VPRM preprocessor 安装及运行指南

By. 赵漾

一、 相关库安装

①很多基于 R 的库可以不用下载编译包，直接 conda 安装

```
conda install r-base
conda install r-rnetcdf
conda install -c conda-forge r-rgdal
……
```

库名查询网址：https://anaconda.org/conda-forge/repo

②h5r 包下载

这个要注意的是下载的是 h5r，不是 hdf5r!

下载网址：https://cran.r-project.org/src/contrib/Archive/h5r/

相关文档：http://www2.uaem.mx/r-mirror/web/packages/h5r/index.html

我刚开始直接装了 hdf5r，结果运行 VPRMpreproc.r 时总是会出现找不到 getH5Dataset 函数这样的报错提示。像 getH5Dataset,getH5Group 等等之类的函数是在 h5r package 里的。

同时，运行前我还在 VPRMpreproc.r 中将 library("hdf5")更改为了 library("h5r")，因为系统会提示找不到 hdf5。

③安装 h4toh5tools 之前需要先安装 hdf5 tools,并且需要注意的是 hdf5 tools 这个包在 CentOS 下的名称为 hdf5-devel:

```
yum -y install hdf5-devel
```

参考网址：https://github.com/deepmind/torch-hdf5/issues/77

④h4toh5tools 安装：

在这之前需要把 hdf4 和 hdf5 的库安装好，也就是整个过程需要 3 个安装包，其中

hdf4 下载网址：

https://portal.hdfgroup.org/display/support/HDF+4.2.15

hdf5 下载网址：

https://portal.hdfgroup.org/display/support/HDF5+1.8.22

h4toh5tools 下载网址：

https://portal.hdfgroup.org/display/support/h4h5tools%202.2.5#files

```
./configure CC=/data1/zy/h4h5tools/hdf-4.2.15/hdf4/bin/h4cc
--with-hdf5=/data1/zy/hdf5/hdf5-1.12.0/hdf5/
--prefix=/data1/zy/h4h5tools/h4h5tools-2.2.5/h4toh5/
```

该命令在 h4h5tools-2.2.5/release_docs/INSTALL_Unix.txt 中找到，所以在安装东西时最好仔细看下说明文档，我之前就想走捷径碰运气直接 configure 一堆报错，所以要好好看说明！

CC 指的是 C Compiler，需要指定为 h4cc
需要注意的是 hdf5 的地址写到 bin 所在的目录，而不是 hdf5/bin/这种 bin 的下级目录!! 这个错误我找了好久！
安装完成后，要在./bashr 里设置好 PATH (for bin) and LD_LIBRARY_PATH (for libs).

⑤拿到的 ldope 和 MODmrt tools 是已经 patch 后的，直接在 config.r 里面设置好路径即可。
VPRM Users Manual 里说需要用 cp_proj_param.patch and math_sds.patch 去对 math_sds.c and cp_proj_param.c 打补丁，可能是因为这个手册比较老，这是针对早先的开发人员进行的说明，我们现在不需要这些操作了。

二、 运行前

①Rsources/ReadSynmapDMM.r 里的 infile 改成自己存放原始 synmap 文件(synmap_LC_jan1)的路径。

②Rsources/getModis.r 里第 115-116 行下载 MODIS 数据用的 username and password 设置好。

③./compile.sh 后，首先要运行 get_synmap.sh (RSources/getSynmap.r)，该代码主要是在 rdatain 下产生 .RDatasynmap.synmaptouse 文件，供运行./preprocess.sh (VPRMpreproc.r) 时用。

如果 rdatain 目录下没有.RDatasynmap.synmaptouse 文件，运行./preprocess.sh 后 run.log 里会报错:"SYNMAP not there. Prepare one..." and "Error: No such file or directory".

注意：当 config.r 里的 ldoy 更改后，要重新运行 get_synmap.sh，否则 VPRMpreproc.r 里的下列代码

```
doyuse<-doyall<-
seq(as.numeric(paste(projinfo$year,"001",sep="")),as.numeric(past
e(projinfo$year,projinfo$ldoy,s     ep="")),by=8)
```

相关代码解读：
https://blog.csdn.net/helloworld987456/article/details/105750820

在构建起始日期时会报错（其他参数更改不知道需不需要重新运行，为了保险起见最好重新运行一份新的.RDatasynmap.synmaptouse 文件吧）。

三、 运行过程中可能遇到的报错

①Run.log：

```
167 > if (!file.exists("TLUT.RData")) TLUT <- getModis(doyuse)
168 The following object is masked _by_ .GlobalEnv:
169
170     projinfo
171
172 /data1/zy/VPRM_tools/ldope/bin/tile_id: error while loading shared libraries: libz.so.1: wrong ELF class: ELFCLASS64
173 Error in system(paste(ldope, "bin/tile_id -proj=SIN -lon=", lons[x], " -lat=",  :
174   error in running command
175 Calls: getModis -> system
176 Execution halted
```

上述错误表示系统的 64 位库不支持，需要 32 位相关 lib

参考网址：https://www.cnblogs.com/sakuraie/p/13341520.html

```
[zy@ustc bin]$ ldd /data1/zy/VPRM_tools/ldope/bin/tile_id
        linux-gate.so.1 =>  (0x55619000)
        libz.so.1 => not found
        libm.so.6 => /lib/libm.so.6 (0x55636000)
        libc.so.6 => /lib/libc.so.6 (0x55678000)
        /lib/ld-linux.so.2 (0x555f5000)
```

首先要查找 libz.so.1 属于哪个包：

yum provides <库名>

参考网址：https://blog.csdn.net/nvd11/article/details/8749335

https://www.jianshu.com/p/80b0641bbd1b

查询到 libz.so.1 属于 zlib 包，然后输入命令：sudo yum install zlib.i686（每个计算机可能不一样，有的是:i386，此处为.i686）

② svn 版本问题

```
svn: E155021: This client is too old to work with the working copy at
'/data1/zy/VPRM' (format 31).
You need to get a newer Subversion client. For more details, see
  http://subversion.apache.org/faq.html#working-copy-format-change
```

出现这个报错可能是服务器 svn 版本的问题，我查看我自己 linux 系统的 svn 版本(svn --version)发现是 1.7 的，而 VPRM/.svn/format 文件里写的是 12，format=12 对应的 svn 版本应该是 1.8，因此就用 conda 对服务器的 svn 进行了更新：

conda install -c enversion subversion

③cannot open shared object file

在 VPRMpreproc.r 运行到最后一步，也就是”write NETCDF files”时，会提示找不到一些 shared libraries,我碰到的有三个：

1. libsasl2.so.2

```
395 > for (dom in 1:length(projinfo$dxy)) {
396 +  writeVegetationMap(dom,projinfo,nveg,resdir)
397 +  for (ind in indices) writeIndices(ind,dom,projinfo,start_day_of_year,doyuse,nveg,resdir)
398 + }
399 svn: error while loading shared libraries: libsasl2.so.2: cannot open shared object file: No such file or directory
400 Error in system("svn info", intern = T) : error in running command
401 Calls: writeVegetationMap -> system
402 Execution halted
```

其实在 Centos 7 中是有这个库的，只不过不叫这个名字，进入到/usr/lib64/目录下查看，发现有一个库 libsasl2.so.3，那么我们只要使用一个软连接，链接

到 libsasl2.so.2 即可。该操作需要 root 权限。
ln -s libsasl2.so.3 libsasl2.so.2
参考网址：
https://blog.csdn.net/fengyunzhenyu/article/details/103147890

2. libssl.so.1.0.0
和上面同理，参考网址（/usr/lib64/）：
https://askubuntu.com/questions/339364/libssl-so-10-cannot-open-shared-object-file-no-such-file-or-directory
ln -s libssl.so.10 libssl.so.1.0.0

3. libcrypto.so.1.0.0
在自己的小环境/home/zy/.conda/envs/R3.6/lib/下找到了这个 package，然后直接：
ln -s libcrypto.so.1.1 libcrypto.so.1.0.0
参考网址：
https://blog.csdn.net/u013429737/article/details/115896493?ops_request_misc=%257B%2522request%255Fid%2522%253A%252216197478991678026988156%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=161974789916780269888156&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~rank_v29-19-115896493.pc_search_result_cache&utm_term=while+loading+shared+libraries%3A+libcrypto.so.1.0.0%3A

④嵌套网格文件名的设置
需要在 RSources/writeOutput.r 里将下面两行代码进行更改：

```
#min <- getr(paste(vari,".minAggregatedClassed.",year,sep=""),datp);min[is.na(min)]<-0
min <- getr(paste(vari,".minAggregatedClassed.",year,".",dom,sep=""),datp);min[is.na(min)]<-0
```

```
#max <- getr(paste(vari,".maxAggregatedClassed.",year,sep=""),datp);max[is.na(max)]<-0
max <- getr(paste(vari,".maxAggregatedClassed.",year,".",dom,sep=""),datp);max[is.na(max)]<-0
```

原始的代码可能默认的是单层嵌套，对于嵌套网格运行后的文件名后面是还有.dom 的，不然会提示找不到这个文件。

运行成功后最终会在 output 目录下得到下列文件：

```
(R3.6) [zy@ustc VPRM_input_case_2019_2019]$ ls
VPRM_input_EVI_d01_2019.nc      VPRM_input_EVI_MIN_d02_2019.nc   VPRM_input_LSWI_MAX_d03_2019.nc
VPRM_input_EVI_d02_2019.nc      VPRM_input_EVI_MIN_d03_2019.nc   VPRM_input_LSWI_MIN_d01_2019.nc
VPRM_input_EVI_d03_2019.nc      VPRM_input_LSWI_d01_2019.nc      VPRM_input_LSWI_MIN_d02_2019.nc
VPRM_input_EVI_MAX_d01_2019.nc  VPRM_input_LSWI_d02_2019.nc      VPRM_input_LSWI_MIN_d03_2019.nc
VPRM_input_EVI_MAX_d02_2019.nc  VPRM_input_LSWI_d03_2019.nc      VPRM_input_VEG_FRA_d01_2019.nc
VPRM_input_EVI_MAX_d03_2019.nc  VPRM_input_LSWI_MAX_d01_2019.nc  VPRM_input_VEG_FRA_d02_2019.nc
VPRM_input_EVI_MIN_d01_2019.nc  VPRM_input_LSWI_MAX_d02_2019.nc  VPRM_input_VEG_FRA_d03_2019.nc
```