Example script for SpatialVAM for spatio-temporal analysis of species interactions data

James Thorson

October 26, 2016

Contents

1 Overview

This tutorial will walk through a simple example of how to use `SpatialVAM` for estimating species interactions.

2 Getting started

To install TMB on a windows machine, we need to first install Rtools. During the installation, please select the option to have Rtools included in your system path. On other operating systems, it is not necessary to install Rtools. We then install `VAST`.

```
devtools::install_github("james-thorson/SpatialVAM")
```

We also install `FishData`, which is used to download data for our example

```
devtools::install_github("james-thorson/FishData")
```

Next load libraries.

```
library(TMB)                # Can instead load library(TMBdebug)
```

```
## Warning: package 'TMB' was built under R version
## 3.3.2
```

```
library(SpatialVAM)
```

2.1 Further information

If you have further questions after reading this tutorial, please explore the GitHub repo mainpage, wiki, and glossary. Also please explore the R help files, e.g., e.g., `?Data_Fn` for explanation of data inputs, or `?Param_Fn` for explanation of parameters.

2.2 Related tools

Related tools for spatio-temporal fisheries analysis are currently housed at www.FishStats.org. These include VAST, a a multispecies model for estimating abundance using delta-models, and www.FishViz.org, a tool for visualizing single-species results using worldwide. `VAST` and `SpatialDeltaGLMM` both use continuous integration to confirm that they give identical estimates when applied to single-species data.

2.3 How to cite SpatialVAM

`SpatialVAM` has involved many publications for developing individual features. If using `SpatialVAM`, please read and cite:

```
citation("SpatialVAM")
```

```
##
## To cite package 'SpatialVAM' in publications
## use:
##
##   James Thorson (2015). SpatialVAM: Spatial
##   vector autoregressive model. R package
##   version 1.0.
##   http://github.com/James-Thorson/SpatialVAM
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {SpatialVAM: Spatial vector autoregressive model},
##     author = {James Thorson},
##     year = {2015},
##     note = {R package version 1.0},
##     url = {http://github.com/James-Thorson/SpatialVAM},
##   }
##
## ATTENTION: This citation information has
## been auto-generated from the package
## DESCRIPTION file and may need manual
## editing, see 'help("citation")'.
```

and also browse the GitHub list of papers.

3 Settings

We use latest version for CPP code

```
Version = "spatial_vam_v14"
```

3.1 Spatial settings

The following settings define the spatial resolution for the model, and whether to use a grid or mesh approximation

```
n_x = c(50, 100)[1] # Number of stations
Kmeans_Config = list( "randomseed"=1, "nstart"=100, "iter.max"=1e3 )
```

3.2 Model settings

The following settings define whether to include spatial and spatio-temporal variation, the rank of this covariance among species, whether its autocorrelated, and whether there's overdispersion

```
Nfactors_est = 3  # Number of dynamic factors in process error
Ncointegrate = 3
Use_REML = FALSE
Estimate_Phi = TRUE  # Phi is the offset of initial and equilibrium abundance
StartFromEquilibriumTF = FALSE
```

```
B_type = c("Independent", "Real_eigenvalue", "Complex_eigenvalue")[3]
Kappa = c("constant", "spatial_vs_spatiotemporal",
    "different")[1]
EigenBounds = c(Lower = -2, Upper = -0.001)
ObsModel = c("Poisson", "LNP", "ZILN")[3]
```

3.3 Stratification for results

We also define any potential stratification of results, and settings specific to any case-study data set

```
strata.limits <- data.frame(STRATA = "All_areas")
```

3.4 Derived objects

In this case, we'll use publicly available data for three groundfishes in the Eastern Bering Sea, so we set `Region` and `Species_set` accordingly. `Region` is used to define both the database for downloading data, as well as the region for extrapolation density, while `Species_set` is only used when downloading data.

```
Region = "Eastern_Bering_Sea"
Species_set = c("Atheresthes stomias","Gadus chalcogrammus","Hippoglossoides elassodon")
```

3.5 Save settings

We then set the location for saving files.

```
DateFile = paste0(getwd(),'/SpatialVAM_output/')
  dir.create(DateFile)
```

4 Prepare the data

4.1 Data-frame for catch-rate data

We then download data for three species using `FishData`.

```
DF = FishData::download_catch_rates(survey = "Eastern_Bering_Sea",
    species_set = Species_set)
Data_Geostat = cbind(spp = DF[, "Sci"], Year = DF[,
    "Year"], Catch_KG = DF[, "Wt"], AreaSwept_km2 = 0.01,
    Vessel = 0, Lat = DF[, "Lat"], Lon = DF[, "Long"])
```

The data is formatted as shown here, with head...

	spp	Year	Catch_KG	AreaSwept_km2	Vessel	Lat	Lon
1982__A-02__67	1	1982	6.98	0.01	0	55	-167
1982__A-03__59	1	1982	4.37	0.01	0	55	-166
1982__A-04__66	1	1982	12.6	0.01	0	55	-166
1982__A-05__58	1	1982	4.28	0.01	0	55	-165
1982__A-06__38	1	1982	0	0.01	0	55	-165
1982__B-02__68	1	1982	10.3	0.01	0	55.3	-167

... and tail

	spp	Year	Catch_KG	AreaSwept_km2	Vessel	Lat	Lon
2016_U-29_156	3	2016	1.15	0.01	0	61.7	-176
2016_V-25_152	3	2016	0	0.01	0	62	-174
2016_V-26_153	3	2016	0	0.01	0	62	-174
2016_V-27_154	3	2016	0	0.01	0	62	-175
2016_V-28_155	3	2016	0	0.01	0	62	-176
2016_Z-05_73	3	2016	28	0.01	0	54.7	-165

4.2 Extrapolation grid

We also generate the extrapolation grid appropriate for a given region. For new regions, we use `Region="Other"`.

```
Extrapolation_List = SpatialDeltaGLMM::Prepare_Extrapolation_Data_Fn(Region = Region,
    strata.limits = strata.limits)
```

4.3 Derived objects for spatio-temporal estimation

And we finally generate the information used for conducting spatio-temporal parameter estimation, bundled in list `Spatial_List`

```
Spatial_List = SpatialDeltaGLMM::Spatial_Information_Fn(grid_size_km = 100,
    n_x = n_x, Method = "Mesh", Lon = Data_Geostat[,
        "Lon"], Lat = Data_Geostat[, "Lat"], Extrapolation_List = Extrapolation_List,
    randomseed = Kmeans_Config[["randomseed"]], nstart = Kmeans_Config[["nstart"]],
    iter.max = Kmeans_Config[["iter.max"]], DirPath = DateFile)
# Add knots to Data_Geostat
Data_Geostat = cbind(Data_Geostat, Spatial_List$loc_UTM,
    knot_i = Spatial_List$knot_i)
```

5 Build and run model

5.1 Build model

To estimate parameters, we first build a list of data-inputs used for parameter estimation. `Data_Fn` has some simple checks for buggy inputs, but also please read the help file `?Data_Fn`.

```
ObsModel_p = rep(switch(ObsModel, Poisson = 0, Lognormal = 1,
    ZILN = 2, LNP = 3), length(unique(Data_Geostat[,
    "spp"])))

TmbData = Data_Fn(Version = Version, obsmodel_p = ObsModel_p,
    n_cointegrate = Ncointegrate, b_i = Data_Geostat[,
        "Catch_KG"], s_i = Data_Geostat[, "knot_i"],
    t_i = Data_Geostat[, "Year"], p_i = Data_Geostat[,
        "spp"], a_x = Spatial_List$a_xl[, 1], B_type = B_type,
```

```
    startFromEquilibriumTF = FALSE, spatial_method = 0,
    MeshList = Spatial_List$MeshList, n_factors = Nfactors_est)
```

We then build the TMB object.

```
TmbList = Build_TMB_Fn(TmbData = TmbData, Version = Version,
    use_REML = ifelse(is.na(Use_REML), TRUE, Use_REML),
    loc_x = Spatial_List$MeshList$loc_x, estimate_phi = Estimate_Phi,
    Kappa = Kappa, eigenbounds = EigenBounds, RunDir = DateFile)
obj = TmbList$Obj  # 'Parameters'=InputList$TmbParams,
```

5.2 Estimate fixed effects and predict random effects

Next, we use a gradient-based nonlinear minimizer to identify maximum likelihood estimates for fixed-effects

```
Opt = TMBhelper::Optimize(obj = obj, lower = TmbList$Lower,
    upper = TmbList$Upper, getsd = TRUE, savedir = DateFile,
    newtonsteps = 3)
```

Finally, we bundle and save output

```
Report = obj$report()
ParHat = obj$env$parList()
```

6 Diagnostic plots

We first apply a set of standard model diagnostics to confirm that the model is reasonable and deserves further attention. If any of these do not look reasonable, the model output should not be interpreted or used.

6.1 Plot data

It is always good practice to conduct exploratory analysis of data. Here, I visualize the spatial distribution of data. Spatio-temporal models involve the assumption that the probability of sampling a given location is statistically independent of the probability distribution for the response at that location. So if sampling "follows" changes in density, then the model is probably not appropriate!

```
SpatialDeltaGLMM::Plot_data_and_knots(Extrapolation_List = Extrapolation_List,
    Spatial_List = Spatial_List, Data_Geostat = Data_Geostat,
    PlotDir = DateFile)
```

6.2 Convergence

Here I print the diagnostics generated during parameter estimation, and I confirm that (1) no parameter is hitting an upper or lower bound and (2) the final gradient for each fixed-effect is close to zero. For explanation of parameters, please see `?Data_Fn`.

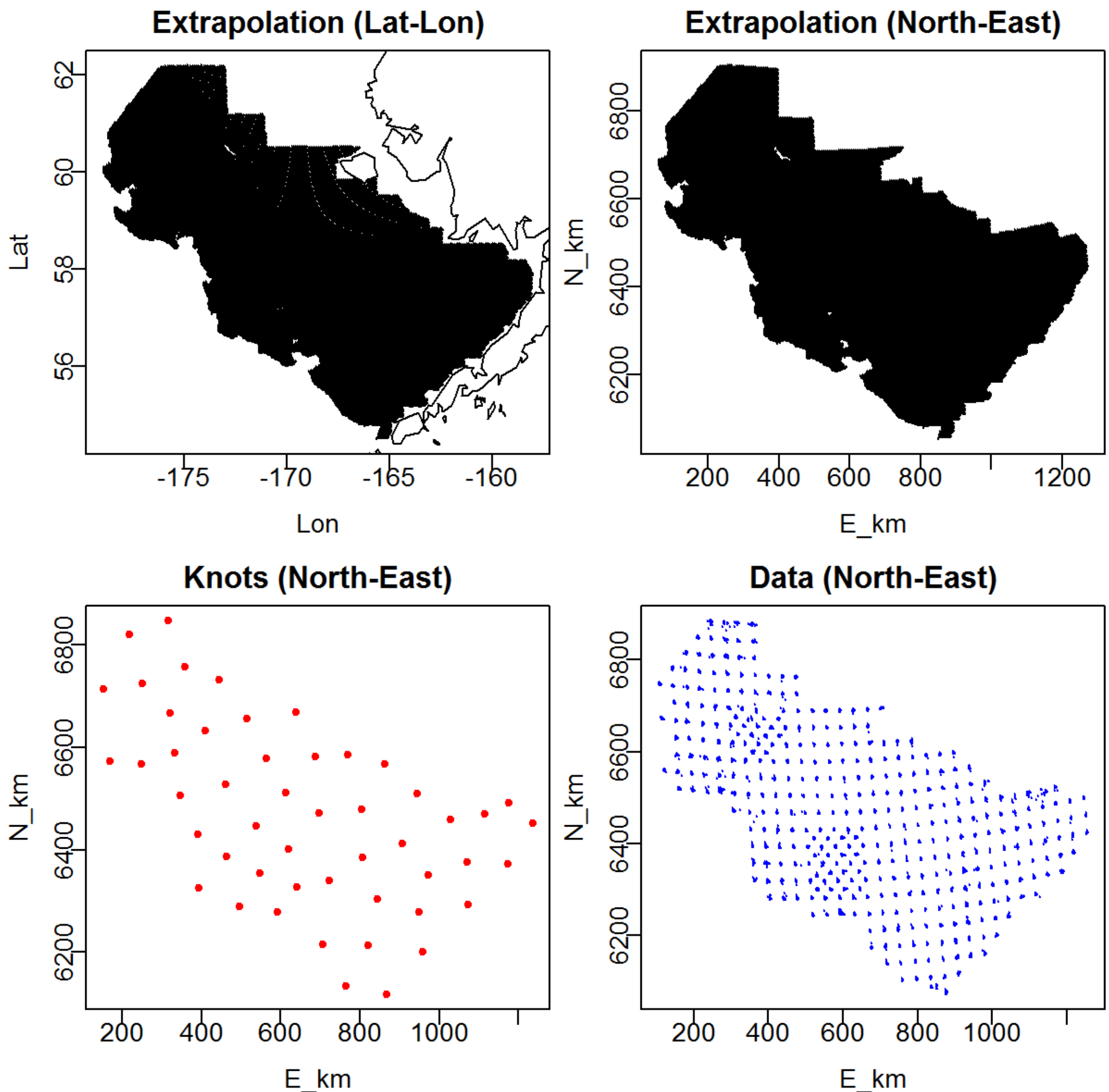

Figure 1: Spatial extent and location of knots

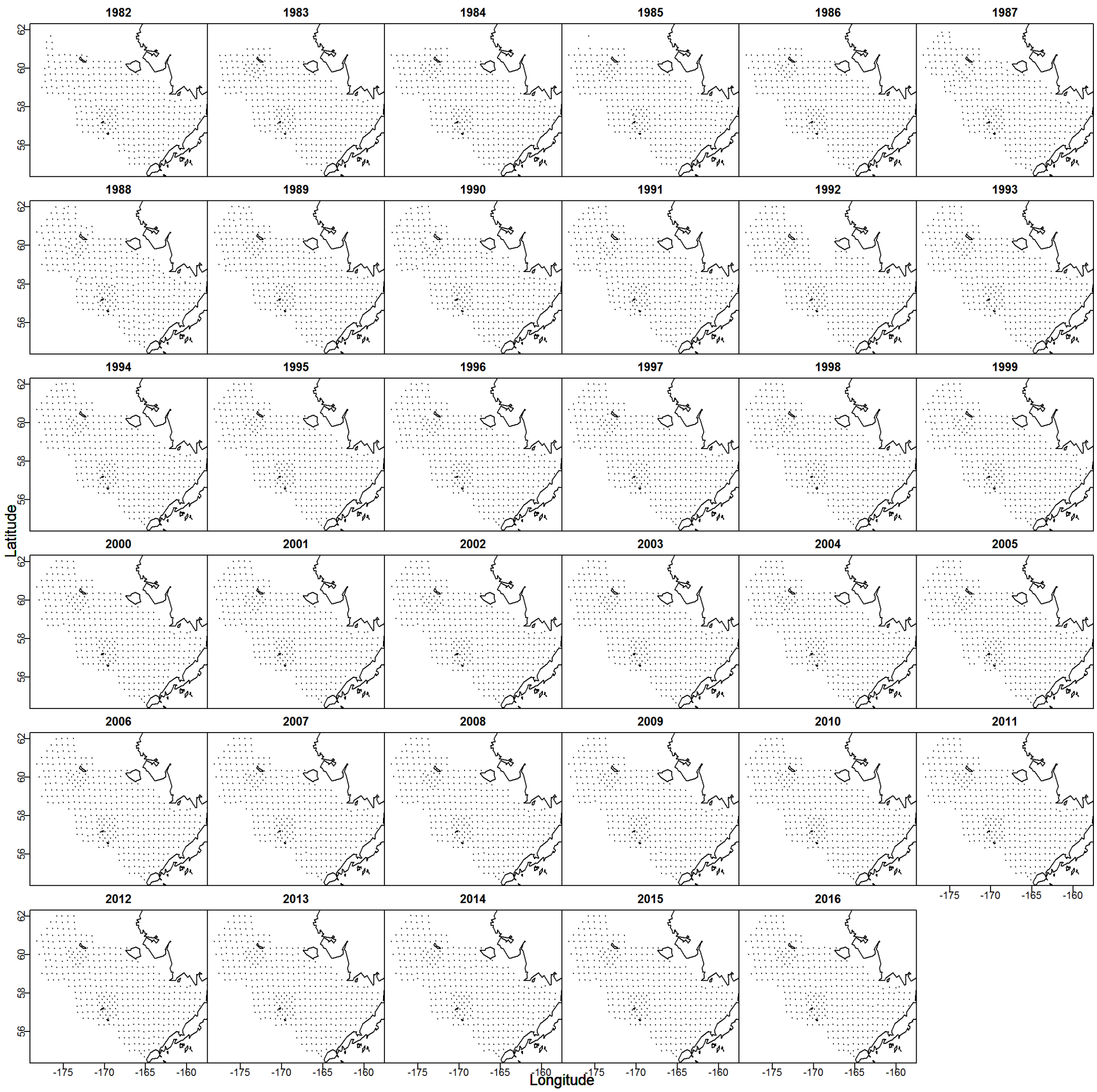

Figure 2: Spatial distribution of catch-rate data

```r
pander::pandoc.table( Opt$diagnostics[,c('Param','Lower','MLE','Upper','final_gradient')] )
```

Param	Lower	MLE	Upper	final_gradient
logkappa_z	-5.978	-5.022	-3.114	3.365e-09
alpha_p	-Inf	-0.9313	Inf	9.687e-14
alpha_p	-Inf	0.9253	Inf	4.551e-14
alpha_p	-Inf	-0.03329	Inf	-6.523e-15
phi_p	-Inf	-1.044	Inf	-3.975e-13
phi_p	-Inf	1.297	Inf	3.558e-14
phi_p	-Inf	0.3537	Inf	4.109e-13
logMargSigmaA_p	-4.605	2.444	Inf	3.143e-09
logMargSigmaA_p	-4.605	1.987	Inf	9.493e-12
logMargSigmaA_p	-4.605	1.536	Inf	-1.519e-10
L_val	-Inf	0.7269	Inf	1.055e-09
L_val	-Inf	0.1435	Inf	-6.608e-11
L_val	-Inf	0.1307	Inf	-3.331e-10
L_val	-Inf	0.8041	Inf	1.019e-10
L_val	-Inf	0.1115	Inf	-6.999e-13
L_val	-Inf	0.2953	Inf	3.105e-10
Alpha_pr	-Inf	-0.5651	Inf	5.677e-09
Alpha_pr	-Inf	0.005647	Inf	-1.981e-10
Alpha_pr	-Inf	0.007715	Inf	-3.245e-09
Alpha_pr	-Inf	0.01215	Inf	-2.229e-10
Alpha_pr	-Inf	-0.3642	Inf	2.87e-11
Alpha_pr	-Inf	0.04586	Inf	1.673e-11
Alpha_pr	-Inf	0.1008	Inf	8.689e-10
Alpha_pr	-Inf	-0.07022	Inf	-2.646e-11
Alpha_pr	-Inf	-0.2667	Inf	-5.645e-10
logsigma_pz	-Inf	0.3294	Inf	5.648e-11
logsigma_pz	-Inf	0.2569	Inf	-3.894e-12
logsigma_pz	-Inf	0.6897	Inf	-2.744e-11
logsigma_pz	-Inf	-0.2089	Inf	-3.666e-12
logsigma_pz	-Inf	0.3	Inf	1.106e-11
logsigma_pz	-Inf	0.4046	Inf	1.357e-12

6.3 Model selection

To select among models, we recommend using the Akaike Information Criterion, AIC, via `Opt$AIC= 2.477\times 10^{5}`.

7 Model output

Last but not least, we generate useful plots by first determining which years to plot (`Years2Include`), and labels for each plotted year (`Year_Set`)

```r
Year_Set = min(DF[,'Year']):max(DF[,'Year'])
```

We then run a set of pre-defined plots for visualizing results

7.1 Density surface for each year

We can visualize many types of output from the model. Here I only show predicted density, but other options are obtained via other integers passed to `plot_set` as described in `?PlotResultsOnMap_Fn`

```r
# Get region-specific settings for plots
MapDetails_List = SpatialDeltaGLMM::MapDetails_Fn(Region = Region,
    NN_Extrap = Spatial_List$PolygonList$NN_Extrap,
    Extrapolation_List = Extrapolation_List)
# Plot maps representing density or other variables
SpatialDeltaGLMM::PlotResultsOnMap_Fn(plot_set = c(3),
    MappingDetails = MapDetails_List[["MappingDetails"]],
    Report = Report, Sdreport = Opt$SD, PlotDF = MapDetails_List[["PlotDF"]],
    MapSizeRatio = MapDetails_List[["MapSizeRatio"]],
    Xlim = MapDetails_List[["Xlim"]], Ylim = MapDetails_List[["Ylim"]],
    FileName = DateFile, category_names = unique(DF$Sci),
    Year_Set = Year_Set, Rotate = MapDetails_List[["Rotate"]],
    Cex = MapDetails_List[["Cex"]], Legend = MapDetails_List[["Legend"]],
    zone = MapDetails_List[["Zone"]], mar = c(0, 0,
        2, 0), oma = c(3.5, 3.5, 0, 0), cex = 1.8)
```

7.2 Index of abundance

The index of abundance is generally most useful for stock assessment models.

```r
Index = SpatialDeltaGLMM::PlotIndex_Fn(DirName = DateFile,
    TmbData = TmbData, Sdreport = Opt$SD, Year_Set = min(Data_Geostat[,
        "Year"]):max(Data_Geostat[, "Year"]), Years2Include = which(min(Data_Geostat[,
        "Year"]):max(Data_Geostat[, "Year"]) %in% sort(unique(Data_Geostat[,
        "Year"]))), strata_names = names(strata.limits)[1],
    category_names = levels(DF[, "Sci"]), use_biascorr = TRUE)
pander::pandoc.table(Index$Table[, c("Category", "Year",
    "Estimate_metric_tons", "SD_mt")])
```

Category	Year	Estimate_metric_tons	SD_mt
Atheresthes_stomias	1982	457.9	54.11
Atheresthes_stomias	1983	772.9	97.71
Atheresthes_stomias	1984	848.2	111.1
Atheresthes_stomias	1985	1280	182.3
Atheresthes_stomias	1986	1281	182.8
Atheresthes_stomias	1987	2308	300.4
Atheresthes_stomias	1988	1793	248.1
Atheresthes_stomias	1989	2415	304.1
Atheresthes_stomias	1990	2299	329.4
Atheresthes_stomias	1991	1708	291
Atheresthes_stomias	1992	1262	187.3
Atheresthes_stomias	1993	2955	377.8
Atheresthes_stomias	1994	2509	389.9
Atheresthes_stomias	1995	1654	258.2
Atheresthes_stomias	1996	2716	349.2
Atheresthes_stomias	1997	2214	308.4

Category	Year	Estimate_metric_tons	SD_mt
Atheresthes_stomias	1998	2370	305
Atheresthes_stomias	1999	878	138.5
Atheresthes_stomias	2000	1615	212.9
Atheresthes_stomias	2001	2542	340.8
Atheresthes_stomias	2002	2114	266.2
Atheresthes_stomias	2003	3486	385
Atheresthes_stomias	2004	3731	432.7
Atheresthes_stomias	2005	5341	588.2
Atheresthes_stomias	2006	4220	559.6
Atheresthes_stomias	2007	2828	389.2
Atheresthes_stomias	2008	3329	446.2
Atheresthes_stomias	2009	1967	294.3
Atheresthes_stomias	2010	3081	407
Atheresthes_stomias	2011	3989	494.3
Atheresthes_stomias	2012	1928	256.5
Atheresthes_stomias	2013	2433	313.2
Atheresthes_stomias	2014	3544	425.1
Atheresthes_stomias	2015	2779	314.9
Atheresthes_stomias	2016	3871	394.1
Gadus_chalcogrammus	1982	9164	1184
Gadus_chalcogrammus	1983	22406	3261
Gadus_chalcogrammus	1984	14393	2129
Gadus_chalcogrammus	1985	19700	3034
Gadus_chalcogrammus	1986	15985	2262
Gadus_chalcogrammus	1987	15624	2212
Gadus_chalcogrammus	1988	27130	3880
Gadus_chalcogrammus	1989	22541	2982
Gadus_chalcogrammus	1990	14497	2023
Gadus_chalcogrammus	1991	21912	2855
Gadus_chalcogrammus	1992	14823	1925
Gadus_chalcogrammus	1993	24315	3008
Gadus_chalcogrammus	1994	19209	2527
Gadus_chalcogrammus	1995	10313	1340
Gadus_chalcogrammus	1996	12105	1483
Gadus_chalcogrammus	1997	11631	1638
Gadus_chalcogrammus	1998	11084	1441
Gadus_chalcogrammus	1999	13155	1876
Gadus_chalcogrammus	2000	13863	1643
Gadus_chalcogrammus	2001	17032	2047
Gadus_chalcogrammus	2002	16363	1971
Gadus_chalcogrammus	2003	25913	3078
Gadus_chalcogrammus	2004	17489	2000
Gadus_chalcogrammus	2005	16641	2040
Gadus_chalcogrammus	2006	13160	1780
Gadus_chalcogrammus	2007	11295	2218
Gadus_chalcogrammus	2008	6553	1126
Gadus_chalcogrammus	2009	4703	710.3
Gadus_chalcogrammus	2010	10245	2149
Gadus_chalcogrammus	2011	14001	1970
Gadus_chalcogrammus	2012	12254	1624
Gadus_chalcogrammus	2013	17698	2565
Gadus_chalcogrammus	2014	37008	4922

Category	Year	Estimate_metric_tons	SD_mt
Gadus_chalcogrammus	2015	41355	5026
Gadus_chalcogrammus	2016	27635	3387
Hippoglossoides_elassodon	1982	962.8	69.49
Hippoglossoides_elassodon	1983	1547	131.2
Hippoglossoides_elassodon	1984	1670	150
Hippoglossoides_elassodon	1985	1883	169.7
Hippoglossoides_elassodon	1986	2119	190.7
Hippoglossoides_elassodon	1987	2380	214.5
Hippoglossoides_elassodon	1988	2701	247.9
Hippoglossoides_elassodon	1989	2990	265
Hippoglossoides_elassodon	1990	3262	289.1
Hippoglossoides_elassodon	1991	3618	334
Hippoglossoides_elassodon	1992	3372	299.9
Hippoglossoides_elassodon	1993	4106	371.3
Hippoglossoides_elassodon	1994	4349	408.3
Hippoglossoides_elassodon	1995	3436	315.8
Hippoglossoides_elassodon	1996	3440	296.6
Hippoglossoides_elassodon	1997	3377	282.5
Hippoglossoides_elassodon	1998	3885	329.5
Hippoglossoides_elassodon	1999	2227	200.8
Hippoglossoides_elassodon	2000	2203	186.9
Hippoglossoides_elassodon	2001	2577	218.3
Hippoglossoides_elassodon	2002	2607	221.5
Hippoglossoides_elassodon	2003	2781	233.3
Hippoglossoides_elassodon	2004	3108	259.5
Hippoglossoides_elassodon	2005	3413	283.5
Hippoglossoides_elassodon	2006	3143	269.6
Hippoglossoides_elassodon	2007	2991	261.5
Hippoglossoides_elassodon	2008	2538	225.9
Hippoglossoides_elassodon	2009	1659	152.2
Hippoglossoides_elassodon	2010	1783	162.1
Hippoglossoides_elassodon	2011	1962	170.4
Hippoglossoides_elassodon	2012	1630	144.8
Hippoglossoides_elassodon	2013	1674	141
Hippoglossoides_elassodon	2014	1913	155.7
Hippoglossoides_elassodon	2015	2009	169.5
Hippoglossoides_elassodon	2016	2626	223.4

7.3 Center of gravity and range expansion/contraction

We can detect shifts in distribution or range expansion/contraction.

```
SpatialDeltaGLMM::Plot_range_shifts(Report = Report,
    TmbData = TmbData, Sdreport = Opt$SD, Znames = colnames(TmbData$Z_xm),
    PlotDir = DateFile, category_names = unique(DF$Sci),
    Year_Set = Year_Set)
```

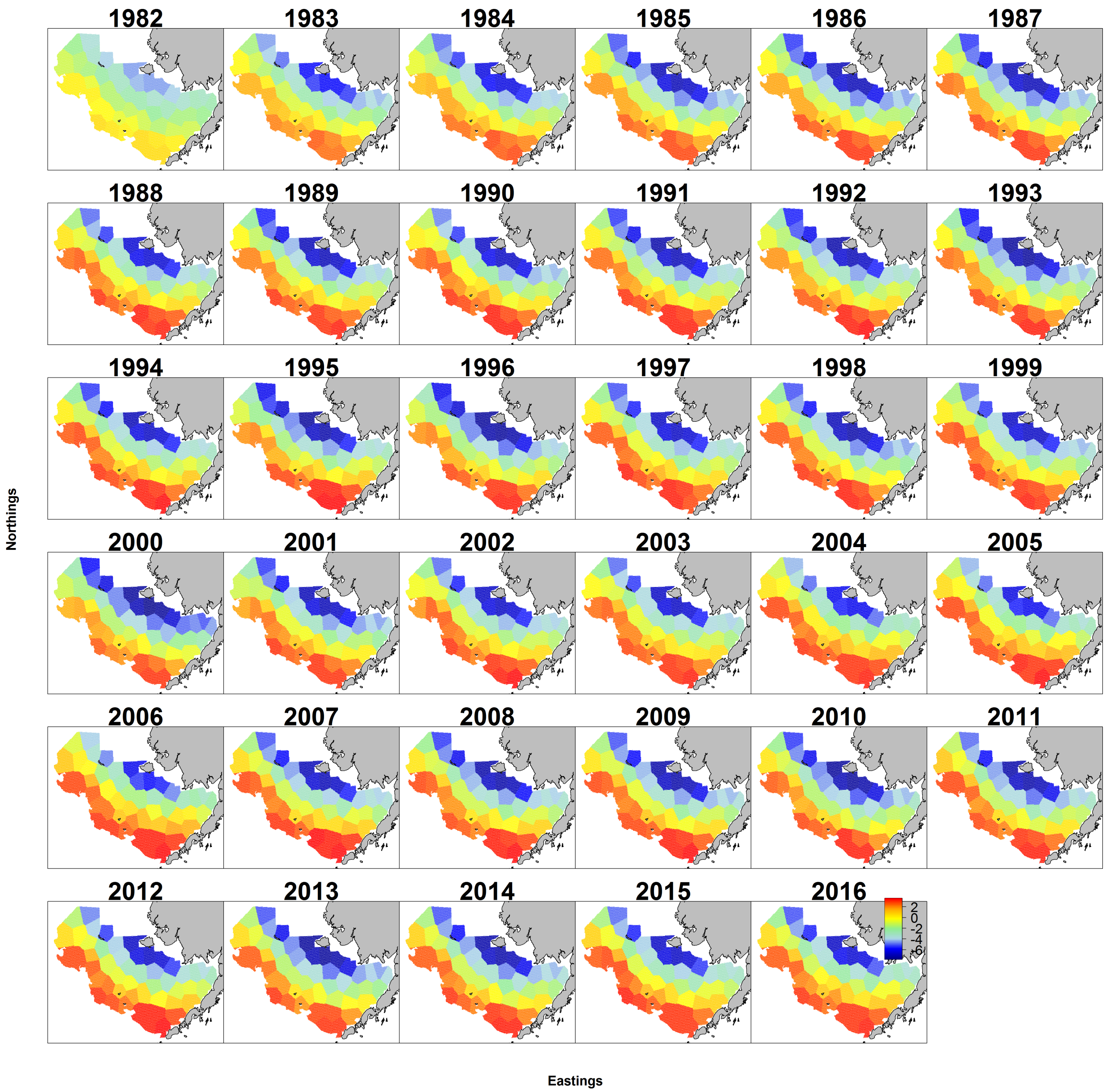

Figure 3: Density maps for each year for arrowtooth flounder

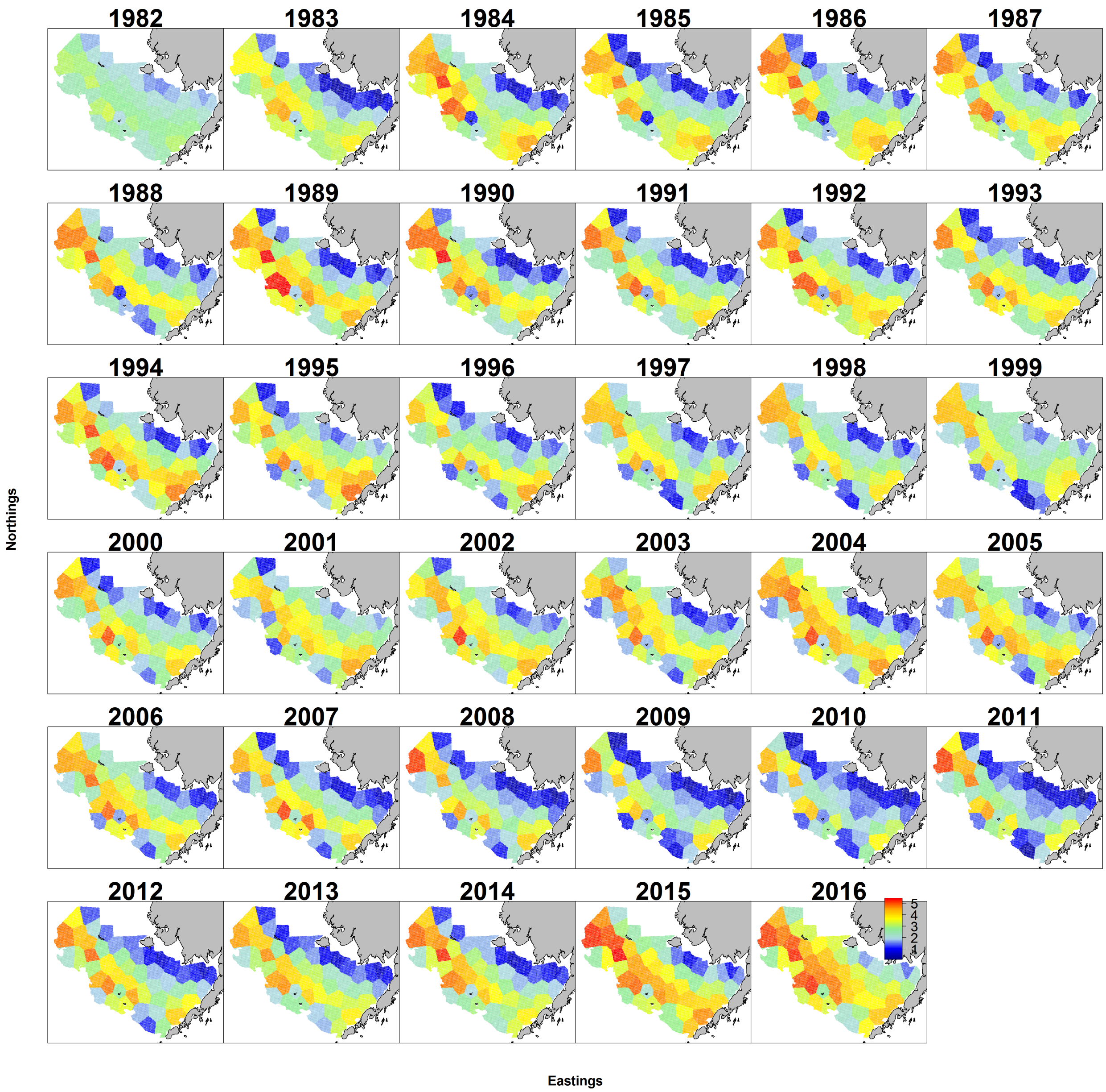

Figure 4: Density maps for each year for Alaska pollock

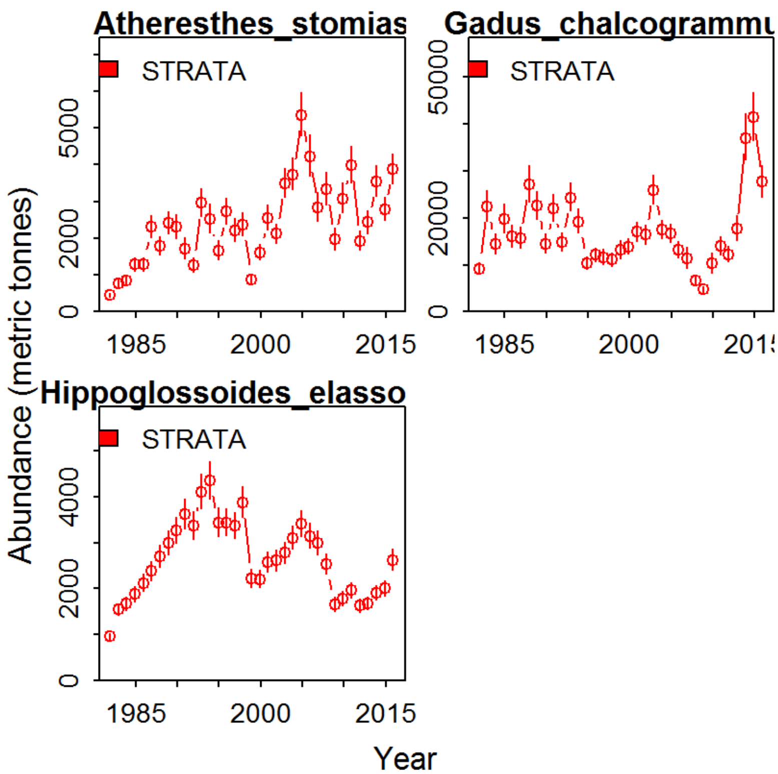

Figure 5: Index of abundance plus/minus 1 standard error