Figure 2.9

(a) and (b) – ice core data.

Start with ice core data available in form of in an Excel spreadsheet via the IPCC. Use office software to save this into a cvs file named su.cvs for sulfate and bc.cvs for black carbon.

Use the Linux shell script “tx2nc.ksh” (pasted below) to convert csv into NetCDF files.

NCL plotting script “plotfgd.ncl” (pasted below) generates the two panels. Set in the top of the script ‘var=”su”’ to plot (a), and ‘var=”bc”’ to plot (b).

(c) and (d) – satellite and AERONET data.

(i) MISR data.

- obtain MISR data from FTP server, select variables, and combine yearly data into one file, using script “getmisr.ksh” (quoted below)

- remap onto a 12°x6° grid by the command:

cdo remapbil,30x30 misr\_2000-2019.nc misr\_2000-2019\_30x30.nc  
with the grid description file 30x30

(ii) MODIS data.

- obtain MODIS from the ICDC server ([http://icdc.cen.uni-hamburg.de/1/daten/atmosphere/modis-aerosol-properties.html](about:blank)) and generate annual means with the script processmodis.ksh (pasted below; runs on the server)

- time series using ncrcat modis\_20??.nc modis\_2000-2019.nc

- cdo remapbil,30x30 modis\_2000-2018.nc modis\_2000-2018\_30x30.nc

- compute trends from MISR and MODIS using aodtrend.F

(iii) AERONET data.

Download data from https://aeronet.gsfc.nasa.gov/data\_push/V3/SDA/SDA\_Level20\_Monthly\_V3.tar.gz

tar xvfz SDA\_Level20\_Monthly\_V3.tar.gz  
generate list of stations using prepaeronet.ksh (pasted below)

convert to NetCDF using tx2nc.F (below)

compute trends using postproc.ksh (below)

(iv) Plot trends using plotmimoanf.ncl (below)

set fine=0 to generate (c) and fine=1 to generate (d)

**Script “tx2nc.ksh”**

#!/bin/ksh

cat > su.txt <<EOF

netcdf rf {

dimensions:

time = 27 ;

variables:

float time(time) ;

time:units = "year" ;

EOF

agents="europe russia arctic southamerica antarctica"

for i in $agents;do

cat >> su.txt <<EOF

float ${i}(time) ;

${i}:units = "ppb";

${i}:missing\_value = -999.f;

EOF

done

echo "data:" >> su.txt

line=`cat su.csv |awk 'BEGIN { FS = "," } ; { printf"%9.3f, ", $1 }'`

cat >> su.txt <<EOF

time = $line;

EOF

i=2

while [ $i -le 6 ];do

line=`cat su.csv |awk -v i="$i" 'BEGIN { FS = "," } ; { printf"%9.3f, ", $i }'`

let j=$i-1

var=`echo $agents | awk -v j="$j" '{print $j}'`

echo $var

cat >> su.txt <<EOF

$var = $line;

EOF

let i=$i+1

done

echo "}" >> su.txt

sed -e 's/\,\ \;/\;/g' su.txt > 1.txt

mv -f 1.txt su.txt

cat su.txt | ncgen -o su.nc

cat > bc.txt <<EOF

netcdf rf {

dimensions:

time = 32 ;

variables:

float time(time) ;

time:units = "year" ;

EOF

agents="europe russia arctic southamerica southasia elbrus greenland antarctica"

for i in $agents;do

cat >> bc.txt <<EOF

float ${i}(time) ;

${i}:units = "ppb";

${i}:missing\_value = -999.f;

EOF

done

echo "data:" >> bc.txt

line=`cat bc.csv |awk 'BEGIN { FS = "," } ; { printf"%9.3f, ", $1 }'`

cat >> bc.txt <<EOF

time = $line;

EOF

i=2

while [ $i -le 9 ];do

line=`cat bc.csv |awk -v i="$i" 'BEGIN { FS = "," } ; { printf"%9.3f, ", $i }'`

let j=$i-1

var=`echo $agents | awk -v j="$j" '{print $j}'`

echo $var

cat >> bc.txt <<EOF

$var = $line;

EOF

let i=$i+1

done

echo "}" >> bc.txt

sed -e 's/\,\ \;/\;/g' bc.txt > 1.txt

mv -f 1.txt bc.txt

cat bc.txt | ncgen -o bc.nc

**Script “plotfgd.ncl”**

load "$NCARG\_ROOT/lib/ncarg/nclscripts/csm/gsn\_code.ncl"

load "$NCARG\_ROOT/lib/ncarg/nclscripts/csm/gsn\_csm.ncl"

load "$NCARG\_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

load "$NCARG\_ROOT/lib/ncarg/nclscripts/csm/shea\_util.ncl"

begin

var="su"

;var="bc"

wks = gsn\_open\_wks ("eps","icesod"+var)

;wks = gsn\_open\_wks ("pdf","icesod"+var)

gsn\_define\_colormap(wks,"AR6\_Line\_Shade")

wks@wkOrientation = "landscape"

res = True

res@vpHeightF= 0.4 ; change aspect ratio of plot

res@vpWidthF = 0.7

res@gsnDraw = False

res@gsnFrame = False

res@tmYRBorderOn = False

res@tmXTBorderOn = False

res@tmYROn = False

res@tmXTOn = False

res@tmXBLabelFont=12

res@tmYLLabelFont=12

res@txFontHeightF = 0.015

if ( var.eq."su" ) then

res@gsnLeftString = "(a) Non-sea salt sulfate"

res@tiYAxisString = "(ng g~S~-1~N~)"

else

res@gsnLeftString = "(b) Refractory black carbon"

res@tiYAxisString = "(ng g~S~-1~N~)"

end if

res@tiYAxisFont=12

res@trXMaxF = 2020.

res@trYMinF = 0.

res@trXMinF = 1700

if ( var.eq."su" ) then

res@trYMaxF = 999 ; sulfate

else

res@trYMaxF = 14.99 ; bc

end if

;res@tmYLPrecision = 1

res@tiMainFont = 12

res@xyLineThicknessF=6.

res@gsnMaximize = True

;res@gsnYRefLine = 0.

txres = True ; text mods desired

txres@txJust = "CenterLeft" ; Default is "CenterCenter".

if ( var.eq."su" ) then

f = addfile("su.nc","r")

else

f = addfile("bc.nc","r")

end if

europe = f->europe

russia = f->russia

arctic = f->arctic

southamerica = f->southamerica

antarctica = f->antarctica

if ( var.eq."bc" ) then

elbrus = f->elbrus

greenland = f->greenland

end if

res@xyLineColor="(/0.031,0.180,0.445/)"

plot = gsn\_csm\_xy(wks,europe&time,europe,res)

res@xyLineColor="(/0.207,0.645,0.770/)"

p1 = gsn\_csm\_xy(wks,europe&time,russia,res)

overlay(plot,p1)

res@xyLineColor="(/0.863,0.328,0.180/)"

p2 = gsn\_csm\_xy(wks,europe&time,arctic,res)

overlay(plot,p2)

res@xyLineColor="(/0.922,0.609,0.180/)"

p3 = gsn\_csm\_xy(wks,europe&time,southamerica,res)

overlay(plot,p3)

res@xyLineColor="(/0.195,0.496,0.316/)"

p4 = gsn\_csm\_xy(wks,europe&time,antarctica,res)

overlay(plot,p4)

if ( var.eq."bc" ) then

res@xyLineColor="(/0.664,0.094,0.094/)"

p5 = gsn\_csm\_xy(wks,europe&time,elbrus,res)

overlay(plot,p5)

res@xyLineColor="(/0.129,0.203,0.855/)"

p6 = gsn\_csm\_xy(wks,europe&time,greenland,res)

overlay(plot,p6)

end if

if ( var.eq."su" ) then

txres@txFontHeightF = 0.02 ; text font height

txres@txFontColor="(/0.863,0.328,0.180/)"

gsn\_text\_ndc(wks,"Arctic",.19,.72,txres)

txres@txFontColor="(/0.207,0.645,0.770/)"

gsn\_text\_ndc(wks,"Russia",.19,.69,txres)

txres@txFontColor="(/0.031,0.190,0.445/)"

gsn\_text\_ndc(wks,"Europe",.19,.66,txres)

txres@txFontColor="(/0.922,0.609,0.190/)"

gsn\_text\_ndc(wks,"South America",.19,.63,txres)

txres@txFontColor="(/0.195,0.496,0.316/)"

gsn\_text\_ndc(wks,"Antarctica",.19,.60,txres)

else

txres@txFontHeightF = 0.02 ; text font height

txres@txFontColor="(/0.863,0.328,0.190/)"

gsn\_text\_ndc(wks,"Arctic",.19,.72,txres)

txres@txFontColor="(/0.129,0.203,0.855/)"

gsn\_text\_ndc(wks,"Greenland",.19,.69,txres)

txres@txFontColor="(/0.207,0.645,0.770/)"

gsn\_text\_ndc(wks,"Russia",.19,.66,txres)

txres@txFontColor="(/0.664,0.094,0.094/)"

gsn\_text\_ndc(wks,"Eastern Europe",.19,.63,txres)

txres@txFontColor="(/0.031,0.190,0.445/)"

gsn\_text\_ndc(wks,"Europe",.19,.60,txres)

txres@txFontColor="(/0.922,0.609,0.190/)"

gsn\_text\_ndc(wks,"South America",.19,.57,txres)

txres@txFontColor="(/0.195,0.496,0.316/)"

gsn\_text\_ndc(wks,"Antarctica",.19,.54,txres)

end if

draw(plot)

frame(wks)

end

**Script “getmisr.ksh”**

#!/bin/ksh

# NASA LARC FTP repository for annual-average aerosol retrievals

#ftp=ftp://l5ftl01.larc.nasa.gov/misrl2l3/MISR/MIL3YAEN.004

ftp=https://opendap.larc.nasa.gov/opendap/MISR/MIL3YAEN.004/

# Download data

y=2000

while [ $y -le 2019 ];do

let y1=$y+1 # in file name year plus one

f=MISR\_AM1\_CGAS\_${y1}\_F15\_0032.nc

echo $y $f

#wget $ftp/$y.12.01/$f

ncatted -agrid\_mapping,,d,, $f

/opt/nco/bin/ncks -3 -O -g Aerosol\_Parameter\_Average -v Aerosol\_Optical\_Depth,Small\_Mode\_Aerosol\_Optical\_Depth -dOptical\_Depth\_Range,0 $f 1.nc

ncwa -O -a Optical\_Depth\_Range 1.nc misr\_$y1.nc

\rm 1.nc

let y=$y+1

done

ncecat -O misr\_20??.nc misr\_2000-2019.nc

**Script “processmodis.ksh”**

#!/bin/ksh

d=/pool/data/ICDC/atmosphere/modis\_terra\_aerosol/DATA/

f0=MODIS-C6.1\_\_MOD08\_\_daily\_\_aerosol-parameters\_\_

f1=\_\_UHAM-ICDC\_\_fv0.2.nc

y=1999

let y1=$y+1

(cdo -O enspctl,50 $d/$y1/${f0}${y1}0???${f1} $d/$y1/${f0}${y1}1[0-1]??${f1} $y.nc; cdo selvar,aod\_landocean,aod550\_ocean\_fm\_qa $y1.nc modis\_$y1.nc; \rm $y1.nc)&

let y=$y+1

while [ $y -le 2002 ];do

let y1=$y+1

(cdo -O enspctl,50 $d/$y/${f0}${y}12??${f1} $d/$y1/${f0}${y1}0???${f1} $d/$y1/${f0}${y1}1[0-1]??${f1} $y.nc; cdo selvar,aod\_landocean,aod550\_ocean\_fm\_qa $y1.nc modis\_$y1.nc; \rm $y1.nc)&

let y=$y+1

done

da=/pool/data/ICDC/atmosphere/modis\_aqua\_aerosol/DATA/

f0a=MODIS-C6.1\_\_MYD08\_\_daily\_\_aerosol-parameters\_\_

while [ $y -le 2018 ];do

let y1=$y+1

(cdo -O enspctl,50 $d/$y/${f0}${y}12??${f1} $d/$y1/${f0}${y1}0???${f1} $d/$y1/${f0}${y1}1[0-1]??${f1} $da/$y/${f0a}${y}12??${f1} $da/$y1/${f0a}${y1}0???${f1} $da/$y1/${f0a}${y1}1[0-1]??${f1} $y.nc; cdo selvar,aod\_landocean,aod550\_ocean\_fm\_qa $y1.nc modis\_$y1.nc; \rm $y1.nc)&

let y=$y+1

done

**Script “prepaeronet.ksh”**

#!/bin/ksh

touch stations

for i in SDA/SDA20/MONTHLY/19930101\_202\*\_lev20;do

stat=`echo $i|sed -e 's/SDA\/SDA20\/MONTHLY\/19930101\\_20190105\\_//g'|sed -e 's/.ONEILL\\_lev20//g'`

y=`tail -1 $i |awk -F"," '{print $39}'`

x=`tail -1 $i |awk -F"," '{print $40}'`

nt=`wc -l $i|awk -F" " '{print $1}'`

let nt=$nt-7

echo $stat >> stations

echo $nt > $stat

echo $x >> $stat

echo $y >> $stat

cat $i|awk 'BEGIN { FS = "," } ; { printf "%12.6f%12.6f\n", $2, $3}' >> $stat

done

**Script “tx2nc.F”**

PROGRAM tx2nc

c gfortran -I$NCDFINC tx2nc.F -L$NCDFLIB -lnetcdf -lnetcdff -o tx2nc.x

IMPLICIT NONE

include "netcdf.inc"

INTEGER nx, ny, nt, n

! PARAMETER (nx=30, ny=30, nt=312, n=1201) ! all stations

! PARAMETER (nx=30, ny=30, nt=312, n=1014) ! stations with > 10 years data

! PARAMETER (nx=30, ny=30, nt=312, n=511) ! stations with > 20 years data

PARAMETER (nx=30, ny=30, nt=323, n=511) ! stations with > 20 years data

REAL aodin(n,nt),aodfin(n,nt)

REAL aod(nx,ny,nt), aodf(nx,ny,nt)

REAL naod(nx,ny,nt), naodf(nx,ny,nt)

INTEGER ncid, aodid, aodfid, xid, yid, tid

INTEGER xdim(1), ydim(1), tdim(1)

INTEGER x1id, y1id

INTEGER start(3), count(3), dims(3)

INTEGER start2(2), count2(2), dims2(2)

INTEGER i,j,l,k,ierr

INTEGER ll, l1

REAL xx,yy

REAL x(nx,ny), y(nx,ny), time(nt)

REAL x1(nx), y1(ny)

REAL miss

PARAMETER (miss=-999.)

CHARACTER\*50 station

aod(:,:,:) = 0.

aodf(:,:,:) = 0.

naod(:,:,:) = 0.

naodf(:,:,:) = 0.

OPEN(1,file='stations')

DO k = 1,n

READ(1,\*) station

OPEN(2,file='stations.txt/'//station)

READ(2,'(i6)') ll

READ(2,'(f9.6)') xx

READ(2,'(f9.6)') yy

DO l=1,ll

READ(2,'(2f12.6)') aodin(k,l),aodfin(k,l)

ENDDO

CLOSE(2)

i = NINT((180.+xx)/12.)+1

j = NINT((90.-yy)/6.)+1

write(\*,\*) station,xx,yy,i,j,ll

l1=0

DO l=nt-ll,nt

l1=l1+1

IF (aodin(k,l1).GT.0.) THEN

aod(i,j,l) = aod(i,j,l)+aodin(k,l1)

naod(i,j,l) = naod(i,j,l) + 1.

ENDIF

IF (aodfin(k,l1).GT.0.) THEN

aodf(i,j,l) = aodf(i,j,l)+aodfin(k,l1)

naodf(i,j,l) = naodf(i,j,l) + 1.

ENDIF

ENDDO

ENDDO

CLOSE(1)

DO i=1,nx

DO j=1,ny

DO l=1,nt

IF ( naod(i,j,l).GT.0. ) THEN

aod(i,j,l) = aod(i,j,l) / naod(i,j,l)

ELSE

aod(i,j,l) = miss

ENDIF

IF ( naodf(i,j,l).GT.0. ) THEN

aodf(i,j,l) = aodf(i,j,l) / naodf(i,j,l)

ELSE

aodf(i,j,l) = miss

ENDIF

ENDDO

ENDDO

ENDDO

start(1:3)=1

count(1)=nx

count(2)=ny

count(3)=nt

start2(1:2)=1

count2(1)=nx

count2(2)=ny

DO i=1,nx

x(i,:)=FLOAT(i-1)\*12-174.

x1(i)=x(i,1)

ENDDO

DO j=1,ny

y(:,j)=87.-FLOAT(j-1)\*6.

y1(j)=y(1,j)

ENDDO

DO l=1,nt

time(l)=FLOAT(l)

ENDDO

!ncid = NCCRE ('aeronet.nc', NCCLOB, ierr)

ncid = NCCRE ('aeronet\_10yrs.nc', NCCLOB, ierr)

IF (ierr.NE.NF\_NOERR) STOP 'error creating file'

ierr = NF\_DEF\_DIM(ncid, 'lon', nx, xdim)

IF (ierr.NE.NF\_NOERR) STOP 'create x d'

ierr = NF\_DEF\_DIM(ncid, 'lat', ny, ydim)

IF (ierr.NE.NF\_NOERR) STOP 'create y d'

ierr = NF\_DEF\_DIM(ncid, 'time', nt, tdim)

IF (ierr.NE.NF\_NOERR) STOP 'create t d'

dims(1)=xdim(1)

dims(2)=ydim(1)

dims(3)=tdim(1)

dims2(1)=xdim(1)

dims2(2)=ydim(1)

ierr = NF\_DEF\_VAR (ncid, 'xlon', NF\_FLOAT, 2, dims2, xid)

IF (ierr.NE.NF\_NOERR) STOP 'def xlon'

ierr = NF\_PUT\_ATT\_TEXT(ncid, xid, 'units', 9, 'degrees\_E')

IF (ierr.NE.NF\_NOERR) STOP 'error def units'

ierr = NF\_DEF\_VAR (ncid, 'ylat', NF\_FLOAT, 2, dims2, yid)

IF (ierr.NE.NF\_NOERR) STOP 'def ylat'

ierr = NF\_DEF\_VAR (ncid, 'lon', NF\_FLOAT, 1, xdim, x1id)

IF (ierr.NE.NF\_NOERR) STOP 'def lon'

ierr = NF\_PUT\_ATT\_TEXT(ncid, x1id, 'units', 9, 'degrees\_E')

IF (ierr.NE.NF\_NOERR) STOP 'error def units'

ierr = NF\_DEF\_VAR (ncid, 'lat', NF\_FLOAT, 1, ydim, y1id)

IF (ierr.NE.NF\_NOERR) STOP 'def lat'

ierr = NF\_PUT\_ATT\_TEXT(ncid, y1id, 'units', 9, 'degrees\_N')

IF (ierr.NE.NF\_NOERR) STOP 'error def units'

ierr = NF\_DEF\_VAR (ncid, 'time', NF\_FLOAT, 1, tdim, tid)

IF (ierr.NE.NF\_NOERR) STOP 'def time'

ierr = NF\_PUT\_ATT\_TEXT(ncid, tid, 'units', 23,

. 'months since 1993-12-15')

IF (ierr.NE.NF\_NOERR) STOP 'error def units'

!

ierr = NF\_DEF\_VAR (ncid, 'aod', NF\_FLOAT, 3, dims, aodid)

IF (ierr.NE.NF\_NOERR) STOP 'def var'

ierr = NF\_PUT\_ATT\_REAL(ncid, aodid, 'missing\_value',

. NF\_FLOAT, 1, miss)

IF (ierr.NE.NF\_NOERR) stop 'error def missing value'

ierr = NF\_DEF\_VAR (ncid, 'aodf', NF\_FLOAT, 3, dims, aodfid)

IF (ierr.NE.NF\_NOERR) STOP 'def var'

ierr = NF\_PUT\_ATT\_REAL(ncid, aodfid, 'missing\_value',

. NF\_FLOAT, 1, miss)

IF (ierr.NE.NF\_NOERR) stop 'error def missing value'

!

ierr = NF\_ENDDEF(ncid)

IF (ierr.NE.NF\_NOERR) STOP 'end def .'

!

ierr = NF\_PUT\_VARA\_REAL (ncid, yid, start2, count2, y)

IF (ierr.NE.NF\_NOERR) STOP 'write y var.'

ierr = NF\_PUT\_VARA\_REAL (ncid, xid, start2, count2, x)

IF (ierr.NE.NF\_NOERR) STOP 'write x var.'

ierr = NF\_PUT\_VARA\_REAL (ncid, y1id, start(2), count(2), y1)

IF (ierr.NE.NF\_NOERR) STOP 'write y1 var.'

ierr = NF\_PUT\_VARA\_REAL (ncid, x1id, start(1), count(1), x1)

IF (ierr.NE.NF\_NOERR) STOP 'write x1 var.'

ierr = NF\_PUT\_VARA\_REAL (ncid, tid, start(3), count(3), time)

IF (ierr.NE.NF\_NOERR) STOP 'write time var.'

!

ierr = NF\_PUT\_VARA\_REAL (ncid, aodid, start, count, aod)

IF (ierr.NE.NF\_NOERR) STOP 'write aod var.'

ierr = NF\_PUT\_VARA\_REAL (ncid, aodfid, start, count, aodf)

IF (ierr.NE.NF\_NOERR) STOP 'write aodf var.'

!

ierr = NF\_CLOSE(ncid)

END PROGRAM tx2nc

**Script “plotmimoanf.ncl”**

load "$NCARG\_ROOT/lib/ncarg/nclscripts/csm/gsn\_code.ncl"

load "$NCARG\_ROOT/lib/ncarg/nclscripts/csm/gsn\_csm.ncl"

begin

fine=0

if ( fine.eq.1 ) then

wks = gsn\_open\_wks("pdf","mimoanf") ; open a pdf file

;wks = gsn\_open\_wks("eps","mimoanf") ; open a pdf file

else

wks = gsn\_open\_wks("pdf","mimoan") ; open a pdf file

;wks = gsn\_open\_wks("eps","mimoan") ; open a pdf file

end if

a = addfile("mimof.nc","r")

if ( fine.eq.1 )then

tmm = a->tmf

smm = a->smf

else

tmm = a->tmm

smm = a->smm

end if

;gsn\_define\_colormap(wks,"BrownBlue12")

;gsn\_define\_colormap(wks,"CBR\_coldhot")

;gsn\_define\_colormap(wks,"AR6\_Temp\_10")

gsn\_define\_colormap(wks,"chem\_div")

;gsn\_define\_colormap(wks,"GreenMagenta16")

;gsn\_define\_colormap(wks,"BlueDarkRed18")

;

txres = True

;txres@txFont=12

txres@txFontHeightF = 0.03 ; font smaller. default big

txres@txJust="CenterCenter"

gsn\_text\_ndc(wks,"% yr~S~-1~N~",0.5,0.2,txres)

;gsn\_text\_ndc(wks,"-71",0.272,0.305,txres)

res = True

res@gsnDraw = False

res@gsnFrame = False

res@mpProjection = "Robinson" ; choose projection

res@mpGridAndLimbOn = True ; turn on lat/lon lines

res@mpGridLineColor = 0.

res@mpPerimOn = False ; turn off box around plot

res@mpGridLatSpacingF = 30. ; spacing for lat lines

res@mpGridLonSpacingF = 30. ; spacing for lon lines

res@mpGridLineDashPattern = 2

res@mpGeophysicalLineThicknessF = 2

res@mpLimbLineDashPattern=0

res@mpLimitMode = "LatLon"

res@mpMinLatF = -70.

res@mpMaxLatF = 70.

res@mpFillOn = False

res@cnFillOn = True ; color plot desired

res@txFontHeightF = 0.015

res@cnFillMode = "CellFill"

res@cnLinesOn = False ; turn off contour lines

res@cnLineLabelsOn = False

res@cnLevelSelectionMode = "ExplicitLevels"

;colors=(/3\*16,4\*16,5\*16,6\*16,7\*16,0\*16,8\*16,9\*16,10\*16,11\*16,12\*16/)

colors=(/3,3+21,3+21\*2,3+4\*21,3+5\*21,3+6\*21,0,3+7\*21,3+8\*21,3+9\*21,3+10\*21,3+12\*21/)

res@cnFillColors = colors

;res@cnLevels = (/-2,-1.8,-1.6,-1.4,-1.2,-1,-.8,-.6,-.4,-.2,0,.2,.4,.6,.8,1,1.2,1.4,1.6,1.8,2.0/)

;res@cnLevels = (/-1.8,-1.2,-.6,0,.6,1.2,1.8/)

res@cnLevels = (/-20,-1.6,-1.2,-0.8,-0.4,0,0.4,0.8,1.2,1.6,20/)

res@pmLabelBarDisplayMode = "Conditional"

res@pmLabelBarOrthogonalPosF = -0.01 ; move farther to plot

res@pmLabelBarParallelPosF = 0.5

res@lbOrientation = "horizontal" ; vertical label bars

;res@lbLabelAlignment = "ExternalEdges"

;res@cnMaxDataValueFormat = "\*+.2^sg"

;res@cnLabelBarEndStyle = "IncludeMinMaxLabels"

res@pmLabelBarHeightF = 0.06

res@pmLabelBarWidthF = 0.4

res@lbLabelFontHeightF = 0.015 ; Decrease font size.

res@cnMissingValFillPattern = 0 ; choose fill pattern 0 (solid fill)

;res@cnMissingValFillColor = "(/0.75,0.75,0.75/)"

res@cnMissingValFillColor = "grey" ; should be darkgrey

;

res@tiMainString = ""

if ( fine.eq.1 ) then

res@gsnLeftString = "(d)"

res@gsnCenterString = " Trend in fine-mode aerosol optical depth (2000 - 2019)" ; don't use the variable long\_name

else

res@gsnLeftString = "(c)"

res@gsnCenterString = "Trend in aerosol optical depth (2000 - 2019)" ; don't use the variable long\_name

end if

res@gsnRightString = "" ; don't use the variable units

res@gsnPaperOrientation = "Landscape"

res@gsnMaximize = True

;cmap\_data = read\_colormap\_file("chem\_div")

;res@cnFillPalette = cmap\_data(::-1,:)

contour = gsn\_csm\_contour\_map(wks,tmm,res) ; create the plot

; www.ncl.ucar.edu/Applications/Scripts/scatter\_6.ncl

a = addfile("../aeronet/antrend.nc","r")

if ( fine.eq.1 ) then

tanet = a->taodf

sanet = a->saodf

else

tanet = a->taod

sanet = a->saod

end if

tanet=mask(tanet,sanet,0.1)

xlon = a->xlon

ylat = a->xlat

an=ndtooned(tanet)

; print(an)

lon=ndtooned(xlon)-6.

lat=ndtooned(ylat)

mkres2 = True

mkres2@gsMarkerIndex = 5

mkres2@gsMarkerOpacityF = 0.5

nx=30

ny=30

sigmark = new(nx\*ny,graphic)

sigmark1 = new(nx\*ny,graphic)

do j=0,ny-1

mkres2@gsMarkerSizeF = 15+30\*cos(ylat(0,j)/180\*3.141)

do i=0,nx-1

if (smm(i,j).lt.0.1) then

sigmark(i\*nx+j)=gsn\_add\_polymarker(wks,contour,xlon(i,j)-9,ylat(i,j)\*-1,mkres2)

sigmark1(i\*nx+j)=gsn\_add\_polymarker(wks,contour,xlon(i,j)-3.,ylat(i,j)\*-1,mkres2)

end if

end do

end do

mkres = True

mkres@mpProjection = "Robinson" ; choose projection

;---Arrays for attaching two sets of markers

nlevels=11

dum\_fill = new(nlevels-1,graphic)

dum\_hollow = new(nlevels-1,graphic)

mkres@gsMarkerThicknessF = 2.0 ; Twice as thick

mkres1 = True

mkres1@gsMarkerThicknessF = 2.0 ; Twice as thick

;

; For each range, gather the data that falls in this range

; and draw the set of markers at those lat/lon locations.

;

levels=res@cnLevels

;colors=(/3,4,5,6,7,0,8,9,10,11,12/)

;colors=gsn\_retrieve\_colormap(wks)

;print(colors)

do i=0,nlevels-2

ii = ind(levels(i).le.an.and.an.lt.levels(i+1))

if (.not.ismissing(ii(0))) then

;print(i)

;print(colors(i))

;print(an(ii))

;print(lon(ii))

;print(lat(ii))

mkres@gsMarkerIndex = 16 ; Filled dots

mkres@gsMarkerSizeF = 10

mkres@gsMarkerColor = colors(i)

dum\_fill(i) = gsn\_add\_polymarker(wks,contour,lon(ii),lat(ii),mkres)

;---Draw the markers again, only this time hollow, to get an outline.

mkres1@gsMarkerIndex = 4 ; Hollow dots

mkres1@gsMarkerSizeF = 10

mkres1@gsMarkerColor = "black"

dum\_hollow(i) = gsn\_add\_polymarker(wks,contour,lon(ii),lat(ii),mkres1)

end if

delete(ii)

end do

draw(contour)

frame(wks)

end

**Script “processmodis.ksh”**