

```

=====;
; tag_emissions.ncl
=====;
;
; Concepts illustrated:
; - load CAM emission files
; - create zero emission everywhere except desired regions
; - output new file as netCDF
;
;
; Acronym      Description
; -----
; AUST         Australasia
; EQAS         Indonesia and Equatorial Asia
; SEAS         Southeast Asia
; INDI         India
; NEAS         Northeast Asia
; SIBR         Siberia
; WRUS         West Russia
; EURO         Europe
; AFME         Northern Africa and Middle East
; NHAf         Northern Africa
; SHAF         Southern Africa
;              South America
;              Central America
;              North America
;              Southern Hemisphere
;              Northern Hemisphere
;
; To run type:
; ncl 'efile_in="<path_to_file>"' tag_emissions.ncl
;
; Requires:
; input emission file
;
;                               RRB March 17, 2015
;                               RRB major edits 20180831
=====;
; load the libraries

begin

; -----
; user input
; -----
tracer = "CO"
type = "fire"

; -----
; emission files
; -----

top_dir = "/glade/work/buchholz/emis/"
if (.not. isvar("efile_in")) then          ; is NRT=<boolean> from command line?
    efile_in = top_dir + "qfed2.5_2014_2017_1x1/" + \
        "qfed.emis_" + tracer + "_0.9x1.25_mol_2014_2017.nc"
end if
print("Processing: " + efile_in)

outpath = top_dir + "tagged_emis/"

; -----
; toggles
; -----
PLOT          = False
PLOTREGIONS   = True
netCDF         = False                      ; for netCDF output
; -----

```

```

; end user input
;-----

;*****
; set up
; -----
; names of data structures
; -----
if (type.eq."fire") then
    emiss = "bb"
else if (type.eq."anthro") then
    emiss = "anthro"
else if (type.eq."ocean") then
    emiss = "ocean"
end if
end if
end if

; -----
; plotting parameters
; -----
if (PLOT) then
    mapType = "x11"
    mapName = tracer+"_region"
end if ;PLOT

if (PLOTREGIONS) then
    mapType2 = "x11"
    mapName2 = tracer+"_regions"
end if ;PLOT

cr = "~C~"

; -----
; Define regions
; -----
region_names = (/ "AUST", "EQAS", "SEAS", "INDI", "NEAS", "SIBR", \
    "BOAS", "EURO", "AFME", "NHAF", "SHAF", "BONA", \
    "TENA", "CEAM", "NHSA", "AMAZ", "SHSA", "NH", "SH" /)

;
; minlat, maxlat, minlon, maxlon
region_select = (/ (-48., -10., 110., 181.), \ ;AUST
    (/ -10., 8., 90., 160.), \ ;EQAS
    (/ 8., 28., 90., 135.), \ ;SEAS
    (/ 4., 36., 60., 90.), \ ;INDI
    (/ 28., 50., 90., 153.), \ ;NEAS
    (/ 50., 80., 90., 190.), \ ;SIBR
    (/ 36., 80., 40., 90.), \ ;BOAS
    (/ 36., 75., 360.-25., 40.), \ ;EURO
    (/ 0., 36., 33., 60.), \ ;AFME
    (/ 0., 36., 360.-18., 33.), \ ;NHAF
    (/ -36., 0., 0., 55.), \ ;SHAF
    (/ 50., 75., 360.-170., 360.-53.), \ ;BONA
    (/ 27., 50., 360.-140., 360.-53.), \ ;TENA
    (/ 9., 27., 360.-120., 360.-40.), \ ;CEAM
    (/ 0., 9., 360.-90., 360.-30.), \ ;NHSA
    (/ -25., -0., 360.-90., 360.-30.), \ ;AMAZ
    (/ -60., -25., 360.-90., 360.-30.), \ ;SHSA
    (/ 0., 90., 0., 360.), \ ;NH
    (/ -90., 0., 0., 360.) /) ;SH

;
; region_names := (/ "KORE", "JAPN", "BONA", "TENA", "AMAZ", "AUST", \
; "SEAS", "INDI", "CEAS", "CEAM", "NEAS", "BOAS", \
; "EURO", "EQAS", "NHAF", "SHAF", "NH", "SH" /)
; Ben's DEFS
; minlat, maxlat, minlon, maxlon
; region_select := (/ (33.5, 38., 125., 129.), \ ;KORE

```

```
;
;      (/ 30., 44.,      130.,      146./),\ ;JAPN
;      \;(/ 18., 50.,      95.,      135.5/),\ ;EastAsia ;
;      \;(/ 20., 70.,      192.,      300./),\ ;NAM
;      \;(/ 23., 50.,      192.,      300./),\ ;CONUS
;      \;(/ 50., 70.,      192.,      300./),\ ;Canada
;      (/ 50., 75.,      192.,      310./),\ ;BONA
;      (/ 27., 50.,      192.,      310./),\ ;TENA
;      (/ -25., 8.5,      275.,      325./),\ ;AMAZ
;      (/ -45., -9.,      110.,      160./),\ ;AUST
;      (/ 7., 28.,      91.,      135.5/),\ ;SEAS
;      (/ 7., 32.,      60.,      90./),\ ;INDI
;      (/ 29., 49.,      91.,      135.5/),\ ;CEAS
;      (/ 9., 27.,      240.,      310./),\ ;CEAM
;      (/ 39., 49.,      91.,      129.5/),\ ;NEAS
;      (/ 50., 75.,      40.,      190./),\ ;BOAS
;      (/ 30., 75.,      335.,      40./),\ ;EURO
;      \;(/ -10., 8.,      95.,      120./),\ ;Indonesia
;      (/ -10., 6.,      95.,      160./),\ ;EQAS
;      \;(/ -45., -10.,      110.,      162./),\ ;Australasia
;      \;(/ -35., 30.,      335.,      50./),\ ;Africa
;      (/ 0., 30.,      335.,      50./),\ ;NHAF
;      (/ -35., 0.,      335.,      50./),\ ;SHAF
;      \;(/ -22., 8.5,      275.,      325./),\ ;Amazon
;      (/ 0., 90.,      0.,      360./),\ ;NH
;      (/ -90., 0.,      0.,      360./) /) ;SH
```

```
n_regions = dimsizes(region_select(:,0))
```

```
;*****
```

```
; load files, extract and tag
```

```
;*****
```

```
fin = addfile(efile_in, "r")
```

```
emiss_in = fin->$emiss$(0:2, :, :)
```

```
; separate into regions
```

```
; loop through regions
```

```
do ireg = 0, n_regions-1
```

```
    region = region_names(ireg)
```

```
    latmin = region_select(ireg,0)
```

```
    latmax = region_select(ireg,1)
```

```
    lonmin = region_select(ireg,2)
```

```
    lonmax = region_select(ireg,3)
```

```
; -----
; determine indices for outside regions and set to zero
```

```
if (region.eq."NH") then
```

```
    ; Latitudes outside region
```

```
    ind_lat_region_2 := ind(emiss_in&lat.le.latmin)
```

```
    emiss_region := emiss_in
```

```
    ; overwrite outside region with zeros
```

```
    emiss_region(:, ind_lat_region_2, :) = 0
```

```
else if (region.eq."SH") then
```

```
    ; Latitudes outside region
```

```
    ind_lat_region_1 := ind(emiss_in&lat.ge.latmax)
```

```
    emiss_region := emiss_in
```

```
    ; overwrite outside region with zeros
```

```
    emiss_region(:, ind_lat_region_1, :) = 0
```

```
else
```

```
    ; Latitudes outside region
```

```
    ind_lat_region_1 := ind(emiss_in&lat.ge.latmax)
```

```
    ind_lat_region_2 := ind(emiss_in&lat.le.latmin)
```

```
    ; Longitudes outside region
```

```
    ind_lon_region_1 := ind(emiss_in&lon.le.lonmin)
```

```

ind_lon_region_2 := ind(emiss_in&lon.ge.lonmax)
; issues because Africa spans 360
if (region.eq."NAfrica".or.\
    region.eq."SHAF".or.\
    region.eq."Europe") then
    ind_lon_region_1 := ind(emiss_in&lon.le.lonmin.and.emiss_in&lon.ge.lonmax)
    ind_lon_region_2 := ind(emiss_in&lon.ge.lonmax.and.emiss_in&lon.le.lonmin)
end if

emiss_region := emiss_in
; overwrite outside region with zeros
;emiss_region(:,ind_lat_region_1,:) = 0
;emiss_region(:,ind_lat_region_2,:) = 0
;emiss_region(:,ind_lon_region_1) = 0
;emiss_region(:,ind_lon_region_2) = 0
end if
end if

; -----
; collect region outlines for plotting later
if (ireg.eq.0) then
    region_lon = new(/1,5/),float)
    region_lat = region_lon
    region_lon(0,:) = (/lonmin, lonmin, lonmax , lonmax, lonmin/)
    region_lat(0,:) = (/latmin, latmax, latmax , latmin, latmin/)
else
    temp1:= region_lon
    temp2:= region_lat
    dum1 := new(/1,5/),float)
    dum2 := dum1
    dum1(0,:) = (/lonmin, lonmin, lonmax , lonmax, lonmin/)
    dum2(0,:) = (/latmin, latmax, latmax , latmin, latmin/)
    region_lon := array_append_record(temp1,dum1,0)
    region_lat := array_append_record(temp2,dum2,0)
end if

;*****
; Write out to an netCDF file
if (netCDF) then
;-----
; create file for each region
;-----
    outfilename = "rcp85_2000-2014_"+tracer+"_"+region+"_noBiogBB_1.9x2.5_mol_c20150313.nc"
    ; copy file to new file
    system("cp "+efile_in+" "+outpath+outfilename)
    ; open the new file to overwrite
    fout = addfile(outpath+outfilename, "w")

;-----
; File
; Attributes
;-----
; will have same file attributes as original emission file
; overwrite the title and date
    fout@creation_date = systemfunc ("date")
    fout@title = "Emissions of "+ tracer+" over "+region+\
        " from FINN emission ratios applied to QFED C02, daily"

    setfileoption(fout, "DefineMode",False)

;-----
; Variables
;-----
emiss_out := emiss_region
copy_VarAtts(emiss_in,emiss_out)
copy_VarCoords(emiss_in,emiss_out)

```

```

    emiss_out@long_name = tracer+" "+type+" emissions over "+region
    emiss_out@vmax      = max(emiss_out)
    emiss_out@vmin      = min(emiss_out)
    printVarSummary(emiss_out)

    copy_VarAtts(oc_emiss_in,oc_emiss_out)
    copy_VarCoords(oc_emiss_in,oc_emiss_out)
    oc_emiss_out@long_name = "zeroed ocean emissions"
    oc_emiss_out@vmax      = max(oc_emiss_out)
    oc_emiss_out@vmin      = min(oc_emiss_out)
    printVarSummary(oc_emiss_out)

    ;--- overwrite emissions
    fout->$emiss$          = emiss_out

end if ;NetCDF

;*****
;-----
; MAP PLOT
;-----
if (PLOT) then
    plot_ind = 0
    wks = gsn_open_wks(mapType,mapName)
    gsn_define_colormap(wks,"BlueWhiteOrangeRed")      ; change colour map
    ;-----
    ; define resources
    ; to use for plotting
    ;-----
    mapres = True
    mapres@tiMainString      = ""                    ; changing main large title
    mapres@gsnLeftString     = region                 ; change left
    mapres@gsnFrame          = False                 ; do not advance frame
    mapres@gsnDraw           = False                 ; don't draw it yet

    ;-----
    ; do the plotting
    ;-----
    map1 = gsn_csm_map(wks,mapres)                    ; draw a blank map

    ;-----
    ; draw emissions
    ;-----
    mapres@cnFillOn          = True
    mapres@cnFillMode        = "CellFill"             ; fill as grid boxes not contours
    mapres@cnLineLabelsOn    = False                 ; turn off contour labels
    mapres@cnLinesOn         = False

    mapres@cnLevelSelectionMode = "ManualLevels"      ; manually set the contour levels
    mapres@cnMinLevelValF     = -5e10                 ; set the minimum contour level
    mapres@cnMaxLevelValF     = 5e10                 ; set the maximum contour level
    mapres@cnLevelSpacingF    = 5e09                 ; set the interval between contours
    mapres@cnMinLevelValF     = -5e11                 ; set the minimum contour level
    mapres@cnMaxLevelValF     = 5e11                 ; set the maximum contour level
    mapres@cnLevelSpacingF    = 5e10                 ; set the interval between contours

    map2 = gsn_csm_contour_map_ce(wks,emiss_region(0,::),mapres)

    ;-----
    ; draw region boxes
    ;-----
    boxres = True
    boxres@gsLineColor        = "black"
    boxres@gsLineThicknessF   = 5

```

```

;                               longitude                               latitudes
temp = gsn_add_polyline(wks,map2,(/lonmin, lonmin, lonmax , lonmax, lonmin/),(/latmin, latmax,
latmax , latmin, latmin/),boxres)

draw(map2)
frame(wks)

end if ; PLOT
end do ; LOOP THROUGH REGIONS

;-----
; MAP PLOT
;-----
if (PLOTREGIONS) then
  wks2 = gsn_open_wks(mapType2,mapName2)
  ;-----
  ; define resources
  ; to use for plotting
  ;-----
  mapres = True
  mapres@tiMainString      = ""           ; changing main large title
  mapres@gsnLeftString     = ""           ; change left
  mapres@gsnFrame          = False        ; do not advance frame
  mapres@gsnDraw            = False        ; don't draw it yet
  mapres@mpCenterLonF      =10

  ;-----
  ; do the plotting
  ;-----
  map1 = gsn_csm_map(wks2,mapres)           ; draw a blank map

  txres = True
  txres@txFontHeightF = 0.01                ; font smaller. default big

  do i=0,n_regions-3
    if (region_names(i).eq."SHAF".or.region_names(i).eq."NHAF"\
      .or.region_names(i).eq."EURO") then
      dum = gsn_add_text(wks2,map1,region_names(i),10,\
        (region_select(i,0)+region_select(i,1))/2,txres)
    else
      dum = gsn_add_text(wks2,map1,region_names(i),(region_select(i,2)+region_select(i,3))/2,\
        (region_select(i,0)+region_select(i,1))/2,txres)
    end if
  end do

  draw(map1)

  ;-----
  ; draw region boxes
  ;-----
  boxres = True
  boxres@gsLineColor      = "black"
  boxres@gsLineThicknessF = 5

  do j=0,n_regions-3
    regions_lon := (/region_select(j,2), region_select(j,2), \
      region_select(j,3) , region_select(j,3), \
      region_select(j,2)/)
    regions_lat := (/region_select(j,0), region_select(j,1), \
      region_select(j,1) , region_select(j,0), \
      region_select(j,0)/)
    gsn_polyline(wks2,map1,regions_lon,regions_lat,boxres)

  end do

```

```
    frame(wks2)
end if ; PLOTREGIONS

end
```