```
netcdf \2A12.20131011.90598.7 {
dimensions:
      nscan = 2920 ;
      npixel = 208 ;
      fakeDim2 = 3 ;
      fakeDim3 = 3 ;
      nspecies = 6 ;
      nlayer = 28 ;
      ncluster = 100 ;
      nfindex = 13 ;
variables:
      short Year(nscan) ;
            Year:hdf_name = "Year" ;
            Year:units = "years" ;
      byte Month(nscan) ;
            Month:hdf_name = "Month" ;
            Month:units = "months" ;
      byte DayOfMonth(nscan) ;
            DayOfMonth:hdf_name = "DayOfMonth" ;
            DayOfMonth:units = "days" ;
      byte Hour(nscan) ;
            Hour:hdf_name = "Hour" ;
            Hour:units = "hours" ;
      byte Minute(nscan) ;
            Minute:hdf_name = "Minute" ;
            Minute:units = "minutes" ;
      byte Second(nscan) ;
            Second:hdf_name = "Second" ;
            Second:units = "s" ;
      short MilliSecond(nscan) ;
            MilliSecond:hdf_name = "MilliSecond" ;
            MilliSecond:units = "ms" ;
      short DayOfYear(nscan) ;
            DayOfYear:hdf_name = "DayOfYear" ;
            DayOfYear:units = "days" ;
      float Latitude(nscan, npixel) ;
            Latitude:hdf_name = "Latitude" ;
            Latitude:units = "degrees" ;
      float Longitude(nscan, npixel) ;
            Longitude:hdf_name = "Longitude" ;
            Longitude:units = "degrees" ;
      byte missing(nscan) ;
            missing:hdf_name = "missing" ;
      byte validity(nscan) ;
            validity:hdf_name = "validity" ;
      byte qac(nscan) ;
            qac:hdf_name = "qac" ;
      byte geoQuality(nscan) ;
            geoQuality:hdf_name = "geoQuality" ;
      byte dataQuality(nscan) ;
            dataQuality:hdf_name = "dataQuality" ;
      short SCorientation(nscan) ;
            SCorientation:hdf_name = "SCorientation" ;
            SCorientation:units = "degrees" ;
```

```
      byte acsMode(nscan) ;
            acsMode:hdf_name = "acsMode" ;
      byte yawUpStat(nscan) ;
            yawUpStat:hdf_name = "yawUpStat" ;
      byte tmiIsStatus(nscan) ;
            tmiIsStatus:hdf_name = "tmiIsStatus" ;
      double FractionalGranuleNumber(nscan) ;
            FractionalGranuleNumber:hdf_name = "FractionalGranuleNumber"
;
      float scPosX(nscan) ;
            scPosX:hdf_name = "scPosX" ;
            scPosX:units = "m" ;
      float scPosY(nscan) ;
            scPosY:hdf_name = "scPosY" ;
            scPosY:units = "m" ;
      float scPosZ(nscan) ;
            scPosZ:hdf_name = "scPosZ" ;
            scPosZ:units = "m" ;
      float scVelX(nscan) ;
            scVelX:hdf_name = "scVelX" ;
            scVelX:units = "m/s" ;
      float scVelY(nscan) ;
            scVelY:hdf_name = "scVelY" ;
            scVelY:units = "m/s" ;
      float scVelZ(nscan) ;
            scVelZ:hdf_name = "scVelZ" ;
            scVelZ:units = "m/s" ;
      float scLat(nscan) ;
            scLat:hdf_name = "scLat" ;
            scLat:units = "degrees" ;
      float scLon(nscan) ;
            scLon:hdf_name = "scLon" ;
            scLon:units = "degrees" ;
      float scAlt(nscan) ;
            scAlt:hdf_name = "scAlt" ;
            scAlt:units = "m" ;
      float scAttRoll(nscan) ;
            scAttRoll:hdf_name = "scAttRoll" ;
            scAttRoll:units = "degrees" ;
      float scAttPitch(nscan) ;
            scAttPitch:hdf_name = "scAttPitch" ;
            scAttPitch:units = "degrees" ;
      float scAttYaw(nscan) ;
            scAttYaw:hdf_name = "scAttYaw" ;
            scAttYaw:units = "degrees" ;
      float SensorOrientationMatrix(nscan, fakeDim2, fakeDim3) ;
            SensorOrientationMatrix:hdf_name = "SensorOrientationMatrix"
;
      float greenHourAng(nscan) ;
            greenHourAng:hdf_name = "greenHourAng" ;
            greenHourAng:units = "degrees" ;
      byte qualityFlag(nscan, npixel) ;
            qualityFlag:hdf_name = "qualityFlag" ;
      byte pixelStatus(nscan, npixel) ;
```

```
      pixelStatus:hdf_name = "pixelStatus" ;
byte surfaceType(nscan, npixel) ;
      surfaceType:hdf_name = "surfaceType" ;
byte landAmbiguousFlag(nscan, npixel) ;
      landAmbiguousFlag:hdf_name = "landAmbiguousFlag" ;
byte landScreenFlag(nscan, npixel) ;
      landScreenFlag:hdf_name = "landScreenFlag" ;
byte oceanExtendedDbase(nscan, npixel) ;
      oceanExtendedDbase:hdf_name = "oceanExtendedDbase" ;
      oceanExtendedDbase:units = "percent" ;
byte oceanSearchRadius(nscan, npixel) ;
      oceanSearchRadius:hdf_name = "oceanSearchRadius" ;
short chiSquared(nscan, npixel) ;
      chiSquared:hdf_name = "chiSquared" ;
byte probabilityOfPrecip(nscan, npixel) ;
      probabilityOfPrecip:hdf_name = "probabilityOfPrecip" ;
      probabilityOfPrecip:units = "percent" ;
byte sunGlintAngle(nscan, npixel) ;
      sunGlintAngle:hdf_name = "sunGlintAngle" ;
      sunGlintAngle:units = "degrees" ;
short freezingHeight(nscan, npixel) ;
      freezingHeight:hdf_name = "freezingHeight" ;
      freezingHeight:units = "m" ;
float surfacePrecipitation(nscan, npixel) ;
      surfacePrecipitation:hdf_name = "surfacePrecipitation" ;
      surfacePrecipitation:units = "mm/hr" ;
float convectPrecipitation(nscan, npixel) ;
      convectPrecipitation:hdf_name = "convectPrecipitation" ;
      convectPrecipitation:units = "mm/hr" ;
float surfaceRain(nscan, npixel) ;
      surfaceRain:hdf_name = "surfaceRain" ;
      surfaceRain:units = "mm/hr" ;
float cloudWaterPath(nscan, npixel) ;
      cloudWaterPath:hdf_name = "cloudWaterPath" ;
      cloudWaterPath:units = "kg/m^2" ;
float rainWaterPath(nscan, npixel) ;
      rainWaterPath:hdf_name = "rainWaterPath" ;
      rainWaterPath:units = "kg/m^2" ;
float iceWaterPath(nscan, npixel) ;
      iceWaterPath:hdf_name = "iceWaterPath" ;
      iceWaterPath:units = "kg/m^2" ;
float seaSurfaceTemperature(nscan, npixel) ;
      seaSurfaceTemperature:hdf_name = "seaSurfaceTemperature" ;
      seaSurfaceTemperature:units = "K" ;
float totalPrecipitableWater(nscan, npixel) ;
      totalPrecipitableWater:hdf_name = "totalPrecipitableWater" ;
      totalPrecipitableWater:units = "mm" ;
float windSpeed(nscan, npixel) ;
      windSpeed:hdf_name = "windSpeed" ;
      windSpeed:units = "m/s" ;
byte freezingHeightIndex(nscan, npixel) ;
      freezingHeightIndex:hdf_name = "freezingHeightIndex" ;
byte clusterNumber(nscan, npixel, nspecies) ;
      clusterNumber:hdf_name = "clusterNumber" ;
```

```
      float clusterScale(nscan, npixel, nspecies) ;
            clusterScale:hdf_name = "clusterScale" ;
      float heightLayerTop(nlayer) ;
            heightLayerTop:hdf_name = "heightLayerTop" ;
            heightLayerTop:units = "km" ;
      float cluster(ncluster, nlayer, nfindex, nspecies) ;
            cluster:hdf_name = "cluster" ;

// global attributes:
            :SwathHeader = "NumberScansInSet=1;\n",
                  "MaximumNumberScansTotal=3100;\n",
                  "NumberScansBeforeGranule=0;\n",
                  "NumberScansGranule=2920;\n",
                  "NumberScansAfterGranule=0;\n",
                  "NumberPixels=208;\n",
                  "ScanType=CONICAL;" ;
            :GprofInfo = "Satellite=TRMM;\n",
                  "Sensor=TMI;\n",
                  "OceanDatabase=gprof_TMIpost_V7_233.dtb;\n",
                  "LandDatabase=TMIrainstr_land_400.dtb;\n",
                  "StructureFlag=-1;\n",
                  "nSpecies=6;\n",
                  "nFIndex=13;\n",
                  "nLayer=28;\n",
                  "nCluster=100;\n",
                  "Comment1=Successful Completion;\n",
                  "Comment2=none;\n",
                  "Dummy=none;" ;
            :FileInfo = "DataFormatVersion=m;\n",
                  "TKCodeBuildVersion=1;\n",
                  "MetadataVersion=m;\n",
                  "FormatPackage=HDF Version 4.2 Release 4, January 25,
2009;\n",
                  "BlueprintFilename=TRMM.V7.2A12.blueprint.xml;\n",
                  "BlueprintVersion=BV_13;\n",
                  "TKIOVersion=1.6;\n",
                  "MetadataStyle=PVL;\n",
                  "EndianType=LITTLE_ENDIAN;" ;
            :NavigationRecord =
"LongitudeOfMaximumLatitude=163.915068;\n",
                  "SolarBetaAngleAtBeginningOfGranule=12.424680;\n",
                  "SolarBetaAngleAtEndOfGranule=12.624200;" ;
            :InputRecord = "InputFileNames=1B11.20131011.90598.7.HDF;\n",
                  "InputAlgorithmVersions=7.0;\n",
                  "InputGenerationDateTimes=2013-10-12T14:05:48.000Z;" ;
            :FileHeader = "AlgorithmID=2A12;\n",
                  "AlgorithmVersion=PPS.V1;\n",
                  "FileName=2A12.20131011.90598.7.HDF;\n",
                  "GenerationDateTime=2013-10-12T18:14:02.000Z;\n",
                  "StartGranuleDateTime=2013-10-11T15:52:35.645Z;\n",
                  "StopGranuleDateTime=2013-10-11T17:24:58.744Z;\n",
                  "GranuleNumber=90598;\n",
                  "NumberOfSwaths=1;\n",
                  "NumberOfGrids=0;\n",
```

```
                "GranuleStart=SOUTHERNMOST_LATITUDE;\n",
                "TimeInterval=ORBIT;\n",
                "ProcessingSystem=PPS;\n",
                "ProductVersion=7;\n",
                "MissingData=0;" ;
            :creation_date = "Wed Mar 17 12:49:54 IST 2021" ;
            :NCL_Version = "6.5.0" ;
            :system = "Linux DESKTOP-7QCT81O 4.4.0-18362-Microsoft #1049-
Microsoft Thu Aug 14 12:01:00 PST 2020 x86_64 x86_64 x86_64 GNU/Linux" ;
            :Conventions = "None" ;
            :hdf_source = "2A12.20131011.90598.7.HDF" ;
            :title = "NCL: convert-HDF-to-netCDF" ;
}
```