scientists, technicians, support staff and ships' crews whose common interests are the coastal waters of British Columbia, the Northeast Pacific Ocean, the Western Arctic and navigable waters east to the Manitoba, Saskatchewan border.

PBS was established in 1908 and is the principal centre for fisheries research on the West Coast. There are some 22 structures on the site including a four-story office/wet lab building, specialty storage structures for hazardous chemicals and salt water pumping

facilities. PBS maintains a number of workshops for research support. There is a wharf used for loading, unloading, and berthage of research vessels, as well as a small boat dock for inshore research boats. PBS also maintains a library and meeting facilities. Aquatic facilities, primarily used by Aquaculture Science, include ambient temperature and heated salt water and fresh water.

Research at PBS responds to stock assessment, aquaculture, marine environment, habitat, ocean science, and fish productivity priorities. Some fisheries management activities are also conducted here.



**Figure 1.** Pacific Biological Station (PBS), Nanaimo BC

For more information, see:

<http://www.pac.dfo-mpo.gc.ca/science/facilities-installations/pbs-sbp/index-eng.htm>.

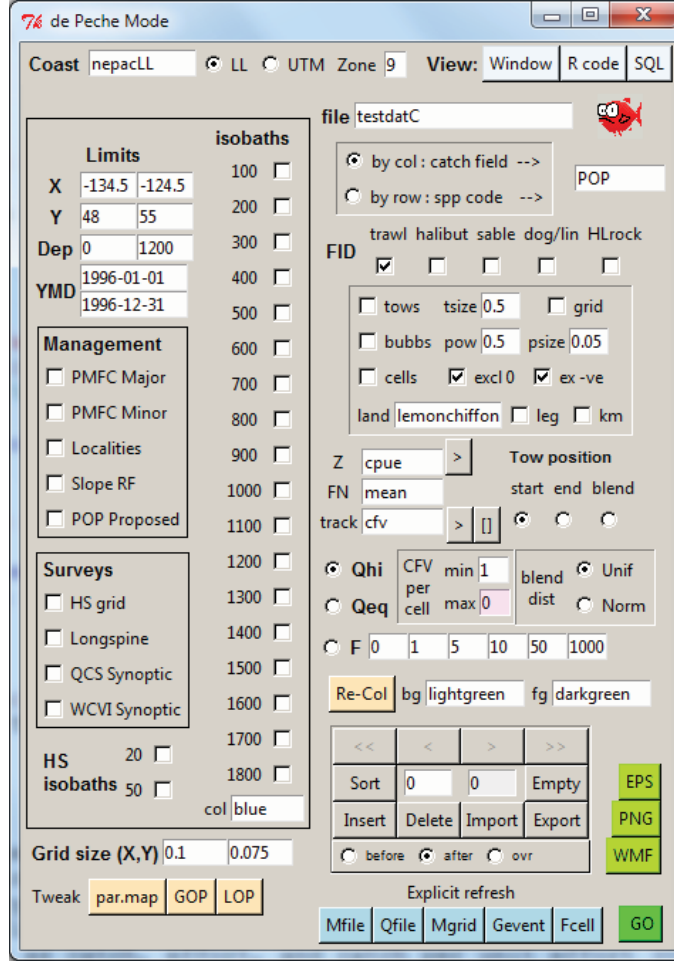
### 3 Mapping GUI

The mapping GUI (Figure 2), whimsically called ‘de Pêche Mode’ after the 1980s New Wave band from England called Depeche Mode<sup>1</sup>, is created by a call to `createMap` from the R console. It defaults to spatial limits (X,Y) that display the BC coast using the `PBSmapping` data object `nepacLL`, which the user can change to another coastline object. The user can also switch between two projections: LL = geographic (degrees longitude and latitude) and UTM = Universal Transverse Mercator (km).

In the control box on the left, the spatial (X = horizontal, Y = vertical, Dep = Depth) and temporal (YMD = Year-Month-Day) limits can be adjusted as needed. The remaining items in the control box on the left refer to BC geographic data objects that delimit management boundaries (not necessarily official), survey regions, and isobaths from 100 to 1800 m at 100 m intervals. Below the control box, a user can change the dimensions of the grid cells Should s/he decide to locate and summarize events in a grid that overlays

---

<sup>1</sup> Excerpt from Wikipedia’s article on Depeche Mode ([http://en.wikipedia.org/wiki/Depeche\\_mode](http://en.wikipedia.org/wiki/Depeche_mode)): When explaining the choice for the new name (taken from a French fashion magazine, *Dépêche mode*) Martin Gore (keyboards, guitar, vocals, chief songwriter after 1981) has said, *It means hurried fashion or fashion dispatch. I like the sound of that.* Rowan agreed, and so de Pêche Mode was born as a GUI that can be loosely translated as *Fishing Method*, *Fishery Dispatch*, or even more facetiously, *Fishery Fashion*.



**Figure 2.** de Pêche Mode GUI generated by `createMap()`. Specify coast line object at top and choose spatial and temporal limits on left. Also in box on left, user can choose various topographical features. Right side of GUI controls the data input to be mapped either as tow points, proportional bubbles, or summary statistics in grid cells.

the display region. These events come from a geo-referenced data object specified by the entry box File.

On the right, various controls coordinate the display of geo-referenced data like catch, effort, and catch per unit effort (CPUE). In the development of this mapping tool, we found that users typically organise their data in one of two ways or shapes (Bolker, 2008). The first is to allow one record per event (e.g., trawl tow) detailing the specifics of that event (e.g., effort, depth, etc.) with additional columns labelled by species that contain the catch of that species in the event. The species catches are said to be ‘arranged by column’, and the table structure approximates a crosstab query in Access or a pivot table in Excel. (Bolker refers to this format as ‘wide’.) The second type of data structure has only one column for species catch, an additional column for species code, and event information repeated numerous times depending on the number of species caught in a particular event. In this case we say that the species catches are ‘arranged by row’, and the table structure has fewer fields but many more records containing a lot of redundant

information. (Bolker refers to this format as ‘long’ or ‘indexed’). We encourage users to organise their data so that species catches are arranged by column (wide format), but the option exists for both structures. R provides a few tools for conversion between the two formats (`stack` and `unstack` in the `utils` package; `reshape` in the `stats` package; `melt`, `cast`, and `recast` in the `reshape` package). Additionally, the package `PBStools` offers a `crossTab` function that uses the `reshape` functions `melt.data.frame` and `cast` to reproduce the ‘wide’ format similar to a Microsoft SQL crosstab query.

### DFO:

A basic data object for a single species can be obtained using the `PBStools` function `getData` and the SQL code file `pht_map_density.sql`:

```
getData("pht_map_density.sql", "PacHarvest", strSpp="396",  
        path=.getSpath(pkg="PBStools"))
```

which creates a global data object, in this case for Pacific Ocean Perch, called `PBSdat` that the user can rename to whatever s/he wishes. Note the hidden function `.getSpath` from the package `PBStools`. Essentially, this function gets the path to the `PBSmapx` directory called `sql`. For example, this might evaluate to:

`C:/Program Files/R/R-3.0.3/library/PBSmapx/sql` on a user’s system. We maintain a few SQL code files here but a much larger collection exists in the `sql` directory of `PBStools`. These can be treated as templates by users at other institutions even though they were designed specifically for users at PBS.

The FID (fishery ID) options include trawl (`trawl`), halibut (`halibut`), sablefish (`sable`), dogfish/lingcod (`dog/lin`), and hook and line rockfish (`HLrock`).

Once a data object has been selected, the user chooses various fields that contains values to display on the map (where land colour is controlled by the entry box labelled `land`). The usual candidates for groundfish trawl data are longitude `X`, latitude `Y`, and some `Z` value, often `catch`, `effort`, or `cpue` (catch per unit effort can be calculated from `catch` and `effort` if absent from the dataset). Not surprisingly, all users have their preferences on what to call fishery data fields. The code expects certain fields to be available and attempts to try obvious candidate substitutions if the required fields do not exist (Table 1). For the `Z`-value alone, the symbol `>` to the right of the `Z` entry box provides a pop-up menu choice of data fields should the user wish to peruse and choose an available field.

The user has three choices on how to display fishing event data. The first is to plot the fishing event locations as points by checking the box labelled `tows`. The point size can be controlled by changing the value in the entry box labelled `tsize`. Normally, tow location is most consistently represented by the start position (`X,Y`); however, the user may wish to use either the end position (`X2,Y2`) or a vectorized blend between (`X,Y`) and (`X2,Y2`). The latter comes in handy for gridding analyses when parsing `Z` values out over a simplified tow track.

Second, `Z`-values can be rendered as bubbles (`bubbs`) proportional to their

**Table 1.** Candidate substitutions for various GUI inputs.

Default	Substitutes fields (in preferential order)
X	longitude, long, lon, x
Y	latitude, lat, y
X2	X, longitude, long, lon, x
Y2	Y, latitude, lat, y
fdep	depth, depth1, fishdepth, Z, z
cfv	CFV, vessel, boat, ship
spp	species, sp, hart, code
catch	cat, total, tcat, landed, land, kept, discarded, discards, discard, C
effort	eff, duration, time, hours, minutes, E
cpue	CPUE, U

magnitude. The function `symbols` takes radii of circles to construct bubbles, so a square root transformation of Z (`pow=0.5` in the GUI) most closely represents Z proportional to the area of the bubble. The maximum bubble size can be adjusted by changing the value in the entry box labelled `psize`.

Third, Z-values can be aggregated into pre-defined grid cells (`cells`), and a summary function (FN) like `mean` can be applied to each set of values in all cells that contain at least one event. (This summary function could be a highly complicated, user-defined algorithm, as long as it exists in the global environment.) With the gridding option, a user sometimes likes to track an associated variable not used in the analysis (`track`). For instance, if we grid the mean CPUE and only display those cells that contain at least 10 distinct commercial fishing vessels (CFV per cell), which CFVs have been retained in the analysis? The button marked [ ] will display a pop-up box of unique CFVs. There are also check boxes to (i) exclude records where the values of Z are either 0 (`excl 0`) and/or negative (`ex -ve`), (ii) add a legend (`leg`) detailing the grid categories and map title, and (iii) add area calculation summaries (`km`) to the legends. At any time, the user can see the entire grid used for analysis by checking the box labelled `grid`, though this display will not often be desired.

The gridding option also needs some way to categorize the summary data so that the cells can be colour-coded accordingly. The default is to use quantiles that emphasize hotspots (`Qhi`). There is also an option the chooses evenly-spaced quantiles (`Qeq`). These automatic options are handy to get a feel for the scale of the summarized Z-values. Once the scale is known, the user has the option to specify six fixed break points (`F`) to form five colour-coded categories. As biological data seem to exhibit skewed distributions (e.g., log-normal, gamma, pareto), the user could choose fixed values that follow some logarithmic pattern (e.g., 0, 0.1, 1, 10, 100, 1000) to display the categorical groups.

The colour coding for points, bubbles, and grid cells can be controlled using the entry boxes marked `bg` (background) and `fg` (foreground). There is also a button marked `Re-Col` that lets the user re-colour the events of an existing plot if the chosen colours aren't

quite right. This can save time redrawing the map from scratch, especially if the user has chosen to perform a gridding exercise with a large number of fishing events. For points and bubbles, pressing the **Re-Col** button will change the interior of the shapes to the **bg** colour and the borders to the **fg** colour. For grid cells, the border colours remain fixed in light grey, and the colour gradient over the range of categories ranges from **bg** to **fg**.

To see a demo of `createMap`, cut and paste the command below into the R console:

```
createMap(hnam=paste(.getWpath(pkg="PBSmapx"),"mapHist.txt",sep="/"))
```

This function call taps into a demo history file. (For details on the **history** widget, see the **PBSmodelling** user's guide.) Press the **history** navigation button **>** to scroll forward through 10 GUI settings and their results (see Figures below). Plotting to the R graphics device is automatic for the demo. Once you're ready to use your own data and settings, most times you need only hit the green **GO** button (lower right corner). At any time, you can hit one of the figure output buttons marked **EPS** (encapsulated postscript), **PNG** (portable network graphics), or **WMF** (windows metafile) to create a figure file based on the GUI settings. Very occasionally, you may want to force explicit refresh steps (blue buttons: **Mfile** = make the master data object; **Qfile** = make the qualified data object; **Mgrid** = make the master grid object; **Gevent** = get the **EeventData** set; **Fcell** = find the events in the cells) if you are having trouble. Enquiring minds can see the complete code by pressing the button labelled **R code** located in the upper right corner of the GUI.

Down in the lower left of the GUI, are a few additional buttons labelled **parmap**, **GOP**, and **LOP**. The first button allows the user to exercise additional **par** control that is not available through the GUI. These details are presented in the next section. The button labelled **GOP** simply stands for 'Grid Over Plot' and can be used to force plotting of coloured grid cells over everything. This action is only temporary so that a user can visually check how many grid cells are being masked by land. This only happens when the original data have badly-specified geographic coordinates. The button labelled **LOP** (Land Over Plot) forces plotting of coloured land over everything. This might come in handy when the user is manually adding **PolySets** or other information to the plot and wishes to mask out the regions where land occurs.

A global list object called **PBSmap** is returned packed with results and/or intermediate data objects. These can be handy for debugging or further analysis and plotting. This global list object differs in name from the standard used in **PBStools**, where functions typically return an object called **PBStool**.

## 4 Additional parameter control

After a call to `createMap`, a list of additional **par** parameters is created (Table 2). These additional parameters can be controlled from the GUI by hitting the **par.map** button, located near the lower left corner. The button uses the function `dataentry` from the **utils** package, which opens a spread-sheet like editor. (Not all operating systems may be able to use this.) Alternatively, the user can control these parameters at any point



**Table 2.** Additional parameter controls.

par	Parameter description
plt	A vector of the form <code>c(x1,x2,y1,y2)</code> giving the coordinates of the plot region as fractions of the current figure region. <i>default:</i> <code>c(0.08,0.99,0.08,0.99)</code> .
mgp	The margin line (in <code>mex</code> units) for the axis title, axis labels and axis line. Note that <code>mgp[1]</code> affects the title whereas <code>mgp[2:3]</code> affect the axes. <i>default:</i> <code>c(2.75,0.5,0)</code> .
las	A numeric in <code>c(0,1,2,3)</code> specifying the style of axis labels: <code>las=0</code> labels always parallel to the axis ( <i>default</i> ); <code>las=1</code> labels always horizontal; <code>las=2</code> labels always perpendicular to the axis; <code>las=3</code> labels always vertical.
cex.axis	The magnification ( <i>default:</i> 1.2) to be used for axis annotation relative to the current setting of <code>cex</code> .
cex.lab	The magnification ( <i>default:</i> 1.75) to be used for x and y labels relative to the current setting of <code>cex</code> .
cex.txt	Not technically a <code>par</code> parameter but it denotes the magnification ( <i>default:</i> 0.9) of the upper right text relative to the current setting of <code>cex</code> .
cex.leg	Not technically a <code>par</code> parameter but it denotes the magnification ( <i>default:</i> 0.9) of the lower left legend relative to the current setting of <code>cex</code> .
pin	Reports the calculated dimensions (width, height) of the displayed plot, which the function uses to scale an output file (where maximum dimensions are 7.5in for PNG and 10in for EPS and WMF). <b>Note:</b> Altering the <code>pin</code> has no effect.
tit.loc	Specifies where the title box can be placed using one of <code>legend</code> 's keywords: <code>c("bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", "center")</code> .
leg.loc	Specifies position of colour-coded legend using relative <code>c(x,y)</code> coordinates 0:1.
leg.font	Specifies legend font (e.g., 1=normal, 2=bold) within the <code>mono</code> family.

during the interactive mapping session using the `PBSmodelling` functions `getPBSOptions` and `setPBSOptions`.

**Manual control:** On the command line, `getPBSOptions("par.map")` displays the list of additional parameters. One or more of the parameters can be changed by `setPBSOptions` using the following format as an example:

```
setPBSOptions("par.map",list(mgp=c(3,1,0),cex.txt=1.2),sublist=TRUE)
```

## 5 GUI demo

The following pages display results from the `PBSmapx` demo, which can be run by issuing the following command:

```
createMap(hnam=paste(system.file(package="PBSmapx"),"/win/mapHist.txt",sep=""))
```

or more simply:

```
demo(D01_map)
```



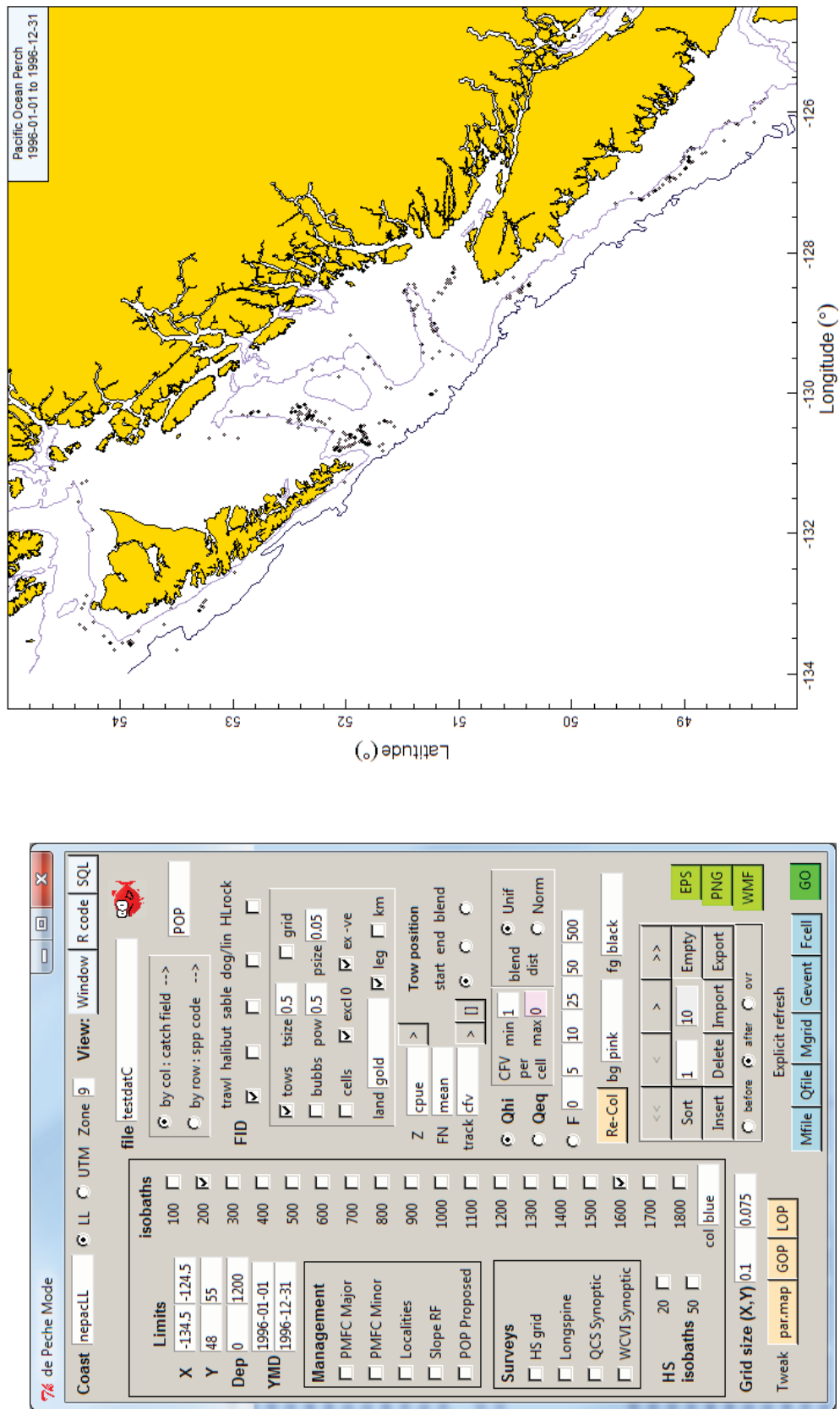


Figure 3. Pacific Ocean Perch (*Sebastes alutus*) tow locations in 1996.

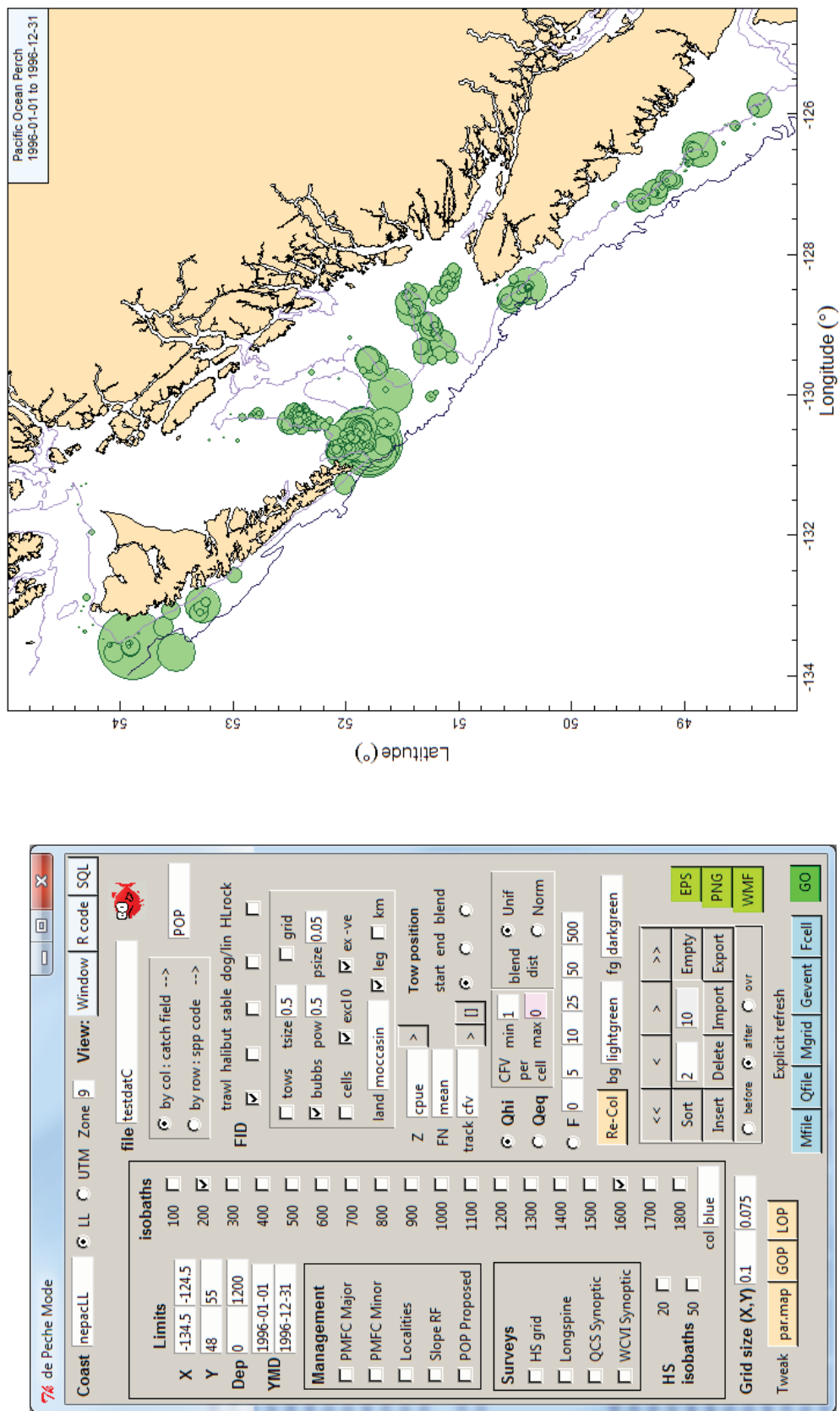
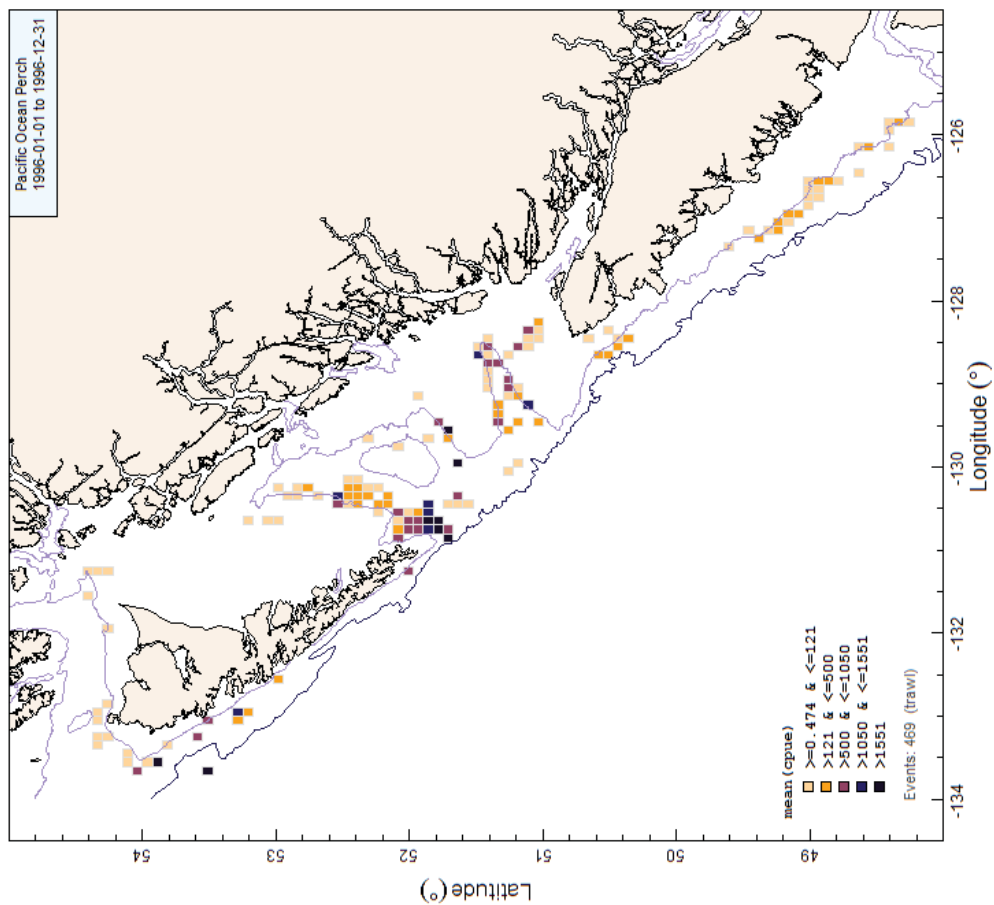


Figure 4. Pacific Ocean Perch CPUE (km/h) proportional to bubble size.



7% de Pêche Mode

Coast nepacLL ☐ LL ☐ UTM Zone 9 View: Window R code SQL

file testdatC  POP

☒ by col: catch field --> ☐ by row: spp code -->

trawl halibut sable dog/lin HLock ☐ FID ☐ ☐ ☐ ☐

☐ tows tsize 0.5 ☐ grid ☐ ☐ bubbs pow 0.5 psize 0.05 ☒ cells ☒ excl0 ☒ ex -ve ☐ land linen ☐ leg ☐ km

Z  cpue  Tow position ☐ start ☐ end ☐ blend

FN  mean  track cfv

☒ Qhi CFV min 1 ☐ blend ☐ Unif ☐ per cell max 12 ☐ dist ☐ Norm

☐ Qeq ☐ F 0 5 10 25 50 500

Re-Col  bg orange  fg navy

<< < > >>

Sort 3 10 Empty

☐ before ☐ after ☐ ovr Explicit refresh

isobaths ☐ 100 ☐ 200 ☒ 300 ☐ 400 ☐ 500 ☐ 600 ☐ 700 ☐ 800 ☐ 900 ☐ 1000 ☐ 1100 ☐ 1200 ☐ 1300 ☐ 1400 ☐ 1500 ☒ 1600 ☐ 1700 ☐ 1800 ☐ col blue

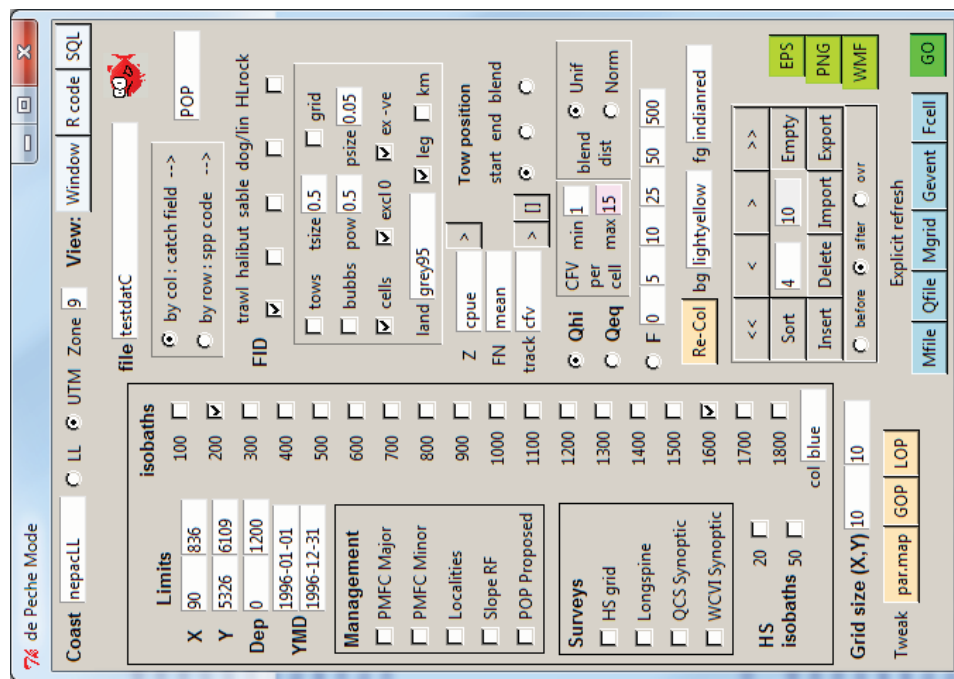
Limits X -134.5 -124.5 Y 48 55 Dep 0 1200 YMD 1996-01-01 1996-12-31

Management ☐ PMFC Major ☐ PMFC Minor ☐ Localities ☐ Slope RF ☐ POP Proposed

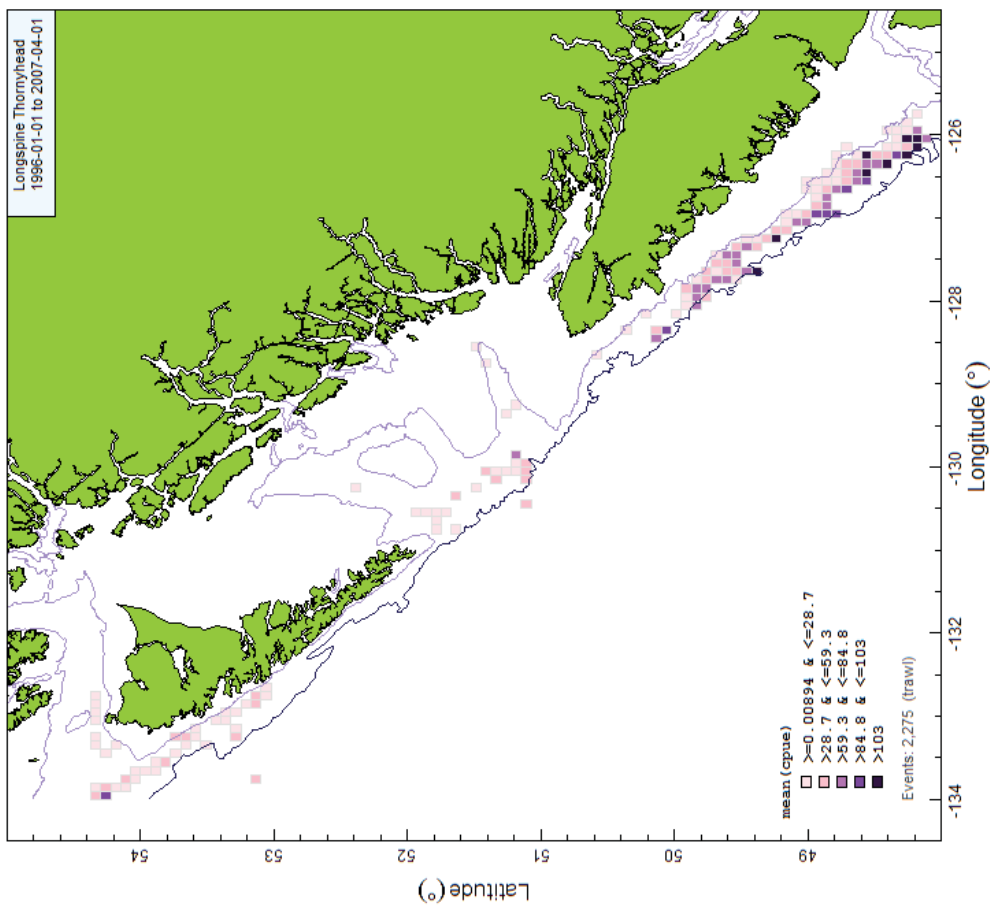
Surveys ☐ HS grid ☐ Longspine ☐ QCS Synoptic ☐ WCVI Synoptic

HS 20 ☐ isobaths 50 ☐ Grid size (X,Y) 0.1 0.075

Figure 5. Pacific Ocean Perch mean CPUE (km/h) represented by coloured grid cells.



12



7% de Pêche Mode

Coast nepacLL ☐ LL ☐ UTM Zone 9 View: Window R code SQL

file testdatC

☒ by col: catch field --> ☐ by row: spp code -->

LST

trawl halibut sable dog/lin HLRock

FID ☒ ☐ ☐ ☐ ☐

☐ tows ☐ tsz 0.5 ☐ grid

☐ bubbs ☐ pow 0.5 ☐ psiz 0.05

☒ cells ☒ excl 0 ☒ ex -ve

land chartreuse ☒ leg ☐ km

Z ☐ cpue ☐ Tow position

FN ☐ mean ☐ start end blend

track cfv ☐ ☐ ☐

☒ Qhi CFV min 1 ☐ blend ☐ Unif

☐ Qeq per cell max 54 ☐ dist ☐ Norm

☐ F 0 5 10 25 50 500

Re-Col bg pink fg purple

<< < > >>

Sort 5 10 Empty

Insert Delete Import Export

☐ before ☒ after ☐ ovr

Explicit refresh

Mfile Qfile Mgrid Gevent Fcell

GO

isobaths

Limits

X -134.5 -124.5

Y 48 55

Dep 0 1200

YMD 1996-01-01 2007-04-01

Management

☐ PMFC Major

☐ PMFC Minor

☐ Localities

☐ Slope RF

☐ POP Proposed

Surveys

☐ HS grid

☐ Longspine

☐ QCS Synoptic

☐ WCVI Synoptic

HS 20 ☐

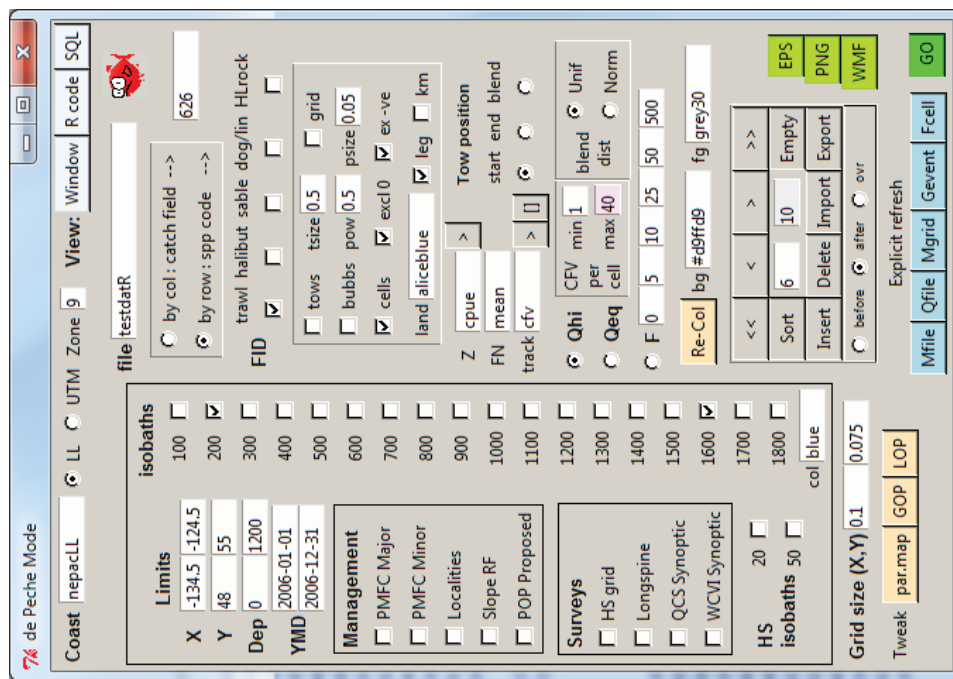
isobaths 50 ☐

col blue

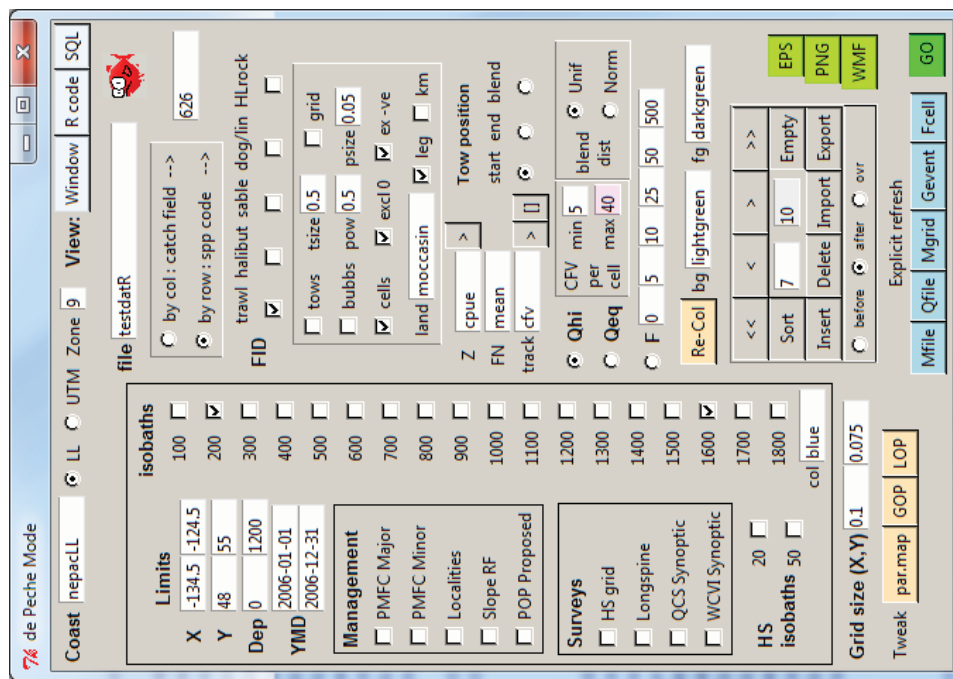
Grid size (X,Y) 0.1 0.075

Tweak par.map GOP LOP

Figure 7. Longspine Thornyhead (*Sebastolobus altivelis*) mean CPUE (km/h) for the period 1996–2007.



**Figure 8.** Dover Sole (*Microstomus pacificus*) mean CPUE (km/h) for 1996.



15





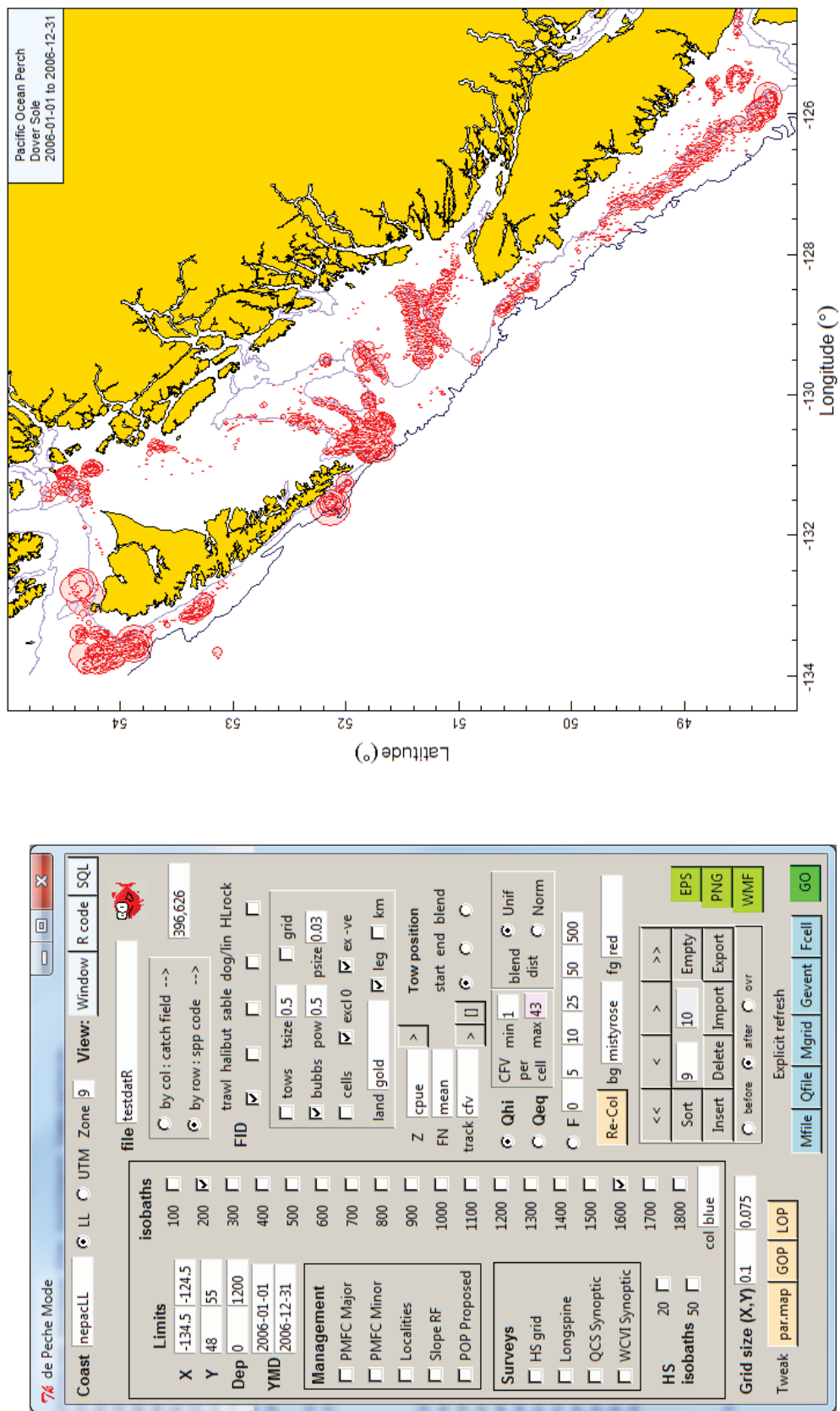


Figure 11. POP and Dover Sole CPUE as in Fig. 10 but represented as bubbles at each tow location.

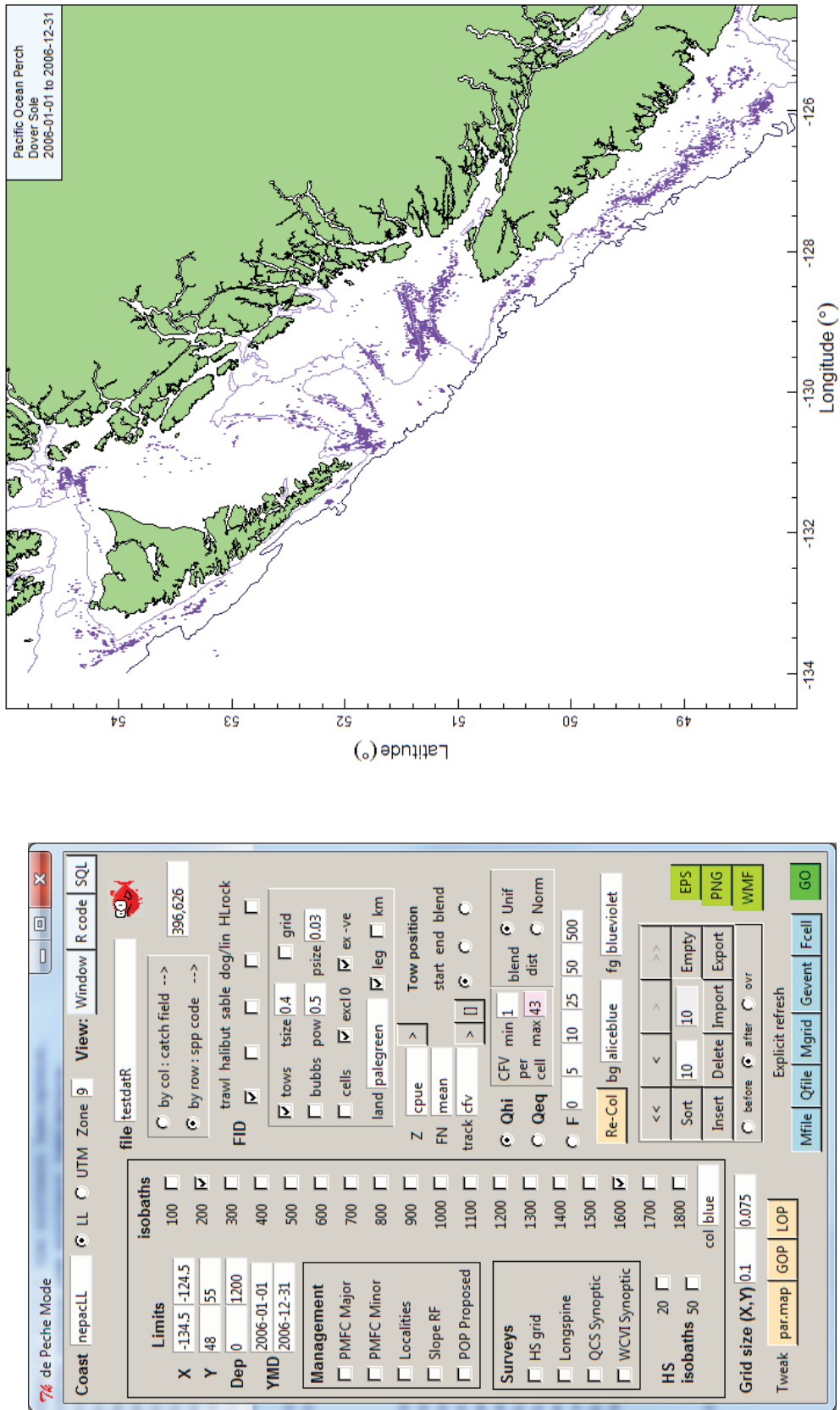


Figure 12. Position of tows capturing POP and/or Dover Sole in 1996.

## Contact Information

### Rowan Haigh

Research Biologist, Groundfish Section  
Marine Ecosystems and Aquaculture Division  
Fisheries and Oceans Canada  
Pacific Biological Station, Nanaimo, BC V9T 6N7  
Tel. +1 250-756-7123 fac simile +1 250-756-7053  
Email [rowan.haigh@dfo-mpo.gc.ca](mailto:rowan.haigh@dfo-mpo.gc.ca)

## References

- Boers NM, Haigh R, Schnute JT (2004). “PBS Mapping 2: developer’s guide.” *Canadian Technical Report of Fisheries and Aquatic Sciences*, **2550**, iv + 38. URL <http://www.dfo-mpo.gc.ca/Library/285693.pdf>.
- Bolker BM (2008). *Ecological Models and Data in R*. Princeton University Press, Princeton, NJ.
- Schnute JT, Boers NM, Haigh R (2004). “PBS Mapping 2: user’s guide.” *Canadian Technical Report of Fisheries and Aquatic Sciences*, **2549**, viii + 126. URL <http://www.dfo-mpo.gc.ca/Library/285683.pdf>.
- Schnute JT, Couture-Beil A, Haigh R (2006). “PBS Modelling 1: user’s guide.” *Canadian Technical Report of Fisheries and Aquatic Sciences*, **2674**, viii + 114. URL <http://www.dfo-mpo.gc.ca/Library/326794.pdf>.