SOweb - Inputs

Andrew Constable

2023-12-30

Contents

```
# 1. Functions ####

# convert Rasters to extend from -180 to 180 (dateline) when original is
# extending from 0 longitude (Greenwich meridian) to 360 -
fConvertRasterEdgeFromGreenwhichToDateLine<-function(r){
ExtentWest<-ext(r); ExtentWest[1]<-ExtentWest[1]+180; ExtentWest[2]<-360
ExtentWestNew<-ExtentWest; ExtentWestNew[c(1:2)]<-ExtentWestNew[c(1:2)]-360
ExtentEast<-ext(r); ExtentEast[1]<-0; ExtentEast[2]<-ExtentEast[2]-180
rE<-crop(r,ExtentEast)
rW<-crop(r,ExtentWest)
ext(rW)<-ExtentWestNew
return(merge(rE,rW))
} # end convert



fPolyMeansCellWtd<-function # return statistic for spatVectors individually (AllVectors=TRUE) or com
( p # spatVector of polygons (may be more than 1)
  ,r # spatRaster
  ,NArm=TRUE
  ,AllVectors=FALSE
  ,wtCellArea=TRUE
){
  if(wtCellArea){CellArea<-cellSize(r,unit="km")} else CellArea<-r^0
  d<-extract(r,p)
  w<-extract(CellArea,p)
  if(NArm) x<-!is.na(d[,2]) else x<-TRUE
  if(AllVectors){
    l<-unique(d[,1])
    res<-lapply(l,function(i,d,w,x){
      v<-(x & d[,1]==i)
      weighted.mean(d[v,2],w[v,2])
    },d,w,x)
    return(data.frame(Polygon = l,Name = values(p),Value = unlist(res)))
  } else return (data.frame(Polygon = 1,Name = paste(c("a","b","c"),collapse ="_"),Value = weighted.
}


fPolyMediansCellWtd<-function # return statistic for spatVectors individually (AllVectors=TRUE) or c
                 ( p # spatVector of polygons (may be more than 1)
                  ,r # spatRaster
                  ,NArm=TRUE
                  ,AllVectors=FALSE
                  ,wtCellArea=TRUE
                  ){
  if(wtCellArea){CellArea<-cellSize(r,unit="km")} else CellArea<-r^0
  d<-extract(r,p)
  w<-extract(CellArea,p)

  if(NArm) x<-!is.na(d[,2]) else x<-TRUE
  if(AllVectors){
    l<-unique(d[,1])
    res<-lapply(l,function(i,d,w,x){
      v<-(x & d[,1]==i)
      if (sum(v)==0) return(NA) else return(weighted.median(d[v,2],w[v,2]))
    },d,w,x)
    return(data.frame(Polygon = l,Name = values(p),Value = unlist(res)))
  } else {
    if (sum(x)==0) res<-NA else res<-weighted.median(d[x,2],w[x,2])
    return (data.frame(Polygon = 1,Name = paste(c("a","b","c"),collapse ="_"),Value = res))
    }
  } # end function



# 2. Input data ####

MEASOareas<-"/Users/acon/Desktop/_w/_d/Shapefiles/MEASO\ polygons/MEASO_polygons.shp"
vMEASO <- vect(MEASOareas)
lMEASO<-as.lines(sMEASO)
plot(vMEASO)
```

Inputs

Steps: 1. Generate relevant Southern Ocean geotiff files for all georeferenced data. These provide the base from which different metrics may be generated from the same dataset.
2. Metrics are summarised for a polygon based on the georeferenced data.

Where time series are generated, time-stamped geotiff files are used for estimating metrics and values interpolated for time-steps between them. ’

Nutrients

Nutrients bound the estimation of biomasses of the different biological nodes.

Nitrogen, phosphorous and carbon are not limiting in the Southern Ocean. Only pools of dissolved silicate (silicic acid) and dissolved iron are included in the steady-state model as they are the limiting nutrients for phytoplankton. In both cases, a climatology of winter concentrations is used to initialise the deep water concentrations below the mixed layer. From that time, changes in the mixed layer and deep water pools are a product of exchange between the pools, primary production, recycling and remineralisation.

Slicic Acid.

The climatology for silicic acid ($\mu mol.kg^{-1}$ was obtained from the World Ocean Atlas 2018 (Boyer et al. 2018) on a 1-degree grid (https://www.ncei.noaa.gov/access/world-ocean-atlas-2018/). The objectively analyzed climatologies are used. These are the objectively interpolated mean fields for oceanographic variables at standard depth levels for the World Ocean. The mean concentration over October through December is used as distribution of silicate.

```
# World Ocean Atlas 2018 File naming conventions: [V][TT][FF][GG].[EXT]
# where:
# [V] - variable
# [TT] - time period
# [FF] - field type
# [GG] - grid (01- 1°, 04 - 1/4° 10 - 1/10°)
# [EXT] - file extention
# Note: '.dat' - ASCII; '.csv' - comma separated value; '.dbf', '.shp', '.shx' - ArcGIS shape files;

dirData<-"/Users/acon/Desktop/_w/_d/Ocean/Silicate/"
dirOut<-"./tif/"
fOut<-"Climatology_Si.tiff"
fData<-paste0(dirData,"woa18_all_i",sprintf("%02d",seq(1,12,1)),"_01.nc")

# for each month, calculate mean concentration in top 300 m (top 29 layers) as depth-weighted mean

mths<-c(10:12)
l<-matrix(c(1:28),ncol=1) # layers for top 300 m

dz<-c(rep(5,20),rep(25,8)) # change in depth for each layer - e.g. layer 0m has 5m before next layer
dzSum<-sum(dz)

ClimSi<-mean(rast(sapply(fData[mths],function(r){
          dIn<-rast(r,"i_an")
          return(sum(rast(sapply(l,function(l,R,dz){R[[l]]*dz[l]},dIn,dz)))/dzSum)
          })))
writeRaster(ClimSi,filename=paste(dirOut,fOut,sep=""))
plot(ClimSi)

# generate means for each MEASO area

Si_means<-fPolyMeansCellWtd(project(sMEASO,ClimSi),ClimSi,AllVectors=TRUE,wtCellArea=TRUE)
save(Si_means,file="./IO In/Si_means.Rdata")
```

Iron

Winter concentrations for the recently published dissolved iron climatology (Huang, Tagliabue, and Cassar 2022), https://zenodo.org/records/6994318

```
#  double dFe_RF [Longitude, Latitude, Depth, Month]
#            units: nmol L-1
#            FillValue: NaN
#            long_name: Monthly dissolved iron simulated from random forest algorithm
#            coordinates: [Longitude, Latitude, Depth, Month]
#     4 dimensions:
#
#        Longitude  Size:357
#            units: degree_north
#            long_name: Longitude
#
#        Latitude  Size:147
#            units: degree_east
#            long_name: Latitude
#
#        Depth  Size:31
#            units: meter
#            long_name: Depth
#
#        Month  Size:13
#           Units: "Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","De#  c",
#                      "Annuual mean"
#           long_name: Month
#
# 4 global attributes:
#        Authors: Yibin Huang & Nicolas Cassar
#        Corresponding author: nicolas.cassar@duke.edu
#        Request_for_citation: If you use these data in publications or presentations, #please cite:
#        "Huang, Y., Tagliabue, A., & Cassar, N. (2022). Data-driven modeling of dissolved iron in
#        the global ocean. Frontiers in Marine Science".

dirData<-"/Users/acon/Desktop/_w/_d/Ocean/Iron/"
dirOut<-"./tif/"
fOut<-"Climatology_Fe.tiff"

fData<-"Monthly_dFe_V2.nc"

rFe<-rast(paste(dirData,fData,sep=""))

# for each month, calculate mean concentration in top 300 m (top 29 layers) as depth-weighted mean

mths<-c(10:12) # month 13 is the average for the year
l<-matrix(c(1:18),ncol=1) # layers for top 300 m
Z<-c( 11.0
     ,25.00021076202393
     ,35.00164031982422
     ,45.00521087646484
     ,55.01295471191406
     ,65.02882766723634
     ,75.06052398681639
     ,85.12305450439452
     ,95.24561309814453
     ,105.4850616455078
     ,115.9520111083985
     ,126.8614730834961
     ,138.6305541992187
     ,152.0652770996094
     ,168.7119445800782
     ,191.4947052001952
     ,225.7935028076171
     ,281.0021057128906
     ,371.9920349121094
     ,518.4190368652341
     ,738.7690429687499
     ,1040.19091796875
     ,1413.504516601562
     ,1839.507568359375
     ,2298.977294921875
     ,2777.8115234375
     ,3267.250732421875
     ,3762.322998046872
     ,4260.339355468752
     ,4759.881347656251
     ,5260.210449218748)
nZ<-length(Z)
dZ<-Z*0
dZ[1]<-11
for(i in 2:nZ) dZ[i]<-(Z[i]-Z[(i-1)])-dZ[(i-1)]
dZ<-dZ*2


dzSum<-sum(dz[l]) # sum of the depths included in the weighting

ClimFe<-mean(rast(
                  sapply(mths,function(m,rFe,dZ,dzSum,l){
                          return(sum(rast(sapply(l,function(l,m,R,dz){R[[(m-1)*nZ+l]]*dz[l]},m,rFe,d
                          },rFe,dZ,dzSum,l)
                  ),na.rm=TRUE)
writeRaster(ClimFe,filename=paste(dirOut,fOut,sep=""))
print(plot(ClimFe))

# generate means for each MEASO area

ClimFe<-rast("./tif/Climatology_Fe.tiff")
Fe_means<-fPolyMeansCellWtd(project(sMEASO,ClimFe),ClimFe,AllVectors=TRUE,wtCellArea=TRUE)
save(Fe_means,file="./IO In/Fe_means.Rdata")
```

Environment

Note that the solution will be very dependent on the proportion of the system expected to be exported. You could configure fixed biomasses to anchor the model and estimate export flux or fix export flux and estimate biomasses. They cannot be estimated together!

Polygon attributes

Summarise polygon attributes, including i) Total Area ii) Mean Latitude, which influences day length over the annual cycle.

```
# create raster
r <- rast(ext(-180, 180, -90, -40), res = c(0.25, 0.25), crs = "EPSG:4326")
values(r)<-sort(rep(seq((-90+0.125),(-40-0.125),0.25),1440),decreasing=TRUE)

Lat_means<-fPolyMeansCellWtd(project(sMEASO,r),r,AllVectors=TRUE,wtCellArea=TRUE)
save(Lat_means,file="./IO In/Lat_means.Rdata")
```

Surface Insolation

Photosynthetically active radiation (PAR) is available as mean Einsteins.$m^{-2}d^{-1}$ (https://oceancolor.gsfc.nasa.gov/resources/atbd/par/). This is converted to Watts.$m^{-2}d^{-1}$.

Sea ice attenuation of light: 1.5 m-1 (Perovich 1996)

Surface Solar Insolation (SSI) is available as monthly means from https://data.giss.nasa.gov/seawifs/ Visualisation products of solar insolation are available as daily, 8-day, or monthly means as Watts.$m^{-2}d^{-1}$. Importantly, this data is for the whole Earth rather than stopping at the sea ice edge. The visualisation products are available using the following link to pages for each year: https://neo.gsfc.nasa.gov/view.php?datasetId=CERES_INSOL_M&year=2012

Available from July 2006 to the present. Monthly data were downloaded by hand. Missing monthly files were:

2012-10-01

Note that very high values were recorded in December (greater than 10000). These were deleted by converting cells to NA

Mike Sumner suggests that they could be accessed using latest GDAL package as an external file e.g. "/vsicurl/https://neo.gsfc.nasa.gov/servlet/RenderData?si=1536111&cs=rgb&format=FLOAT.TIFF&width=1440&height=720" Note that this is rendering monthly data where the month is coded as si=1536111. Need to work out how to decode the numbering for dates with format 2011-11-01.

```
fPath<-"/Users/acon/Desktop/_w/_d/Ocean/Insolation/"
fTiff<-list.files(fPath)
dirOut<-"./tif/"
fOut<-"Insolation.tiff"

# files range from 200607 to 202301
# choose start and end file for averaging over years for each month
fFirst<-"CERES_INSOL_M_2010-07-01_rgb_1440x720.FLOAT.TIFF"  # year is substring(f,15,18) month is su
fLast <-"CERES_INSOL_M_2020-06-01_rgb_1440x720.FLOAT.TIFF"

# vector of filenames to use
fTiff<-fTiff[seq(which(fTiff==fFirst),which(fTiff==fLast),1)]

# note missing: 2012-10-01, 2013-08-01

Mths<-sprintf("%02d",seq(1,12,1))
res<-sapply(Mths,function(m,f){
            res<-f[substring(f,20,21)%in%m]
            res1<-rast(paste0(fPath,res))
            values(res1)[values(res1)>1000]<-NA  # excluding very high values which are rare blips
            return(mean(res1,na.rm=TRUE))
            },fTiff)
Insol<-rast(res)
writeRaster(Insol,filename=paste(dirOut,fOut,sep=""))
rm(res)

# generate means for each MEASO area

ClimInsol<-rast("./tif/Insolation.tiff")
Insol_means<-do.call(rbind,lapply(seq(1,length(names(ClimInsol))),function(m,CI){
  fPolyMeansCellWtd(project(sMEASO,CI[[m]]),CI[[m]],AllVectors=TRUE,wtCellArea=TRUE)[,3]
},ClimInsol))
dimnames(Insol_means)<-list(paste0("M",sprintf("%02d",c(1:12))),unlist(values(sMEASO)))
save(Insol_means,file="./IO In/Insol_means.Rdata")
```

Mixed Layer Depth

Use BRAN data

single tif saved with all months as layers note need to convert edge from greenwich meridian to dateline meridian

```
fName<-"./tif/Ocean/rMLD.tif"
nMonths<-12
refMonths<-sprintf("%02d",seq(1,nMonths,1))
rMLD<-rast(fName)
rMLD<-fConvertRasterEdgeFromGreenwhichToDateLine(rMLD) # from BRAN requires converting edge from Gre

MLD_means<-do.call(rbind,lapply(refMonths,function(f,r){
  fPolyMeansCellWtd(project(sMEASO,r),subset(r,f),AllVectors=TRUE,wtCellArea=TRUE)[,3]
},rMLD))
dimnames(MLD_means)<-list(paste0("M",refMonths),unlist(values(sMEASO)))
save(MLD_means,file="./IO In/MLD_means.Rdata")
```

Temperature

BRAN data for current

```
# BRAN data processed on ace-ecostats
# generated monthly raster brick of mean temperature in MLD and below MLD down to 1000m
# tif files copied into temperature directory

fPath<-"./tif/Temp/"
fNames<-list.files(fPath)
# range of files: 01 to 12 (months)
nMonths<-12
fNames<-paste0("rTemp_",sprintf("%02d",seq(1,nMonths,1)),".tif")

Temp_MLD_means<-do.call(rbind,lapply(fNames,function(f){
  # from BRAN requires converting edge from Greenwich meridian to Dateline
  r<-fConvertRasterEdgeFromGreenwhichToDateLine(subset(rast(paste0(fPath,f)),"Temp_MLD"))
  fPolyMeansCellWtd(project(sMEASO,r),r,AllVectors=TRUE,wtCellArea=TRUE)[,3]
}))
dimnames(Temp_MLD_means)<-list(paste0("M",sprintf("%02d",c(1:nMonths))),unlist(values(sMEASO)))
save(Temp_MLD_means,file="./IO In/Temp_MLD_means.Rdata")

Temp_Deep_means<-do.call(rbind,lapply(fNames,function(f){
  # from BRAN requires converting edge from Greenwich meridian to Dateline
  r<-fConvertRasterEdgeFromGreenwhichToDateLine(subset(rast(paste0(fPath,f)),"Temp_Deep"))
  fPolyMeansCellWtd(project(sMEASO,r),r,AllVectors=TRUE,wtCellArea=TRUE)[,3]
}))
dimnames(Temp_Deep_means)<-list(paste0("M",sprintf("%02d",c(1:nMonths))),unlist(values(sMEASO)))
save(Temp_Deep_means,file="./IO In/Temp_Deep_means.Rdata")
```

Sea ice

Notes on sea ice Will Hobbs paper (to come) Changes in season to season persistence -Libera etal 2022 (should lose memory from one summer to next because of depth of mixed layer see Schroeter et al 2023 on spatial coherence in sea ice anomalies since abrupt critical transition (2006)
) Alexander Haumann (AWI) abrupt sea ice ocean transition following on Will Hobbs talk on role of ocean in sea ice transition declines observed previouslyin spring season but now in autumn (growing) season abrupt change in complex systems (Scheffer etal 2009) Importance of subsurface ocean warming (Meehle)

```
# raadtools used to extract satellite sea ice concentration
# generated raster brick of mean sea ice concentration each month

fName<-"./tif/Sea ice/rCICE.tif"
nMonths<-12
refMonths<-sprintf("%02d",seq(1,nMonths,1))
rCICE<-rast(fName)
```

```
CICE_means<-do.call(rbind,lapply(refMonths,function(f,r){
  fPolyMeansCellWtd(project(sMEASO,r),subset(r,f),AllVectors=TRUE,wtCellArea=TRUE)[,3]
},rCICE))
dimnames(CICE_means)<-list(paste0("M",refMonths),unlist(values(sMEASO)))
save(CICE_means,file="./IO In/CICE_means.Rdata")
```

Sea Ice Thickness Fons et al (2023) published gridded monthly estimates of Antarctic sea ice physical properties derived from CryoSat-2 Baseline-D SAR and SARIn data spanning July 2010 through August 2021. Data are generated using the CryoSat-2 Waveform-Fitting method for Antarctic sea ice (CS2WFA). Files include:

CS2WFA_25km_YYYYMM.nc: Monthly mean values gridded on the NSIDC 25km polar stereographic grid. Each file represents 1 month of data (134 files total). Each monthly file contains:

latitude, longitude, and time (in months since January 2000) Grids of retrieved snow depth, snow freeboard, ice freeboard, and sea ice concentration Grids of seasonal snow, sea ice, and seawater density Grids of sea ice thickness and estimated thickness uncertainty, as well as thickness estimated using a 70% threshold retracker and using a zero-ice-freeboard assumption.

Note that the projection of the data is the NSIDC Polar Stereographic at 25km resolution. On the NSIDC web site (https://nsidc.org/data/user-resources/help-center/guide-nsidcs-polar-stereographic-projection), this could have two EPSG codes depending on which projection method - based on WGS1984 (epsg 3976) or based on Hughes 1980 Ellipsoid (epsg 3412)

```
fPath<-"/Users/acon/Desktop/_w/_d/Ocean/SeaIce/Fons et al 2023/"
fNames<-list.files(fPath)
dirOut<-"./tif/Sea ice/"
fOut<-"SeaIceThickness.tiff"

# range of files: 201007 to 202108
# choose start and end file for averaging over years for each month
fFirst<-"CS2WFA_25km_201007.nc"  # year is substring(f,13,16) month is substring(f,17,18)
fLast <-"CS2WFA_25km_202006.nc"

# vector of filenames to use
fNames<-fNames[seq(which(fNames==fFirst),which(fNames==fLast),1)]

Mths<-sprintf("%02d",seq(1,12,1))
res<-sapply(Mths,function(m,f){
            res<-f[substring(f,17,18)%in%m]
            res1<-rast(sapply(res,function(fN){
                   r <- flip(rast(paste0(fPath,fN)), "vertical")
                   set.ext(r, ext(-3950000, 3950000, -3950000, 4350000))
                   set.crs(r, "EPSG:3031")
                   targetgrid <- rast(ext(-180, 180, -90, -40), res = c(0.25, 0.25), crs = "EPSG:432
                   rRes<-project(subset(r,"sea_ice_thickness"), targetgrid, by_util = TRUE)
                   values(rRes)[is.nan(values(rRes))]<-NA
                   rRes
                   }))
            return(mean(res1,na.rm=TRUE))
            },fNames)
SIthickness<-rast(res)
writeRaster(SIthickness,filename=paste(dirOut,fOut,sep=""))

# generate means for each MEASO area

SIthickness<-rast("./tif/SeaIceThickness.tiff")
SIT_medians<-do.call(rbind,lapply(seq(1,length(names(SIthickness))),function(m,SIT){
  fPolyMediansCellWtd(project(sMEASO,SIT[[m]]),SIT[[m]],AllVectors=TRUE,wtCellArea=TRUE)[,3]
},SIthickness))
dimnames(SIT_medians)<-list(paste0("M",sprintf("%02d",c(1:12))),unlist(values(sMEASO)))
save(SIT_medians,file="./IO In/SIT_medians.Rdata")
```

Biota

The main requirement is to estimate nutrient loss from each biomass pool as a rate relative to total biomass consumption (total mass, not nutrient mass) per unit pool biomass (total mass, not nutrient mass).

Modelling environmental impacts on different biota is accommodated in the following ways:

i) **mixed layer depth** affects availability of prey to predators and, in the case of phytoplankton, altering light and nutrient conditions

ii) **sea ice retreat and advance** affects availability of prey production to predators (in time and in space), e.g. exposure of krill larvae to predation, exposure of Emperor penguin young of the year to predation (reduces handling time/increases maximum ingestion rate of predators)

iii) **sea ice extent/concentration** affects foraging time and therefore metabolic requirements of air-breathers, as well as reduces light for production etc.

iv) **rainfall** affects breeding conditions and young of the year survival in land-based colonies, which could be increased non-predation mortality that is lost to the system (feeding shore birds not included here) or adds more to detrital/carcass pools.

v) **iceburg calving & ice shelf collapse** affects iron supply and recruitment rate of silverfish.

Describe different nodes encompassing biology and why they are grouped that way - are broadly grouped into 8 classes based on their behaviour and distribution: 1. nutrients & organic matter 2. protists 3. benthos 4. dispersed pelagos 5. schooling pelagos 6. filtering divers (Baleen whales) 7. divers (flying birds, penguins, and toothed whales. These are grouped into depth strata where most feeding occurs, ice association and prey.) 8. hunters (top predators - leopard seals and killer whales)

These are further divided based on information available and their separation in time, space and feeding type contributing to broad differences in their ecological functions. As a result there is a natural bias to greater detail amongst larger organisms. Nevertheless, greater differentiation is possible within this framework, should such differentiation be required.

There are few regional estimates of abundance that could be used to anchor the model. The approach is to use ratios of abundance where those ratios can be determined. For example, the consumption analyses for risk assessments in 48.1 (Trathan etc) have population estimates of predators which can be used for a "current" set of relationships.

Abundances of biomass pools can be estimated or fixed (with error).

see Pinkerton et al for what he does.

etc. Cross-tabulate environmental changes of different scenarios with how the parameters of models of different biota are altered to reflect the changes.

Primary Producers

Two phytoplankton groups are represented in this model. Diatoms have been divided in some biogeochemical models (e.g. Losa et al 2019) into small and large. However, they are kept as one group because they are dominated by the faster growing, heavily silicious diatoms preferred by krill. The second group includes all other phytoplankton, including Phaeocystis and coccolithopheres, which are smaller and not dependent on silicic acid. This latter group tends to be important when iron is depleted, which also corresponds to when silicic acid is depleted (??check Henley et al 2020).

The environments of phytoplankton are changing through light availability, nutrient supply, and ocean temperature. Changes in these factors are influenced by cloudiness, winds and sea ice.
A great challenge is to model iron and silicic acid limitation. Jess' work estimates a control parameter that encompasses iron and light limitation.

Iron is factored into the Kearney and Oke models and more easily allows incorporation of iron limitation and resupply. (could test sensitivity of outcomes to this - centre the relative calculations on small primary producers and see how everything scales given different levels of iron supply/limitation).

Need to incorporate limitation of silicic acid and how this can control the abundance of diatoms - how might this limitation change (positively or negatively)

Productivity of phytoplankton groups will govern available production in higher trophic levels. How to partition total production between phytoplankton groups. . . .

Maximum Growth rate is governed by temperature. Here I use the Jeffrey formulation for simplicity but has parameters suitable for diatoms and other phytoplankton (notably, Phaeocystis). Maximum growth of diatoms with temperature is the same as that of Hauck et al (2013) and of Phaeocystis is the same as Oke et al (2013) and Melbourne-Thomas et al (2015). The only parameter needed is the maximum growth rate, μ_{max}.

$$\mu(T) = \mu_{max} e^{0.06T}$$

Table 1: Parameters used for primary production in diatoms and other phytoplankton (modelled on haptophytes) in the NPZD model of Base Production

Parameter	Symbol	Value		Units
Light		*Value*		
Attenuation PAR through sea ice	k_{si}	1.5[*]		m^{-1}
Attenuation PAR through water	k_w	0.04[*1]		m^{-1}
Proportion incident radiation photosynthetically active	PAR	0.43[*1]		−
Growth rate		*Diatoms*	*Other*	
Photosynthesis efficiency (initial slope of P-I curve)	α		control[*1]	
Maximum growth rate parameters	μ_{max}	1.44 [*2]	0.63 [*2]	d^{-1}
Half saturation constants for nutrient uptake				
Silicate	K_{SiO_4}	4.0 [*2]	0.0 [*2]	$mmol.m^{-3}$
Dissolved Iron	K_{dFe}	1.0 [*2]	0.1 [*2]	$\mu mol.m^{-3}$
Stoichiometry				
Algal carbon to nitrogen ratio	$R_{c:n}$	7.0 [*2]	7.0 [*2]	$mol.mol^{-1}$
Algal chl-a to nitrogen ratio	$R_{chla:n}$	2.1 [*2]	0.84 [*2]	$g.mol^{-1}$
Algal iron to nitrogen ratio	$R_{fe:n}$	0.023 [*2]	0.7 [*2]	$mmol.mol^{-1}$
Algal silica to nitrogen ratio	$R_{fe:n}$	1.8 [*2]	0.0 [*2]	$mol.mol^{-1}$
			*	
			*	

References: * 1 Melbourne-Thomas et al 2015; * 2 Jeffery et al 2020

Higher Trophic Levels

Ecopath models are used to estimate maximum ingestion rates, vulnerabilities, combined respiration & non-assimilated waste, and non-predatory mortality (disease etc.).

The literature is used to do the following: i) allometry to differentiate respiration from non-assimilated waste, ii) carbon:silica and carbon:iron ratios in each species (used to differentiate waste into different pools) iii)

Two types of Ecopath models are used: a) unadjusted models from the literature (thanks to Roshni and Stacey for compiling) Ballerini 2014 (WAP) Dahood 2019 (WAP) Gurney 2014 (PEI) Maldonado 2016

McCormack 2019 (Prydz Bay) Pinkerton 2010 (Ross Sea) Subramanium 2020 (Kerguelen) Suma 2014 Supernand 2017

b) standardised models from Hill et al 2021

General parameters estimated from existing Ecopath models.
Outline the estimation and the results.

Incorporating impacts on recruitment
Marine mammals and birds - Incorporate environmental impact on young of the year - related to condition of adults (breeding condition - irrelevant in equilibrium model as density dependent) and mortality of young (see papers by Jenouvrier on this).
Recruitment related to perimeter of ice-edge in November. Approximate this as the square-root of sea ice >90% concentration

Consumption

??revise this text to be deriving model parameters. move rest of text to the model

, most often as Holling disc/Type II response. Type III responses usually relate to predators that have difficulty feeding on low densities or prey. Type I responses relate to consumption of resources solely according to their density without any limitation. These functional responses relate a single predator to a single prey, yet experts acknowledge the need for multispecies functional responses, including different alternative prey, the potential for interference by predators and the possibility of higher predators influencing predation of lower trophic level predators. As yet, there is no formal mathematical formulation of multispecies functional responses (Abrams 2022).

At equilibrium, the functional relationships of predators and prey in an assemblage are distilled into a "diet matrix" of the proportions of total consumption by a predator to which each species contributes. This matrix underpins the formulation of equilibrial food webs in Ecopath. However, the diet matrix is unlikely to be true outside of that equilibrium when the forces of foraging ecology are at work. Moreover, a diet matrix will constrain the solution of relative abundances in the biomass pools of an equilibrial food web, irrespective of other pressures that may be influencing the food web.

What is a suitable approximation of multi-species functional relationships that can account for the diet matrix at a specified equilibrium but will enable flexibility in the solution for a food web under conditions different from that equilibrium?

For pelagic systems, the probability of encountering suitable prey (e.g. based on suitable size range) is a good starting point, as in size-based food-web modelling (Blanchard et al 2010). Can all potential prey be considered to be available as a pool, with consumption according to relative abundances? Is encounter probability based on presence/absence alone or includes relative size of individuals? How does the probability of encounter change between pools, such as through swarming by krill? (see Maury et al)

A simplified version of a consumption model is used to estimate parameters for a food web estimated using Ecopath. While Ecosim also estimates model parameters, a general version taking account of different size-based formulations is used here. The aim here is to provide flexibility in estimating different potential equilibrial network relationships under different conditions rather than precisely specifying different parameters for a long-term simulation. For that reason, a Holling Type II functional relationship is used with a multi-species approach based on time spent foraging on individual species.

The basic model of consumption, Q, of prey,i, by predator,j, is

$$Q'_{i,j} = B_j \frac{c_{i,j}}{\sum_{i'} c_{i',j}} g_{i,j}(N_i, \tilde{I}_{i,j}) EE_i \left(\frac{P_i}{B_i}\right) \tag{1}$$

where $c_{i,j}$ is the selectivity of a prey item, B is the biomass of the predator, N is the biomass of prey and g is the per biomass functional relationship determining the ingestion rate of that prey species if no other prey types are available. The amount consumed is that which maintains equilibrium, which is the production arising from that biomass.

A Holling Type II function response is used here with handling time per prey biomass being inversely proportional to the maximum ingestion rate relative to the standing stock (Christensen et al 2008), such that

$$g_{i,j}(N_i, \tilde{I}_{i,j}) = \frac{a_{i,j} N_i}{1 + \frac{a_{i,j}}{\tilde{I}_{i,j}} N_i} \tag{2}$$

For completeness, the maximum ingestion rate per predator biomass per time is the product of the standing stock ingestion rate and the production to biomass ratio of the prey discounted by the ecotrophic efficiency, although this is not used in the calculations.

$$\hat{I}_{i,j} = \tilde{I}_{i,j} \left(\frac{P_i}{B_i} \right) EE_i \tag{3}$$

Availability of prey to predators

Availability of prey to predators is different from the selectivity function, which is where shifting preferences may be included. Here, availability represents the proportion of the prey that is available to a predator. This can involve geographic separation as well as separation by depth. Geographic separation could relate, for example, by the proportions of a population associated with sea ice or with the continental shelf.

Availability determined by overlap in pelagic depth strata is the product of the probabilities of predator and prey in the area of overlap, such that

$$A_{i,j} = \int_{seafloor}^{surface} \mathrm{p}(B_i, z)\mathrm{p}(B_j, z) dz$$

For the present, I assume a uniform probability distributions of predators and prey within their depth ranges.

```
calcAvailabilityMatrixFromDepthRange<-function(D){ # two column matrix - rows = nodes, cols = top,bo
sapply(seq(1:nrow(D)),function(p,D){
pD<-D[p,]
apply(D,1,function(nD,pD){
  Os<-min(nD[1],pD[1])
  Od<-max(nD[2],pD[2])
  return(if(Od<Os) (Os-Od)^2/((nD[1]-nD[2])*(pD[1]-pD[2])) else 0)
},pD)
},D)
}# end calcAvailabilityMatrixFromDepthRange

DepthRanges<-as.matrix(eMod$params[,c("DepthTop","DepthBottom")])
PreyAvail<-calcAvailabilityMatrixFromDepthRange(DepthRanges)
print(PreyAvail)
```

Per predator biomass ingestion rate of prey

Ingestion rates of prey by predators is governed by the abundance and productivity of prey, the vulnerability of prey and the ingestion rates of predators given the available prey. Their co-dependence makes reliable estimation more difficult. Nevertheless, the vulnerability and functional relationships to be used in the steady-state model can be approximately estimated from Ecopath results.

Using terminology for Ecopath inputs and outputs, the equations to be solved are based on the predator-prey consumption matrix such that the total consumption of a prey, i, across all predators, $\sum_j$, is equal to its production discounted by the ecotrophic efficiency:

$$\sum_j \left[d_{i,j} \left(\frac{Q_j}{B_j} \right) B_j \right] = EE_i \left(\frac{P_i}{B_i} \right) B_i \tag{4}$$

In a steady-state, the consumption by predators can be represented by the model equations, such that:

$$\left(\frac{Q_j}{B_j}\right) B_j = B_j \sum_i \left[g\left(\vec{B}, \vec{s}_j, \vec{I}_j\right)_{i,j} I_{i,j} \right] \tag{5}$$

In order to give equal weight to all pools, the following equation could be minimised when vulnerability can be estimated externally:

$$SS(\vec{I}) = \sum_i \left(1 - \frac{\sum_j d_{i,j} B_j \sum_i \left[g\left(\vec{B}, \vec{v}_j, \vec{I}_j\right)_{i,j} I_{i,j} \right]}{EE_i \left(\frac{P_i}{B_i}\right) B_i} \right)^2 \tag{6}$$

However, diet information is rarely sufficient for estimating vulnerability of prey to predators.

Instead, vulnerability and maximum ingestion rate can be combined in the estimation if the maximum ingestion rate is the maximum ingestion rate for a predator across all prey and the vulnerability then relates to the combined availability of prey with the expected production of prey, independently of other predators. Thus,

$$SS_j == \left[g\left(\vec{B}, \vec{v}_j, I_j\right)_{i,j} I_j - d_{i,j} \left(\frac{Q_j}{B_j}\right) \right]^2 \tag{7}$$

```r
# read in Ecopath
library(PopED)

fnReadEcopath<-function(root,fname){
  names<-read.csv(paste(root,fname,"_names.csv",sep=""),header=TRUE)
  params<-read.csv(paste(root,fname,"_params.csv",sep=""),header=TRUE)
  diet<-read.csv(paste(root,fname,"_diet.csv",sep=""),header=FALSE)
  return(list(names=names,params=params,diet=diet))
     }

calcPredConsume<-function( # returns vector of prey consumed
                 I   # scalar or vector - max ingestion rates for each prey of predator
                ,Br  # vector - biomasses of prey
                ,v   # vector - vulnerability of prey to predator
                ,Bp  # scalar - Biomass of predator
                ){
                return(sum(Holling2(Br,I,v)*I)*Bp)
                }
Holling2<-function(   # returns vector proportion of max ingestion consumed
                   Br # vector - resource (prey) biomasses
                  ,I  # scalar
                  ,v  # vector - vulnerability of prey to predator
                  ){
  mask<-v>0
  res<-v*0
  sum_vBr<-sum(v[mask]*Br[mask])
  res[mask]<-v[mask]*Br[mask]/(I+sum_vBr)
  return(res)
  }

fnCheckEcoPathFit<-function(B,QB,PB,d,EE){
       n<-length(B)
       pool<-seq(1,n,1)
       res<-lapply(pool,function(i,B,QB,PB,d,EE){
              left<-sum(unlist(
                    sapply(pool,function(j,i,d,B,QB)
                      {d[i,j]*B[j]*QB[j]},i,d,B,QB)
                    )) # end unlist sum
                   right<-EE[i]*PB[i]*B[i]
                   return((left-right)/right)
                 },B,QB,PB,d,EE)
       return(unlist(res))
 } # end check function

#     Libraries to support optimisation ####

#     Root function for minimisation routine ####

rootFnIhat<-function(par,B,QB,d,useLog){
        I<-par[1]
        v<-par[-1]
        res<-v*B
        sumRes<-sum(res)
        if(useLog) {res<-sum( (log(I*res/(I+sumRes)) - log(d*QB))^2)} else
        {res<-sum( (I*res/(I+sumRes) - d*QB)^2)}
        return(res)
        } # end find Ihat

#     Other functions to support minimising I,v ####

trialV<-function(v,I,iB,iQB,id,iUseLog,boundLo,boundHi){
  resI<-NULL
  resV=NULL
  resValue=NULL
  # unadjusted
  Vstart<-c(I,rep(v,length(id)))
res<-optim_ARS(par=Vstart,fn=rootFnIhat,lower=boundLo,upper=boundHi,iter=10000,iter_adapt=10,trace=F
  resI<-c(resI,res$par[1])
  resV<-rbind(resV,res$par[-1])
  resValue<-c(resValue,res$ofv)

    # adjust Iup
  Vstart<-c(resI[1]*1.5,resV[1,])
res<-optim_ARS(par=Vstart,fn=rootFnIhat,lower=boundLo,upper=boundHi,iter=10000,iter_adapt=10,trace=F
  resI<-c(resI,res$par[1])
  resV<-rbind(resV,res$par[-1])
  resValue<-c(resValue,res$ofv)

    # adjust Idown
  Vstart<-c(resI[1]*0.5,resV[1,])
res<-optim_ARS(par=Vstart,fn=rootFnIhat,lower=boundLo,upper=boundHi,iter=10000,iter_adapt=10,trace=F
  resI<-c(resI,res$par[1])
  resV<-rbind(resV,res$par[-1])
  resValue<-c(resValue,res$ofv)

    # adjust inverse V
  Vstart<-c(resI[1],(1-resV[1,]))
res<-optim_ARS(par=Vstart,fn=rootFnIhat,lower=boundLo,upper=boundHi,iter=10000,iter_adapt=10,trace=F
  resI<-c(resI,res$par[1])
  resV<-rbind(resV,res$par[-1])
  resValue<-c(resValue,res$ofv)
```

```
      # adjust invert last V
  Vstart<-c(res$par[1]*0.5,(1-res$par[-1]))
res<-optim_ARS(par=Vstart,fn=rootFnIhat,lower=boundLo,upper=boundHi,iter=10000,iter_adapt=10,trace=F
  resI<-c(resI,res$par[1])
  resV<-rbind(resV,res$par[-1])
  resValue<-c(resValue,res$ofv)

  return(list(name=c("unadj","Iup","Idown","inverseV","lastInvert"),I=resI,v=resV,value=resValue))
} # end function

minV<-function(r){
   minR<-which(r$value==min(r$value))
   return(list(name=r$name[minR],I = r$I[minR],v = r$v[minR,],value = r$value[minR]))
    } # return minimum result from list of results

searchV<-function(iB,iQB,id,iUseLog,boundLo,boundHi){
  res<-lapply(seq(0.1,0.9,0.1),trialV,iQB,iB,iQB,id,iUseLog,boundLo,boundHi)
  resMin<-unlist(lapply(res,function(r){
    r1<-which(unlist(r$value)==min(unlist(r$value) ))
    return(r1[1])}))
  resI<-unlist(sapply(seq(1,length(resMin),1),function(i,r,rMin){r[[i]]$I[rMin[i]]},res,resMin))
  resV<-do.call(rbind,lapply(seq(1,length(resMin),1),function(i,r,rMin){r[[i]]$v[rMin[i],]},res,resM
  resValue<-unlist(sapply(seq(1,length(resMin),1),function(i,r,rMin){r[[i]]$value[rMin[i]]},res,resM
  resName<-unlist(sapply(seq(1,length(resMin),1),function(i,r,rMin){r[[i]]$name[rMin[i]]},res,resMin
  return(list(name=resName,I=resI,v=resV,value=resValue))
} # end function

fnCheckEstI_V<-function(I              # estimated I vector
                        ,v             # estimated v matrix
                        ,d             # diet matrix
                        ,B             # vector of biomasses
                        ,QB            # vector - consumption by each pool
                 ){
                 n<-length(QB)
                 pool<-seq(1,n,1)
                 # vector of total consumption by predators
                dprime<-lapply(pool,function(j,I,v,d,B,QB){
                         if(I[j]==0) return(v[,j]*0)
                         res<-Holling2(B,I[j],v[,j])*I[j]
   res<-round(((res-d[,j]*QB[j])/(d[,j]*QB[j]))*100,0)
   res[is.nan(res)]<-0
                         return(res)
                         },I,v,d,B,QB)
                 dprime<-do.call(cbind,dprime)

                 return(dprime)
 } # end check function

fnCalcI<-function(i          # target prey species
                  ,B        # vector of biomasses of all taxa
                  ,v         # vulnerabilities of biomasses to predator
                  ,QBj       # consumption per biomass of predator
                  ){sum(v*B)/(v[i]*B[i]/QBj-1)}
```

```
root<-"/Users/acon/Desktop/_w/_p/SOfoodweb & EcoVelocity/Ecosystem network synthesis/Ecopath files/R

# Inputs for Ecopath Analysis ####
```

```r
units<-365 # time units: 1=year, 365=days

iUseLog<-FALSE # log transform in root function

Ecopath_Mods<-c("Ballerini2014","Ballerini2014_Stand","Dahood2019","Gurney2014_Yr2000","Gurney2014_S


# designate Ecopath model in order to generalise code ####

eMod<-fnReadEcopath(root,Ecopath_Mods[1])

# diet - columns are the proportion diet composition for a consumer
# check diet adds to 1 for each predator (not always the case) ####

apply(eMod$diet,2,sum,na.rm=TRUE)

# Prepare data ####

Npool<-nrow(eMod$params)

In_B<-eMod$params[,"B"]
In_PB<-eMod$params[,"P.B"]
In_PB[is.na(In_PB)]<-1  # consume biomass even though no production occurs (e.g. detritus)
In_QB<-eMod$params[,"Q.B"]
In_QB[is.na(In_QB)]<-0  # consume biomass even though no production occurs (e.g. detritus)
In_EE<-eMod$params[,"EE"]

Diet<-eMod$diet
Diet[is.na(Diet)]<-0

# Check how well Ecopath model is consistent in diet matrix ####

fnCheckEcoPathFit(In_B,In_QB,In_PB,PredDiet$d,In_EE)

# estimate vulnerability matrix (v) and predator max ingestion rates (I) ####


#     test on a single predator ####
tgt<-2
units<-365 # time units: 1=year, 365=days
mask<-Diet[,tgt]>0
iB<-In_B[mask]
iQB<-In_QB[tgt]/units
id<-Diet[mask,tgt]
#iUseLog<-FALSE # log transform in root function

Vstart<-c(iQB/2,rep(0.9,length(id)))
boundLo<-1E-10
boundHi<-c(Inf,rep(1,length(id)))  # does not like Inf in vector but if it was to be changed to 1 th
rootFnIhat(Vstart,B=iB,QB=iQB,d=id,useLog=iUseLog)

Vstart<-c(iQB/2,rep(0.2,length(id)))

res<-optim_ARS(par=c(iQB/2,rep(0.5,length(id))),fn=rootFnIhat,lower=boundLo,upper=boundHi
               ,iter=10000,iter_adapt=10,trace=FALSE,B=iB,QB=iQB,d=id,useLog=iUseLog)
```

```
res<-trialV(0.1,iQB,iB,iQB,id,iUseLog,boundLo,boundHi)
res<-searchV(iB,iQB,id,iUseLog,boundLo,boundHi)
res1<-minV(res)
```

```
#     minimisation ####
# and saving of ingestion rates and vulnerability matrix given the ecopath model

res<-do.call(cbind,lapply(seq(1,Npool,1),function(j,iB,iQB,Diet,units,iUseLog){
      mask<-Diet[,j]>0
      B  <- iB[mask]
      QB <- iQB[j]/units
      d  <- Diet[mask,j]
      if(sum(mask)==0) return(c(QB,Diet[,j]))
      boundLo <- c(1E-10,rep(1E-10,length(d)))
      boundHi <- c(Inf,rep(1,length(d)))
      res<-minV(searchV(B,QB,d,iUseLog,boundLo,boundHi))
      v<-Diet[,j]*0
      v[mask]<-res$v
      return(c(res$I,v))
     },In_B,In_QB,Diet,units,iUseLog))
estI<-res[1,]
estV<-res[-1,]

save(estI,estV,file="IO in/estI_V_ordinaryLS.Rdata")

# check diet matrix ####

chk<-fnCheckEstI_V(estI,estV,Diet,In_B,In_QB/units)
pchk<-unlist(chk)
pDiet<-unlist(Diet)
mask<-pchk!=0
plot(pDiet[mask],pchk[mask])
# check that most of the wayward estimates are very low proportions in the diet with very little wei
```

Non-Predation mortality from Ecopath data Non-predation mortality rate per biomass (in the time units of production rate) of a species in Ecopath is the loss of production not due to predation, the latter of which is explained by the ecotrohic efficiency.

$$m0_j = (1 - EE_j)\left(\frac{P_j}{B_j}\right)\ m.t^{-1}$$

```
m0pd<-(1-unlist(eMod$params[,"EE"]))*unlist(eMod$params[,"P.B"])/365
```

As a rate, t^{-1} relative to biomass

$$M0_j = -log\left[1 - \left(\frac{P_j}{B_j}\right)(1 - EE_j)\right]\ t^{-1}$$

```
M0pd<--log(1-m0pd)
```

Respiration and Non-assimilated Mass Proportion of consumption that is either defaecated (not assimilated) or respired/excreted.

$$RNA_j = \frac{(Q_j - P_j)}{Q_j}$$

```
RNA<-(unlist(eMod$params[,"Q.B"])-unlist(eMod$params[,"P.B"]))/unlist(eMod$params[,"Q.B"])
```

```
# code brought down from above and run across all models

root<-"/Users/acon/Desktop/_w/_p/SOfoodweb & EcoVelocity/Ecosystem network synthesis/Ecopath files/R

Ecopath_Mods<-c("Ballerini2014")

# Ecopath_Mods<-c("Ballerini2014","Ballerini2014_Stand","Dahood2019","Gurney2014_Yr2000","Gurney2014


units<-365
iUseLog=FALSE

resEcopath<-lapply(Ecopath_Mods,function(EcoMod){
  EcoMod<-Ecopath_Mods[1]
  print(EcoMod)
  eMod<-fnReadEcopath(root,EcoMod)
  checkPredTotalDiet<-apply(eMod$diet,2,sum,na.rm=TRUE)
  Npool<-nrow(eMod$params)

  In_B<-eMod$params[,"B"]
  In_PB<-eMod$params[,"P.B"]
  In_PB[is.na(In_PB)]<-1
  In_QB<-eMod$params[,"Q.B"]
  In_QB[is.na(In_QB)]<-0
  In_EE<-eMod$params[,"EE"]
  Diet<-eMod$diet
  Diet[is.na(Diet)]<-0

  checkEcopathFit<-fnCheckEcoPathFit(In_B,In_QB,In_PB,Diet,In_EE)
  res<-do.call(cbind,lapply(seq(1,Npool,1),
                    function(j,iB,iQB,Diet,units,iUseLog){
        print(paste0(EcoMod," Pool ",j))
                    mask<-Diet[,j]>0
                    B  <- iB[mask]
                    QB <- iQB[j]/units
                    d  <- Diet[mask,j]
                    if(sum(mask)==0 | sum(B>0)==0) return(c(QB,Diet[,j]))                    boundHi
                    r1<-minV(searchV(B,QB,d,iUseLog,boundLo,boundHi))
                    if(length(r1$I)>1){
                          v<-Diet[,j]*0
                          v[mask]<-rep(1,length(d))
                          rI<-fnCalcI(which(mask)[1]        # target prey species
                                      ,iB      # vector of biomasses of all taxa
                                      ,v       # vulnerabilities of biomasses to predator
                                      ,QB     # consumption per biomass of predator
                                      )
                          r2<-c(rI,v)  # need to check if applied to more than one prey
                      print("Minimum of Search returned more than one result.  Vulnerabilities retur
                      print(r1$v)

                    } else {
                    v<-Diet[,j]*0
                    v[mask]<-r1$v
                    r2<-c(r1$I,v)
                    }
                    return(r2)
                    },In_B,In_QB,Diet,units,iUseLog))
  print(res)
  estI<-res[1,]
  estV<-res[-1,]
  print(estV)
  mOpd<-(1-unlist(eMod$params[,"EE"]))*unlist(eMod$params[,"P.B"])/units
  MOpd<--log(1-mOpd)
  RNA<-(unlist(eMod$params[,"Q.B"])-unlist(eMod$params[,"P.B"]))/unlist(eMod$params[,"Q.B"])

EcopathEst<-list(
             PoolN = Npool
            ,checkPredTotalDiet = checkPredTotalDiet
            ,checkEcopathFit = checkEcopathFit
            ,IngestionMax = estI
            ,v_matrix = estV
            ,NonPredMortRate = MOpd
            ,Resp_NonAssim = RNA
            )
save(EcopathEst,file=paste0("EcopathEst_",EcoMod,"_day_OLS.Rdata"))
return(EcopathEst)
})

names(resEcopath)<-Ecopath_Mods
save(resEcopath,file="EcopathEstimation_day_OLS.Rdata")
```

Processing Ecopath data difficulty with Hill et al aggregations is that there is no differentiation between krill and fish pathways in the species pools. e.g. fish-eating penguins versus krill-eating penguins.

Boyer, T. P., O. K. Baranova, C. Coleman, H. E. Garcia, A. Grodsky, R. A. Locarnini, A. V. Mishonov, et al. 2018. "World Ocean Database 2018. A. V. Mishonov, Technical Editor, NOAA Atlas NESDIS 87." Report.

Huang, Yibin, Alessandro Tagliabue, and Nicolas Cassar. 2022. "Data-Driven Modeling of Dissolved Iron in the Global Ocean." Journal Article. *Frontiers in Marine Science* 9. https://doi.org/10.3389/fmars.2022.837183.

Perovich, D. K. 1996. "The Optical Properties of Sea Ice. Monograph 96-1." Report. Cold Regions Research & Engineering Laboratory, US Army Corps of Engineers.