

## R v3.1.3 语言及相关包编译安装指南

更新时间：2018 年 07 月 25 日

### 一、R 语言简介

R 语言是用于统计分析、绘图的语言和操作环境。R 是属于 GNU 系统的一个自由、免费、源代码开放的软件，它是一个用于统计计算和统计制图的优秀工具。

目前 R 最新版为 3.5.0（目前中心的 zlib 无法满足该版本，Y 分区系统与 G 分区系统不一样，建议在 G 分区安装），v 3.1.3 版可以安装。

**注：**在开始操作前，建议使用者熟悉 Linux 操作的几个常用命令（例如：vi 或 vim、cd、mkdir、ls、pwd 等）。

### 二、安装的前期准备

1. 在官网（<https://cran.r-project.org/mirrors.html>）下载源代码安装包：R-3.1.3.tar.gz。
2. 在 Windows 系统用 Xshell 客户端登录高性能帐号，编译软件必须在编译节点，以 GK 分区为例（其他分区见下表 1，请选择相应分区的编译节点）从登录节点进入编译节点：**ssh gk0110**

表 1 各分区编译节点

| 计算分区    | 编译节点   | IP 地址                                    |
|---------|--------|--|
| GG 计算分区 | gg0110 | 编译节点不可以直接访问,请从登陆节点 ssh 节点名 直接跳转即可。无密码访问。 |
|         | gg0111 |  |
| GK 计算分区 | gk0110 |  |
|         | gk0111 |  |
| GS 计算分区 | gs0110 |  |
|         | gs0111 |  |
| FN 计算分区 | fn001  |  |

|         |        |  |
|---------|--------|--|
|         | fn002  |  |
| YW 计算分区 | yw0410 |  |
|         | yw0411 |  |
| YS 计算分区 | ys0410 |  |
|         | ys0411 |  |
| YG 计算分区 | yg0110 |  |
|         | yg0111 |  |

3. 如果已创建放置源代码包的文件夹，此步骤可忽略。若没有创建，则需先创建（本次创建 Softwares/R，名称可以根据个人习惯而不同）：**mkdir -p Softwares/R**

```
nsc731_TEST@gk0110:~> mkdir -p Softwares/R
```

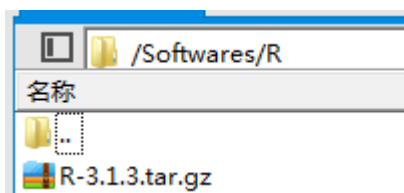
4. 如果已创建放置测试算例的文件夹，此步骤可忽略。若没有创建，则需先创建（本次创建 TEST/R，名称可以根据个人习惯而不同）：**mkdir -p TEST/R**

```
nsc731_TEST@gk0110:~> mkdir -p TEST/R
```

5. 输入查看命令就可以看到目录下创建的文件夹：**ls**

```
nsc731_TEST@gk0110:~> ls
MCNP5  Softwares  TEST  bin  cmake  install  intel  scratch  vasp
nsc731_TEST@gk0110:~>
```

6. 用高性能帐号登录 Xftp 客户端，将已下载的上述安装包上传至高性能帐号中（提前创建的文件夹，本指南中为：Softwares/R）。



7. 确认环境变量中是否已经添加需要的环境变量：**vim ~/.bashrc**
8. 输入“i”进入编辑模式，本次使用的是 intel-2016.sh 和 openmpi-1.4.4-intel.sh，则在.bashrc 中添加：

```
source /home-gk/env/intel-2016.sh
```

```
source /home-gk/env/openmpi-1.4.4-intel.sh
```

**注：**intel-12.1.sh 与 intel-2011.sh 为同一个编译器；intel-2011.sh 与 intel-2016.sh 为不同版本；openmpi-1.6.5-intel.sh 与 openmpi-1.4.5-intel.sh 为不同版本；在其他分区还会有其他版本，可以根据需要进行选择。

```
source /home-gk/env/intel-2016.sh
source /home-gk/env/openmpi-1.4.4-intel.sh
```

9. 按下键盘“Esc”键，输入“:wq”保存退出。
10. 保存以后需要运行，才能有效：`source ~/.bashrc`

### 三、编译安装 R v3.1.3

1. 进入放有源代码包的文件夹：`cd Softwares/R`
2. 解压 R 安装包：`tar -zxvf R-3.1.3.tar.gz`
3. 解压完成后，会生成 R-3.1.3，进入文件夹：`cd R-3.1.3/`
4. 在 R-3.1.3 文件夹下创建 install 和 build 文件夹：`mkdir install build`

```
nsc731_TEST@gk0110: ~/Softwares/R/R-3.1.3$ mkdir install build
```

```
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3> ls
COPYING  INSTALL  Makefile.fw  README  VERSION  build  configure  doc  install  po  src  tools
ChangeLog  Makeconf.in  Makefile.in  SVN-REVISION  VERSION-NICK  config.site  configure.ac  etc  m4  share  tests
```

5. 进入 build 文件夹：`cd build`
6. 查看路径，后续操作会用到该路径，输入命令：`pwd`

```
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3> cd build/
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3/build> pwd
/home-gk/users/nsc731_TEST/Softwares/R/R-3.1.3/build
```

7. 进行安装前的配置（后面的路径为安装的路径）：`../configure --prefix=/home-gk/users/nsc731_TEST/Softwares/R/R-3.1.3/install/`

```

R is now configured for x86_64-unknown-linux-gnu

Source directory:      ..
Installation directory: /home-gk/users/nsc731_TEST/Softwares/R/R-3.1.3/install

C compiler:           gcc -std=gnu99 -g -O2
Fortran 77 compiler:   gfortran -g -O2

C++ compiler:         g++ -g -O2
C++ 11 compiler:      gfortran -g -O2
Fortran 90/95 compiler: gfortran -g -O2
Obj-C compiler:       gfortran -g -O2

Interfaces supported:  X11
External libraries:   readline
Additional capabilities: PNG, JPEG, TIFF, NLS, ICU
Options enabled:      shared BLAS, R profiling

Capabilities skipped:  cairo
Options not enabled:   memory profiling

Recommended packages:  yes

configure: WARNING: you cannot build PDF versions of the R manuals
configure: WARNING: you cannot build PDF versions of vignettes and help pages
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3/build> █

```

8. 没有出现 error 则先编译: **make**

```

JAVA_HOME      : /usr/lib64/jvm/jre
Java library path:
JNI cpp flags  :
JNI linker flags:
Updating Java configuration in /home-gk/users/nsc731_TEST/Softwares/R/R-3.1.3/build
Done.

make[1]: Leaving directory `/home-gk/users/nsc731_TEST/Softwares/R/R-3.1.3/build'
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3/build> █

```

9. 编译没有报错再安装: **make install**

10. 安装完成后, 在 R-3.1.3/install/bin 文件夹中看到这两个可执行文件:

```

nsc731_TEST@gcn03:~/Softwares/R/R-3.1.3/install/bin> ls
R  Rscript
nsc731_TEST@gcn03:~/Softwares/R/R-3.1.3/install/bin> █

```

11. 可以在账号中另建一个文件夹拷贝生成的可执行文件。

12. 打开环境变量文件: **vim ~/.bashrc**

在环境变量中将 R 添加: **export**

**PATH=/home-gk/users/nsc731\_TEST/Softwares/R/R-3.1.3/install/bin:\$PATH**

**export**

**R\_LIBS=/home-gk/users/nsc731\_TEST/Softwares/R/R-3.1.3/install/packages:\$R\_LIBS**

```
# For R
export PATH=/home-gk/users/nsc731_TEST/Softwares/R/R-3.1.3/install/bin:$PATH
export R_LIBS=/home-gk/users/nsc731_TEST/Softwares/R/R-3.1.3/install/packages:$R_LIBS
```

13. 保存后，运行：`source ~/.bashrc`

## 四、编译安装 R 相关包

1. R 软件还有大量包提供安装，其中大部分在 CRAN 网上 <https://cran.r-project.org/>，进入网页后如下图所示，按照玫红色箭头将 xtable 包（以它为例进行说明）下载完成。

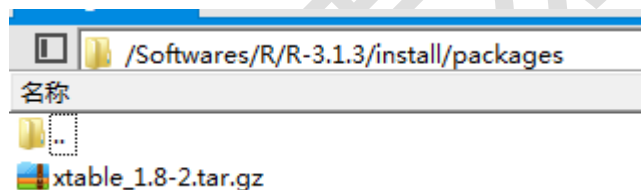
|  |  |
|--|--|
| <p>CRAN</p> <p><a href="#">Mirrors</a></p> <p><a href="#">What's new?</a></p> <p><a href="#">Task Views</a></p> <p><a href="#">Search</a></p> <p>About R</p> <p><a href="#">R Homepage</a></p> <p><a href="#">The R Journal</a></p> <p>Software</p> <p><a href="#">R Sources</a></p> <p><a href="#">R Binaries</a></p> <p><a href="#">Packages</a> ←</p> <p><a href="#">Other</a></p> <p>Documentation</p> | <p>versions of R:</p> <ul style="list-style-type: none"> <li><a href="#">Download R for Linux</a></li> <li><a href="#">Download R for (Mac) OS X</a></li> <li><a href="#">Download R for Windows</a></li> </ul> <p>R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.</p> <p>Source Code for all Platforms</p> <p>Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!</p> <ul style="list-style-type: none"> <li>The latest release (2018-04-23, Joy in Playing) <a href="#">R-3.5.0.tar.gz</a>, read <a href="#">what's new</a> in the latest version.</li> </ul> |
| <p>CRAN</p> <p><a href="#">Mirrors</a></p> <p><a href="#">What's new?</a></p> <p><a href="#">Task Views</a></p> <p><a href="#">Search</a></p> <p>About R</p> <p><a href="#">R Homepage</a></p> <p><a href="#">The R Journal</a></p> <p>Software</p> <p><a href="#">R Sources</a></p> <p><a href="#">R Binaries</a></p> <p><a href="#">Packages</a></p>   | <p><a href="#">Table of available packages, sorted by date of publication</a></p> <p><a href="#">Table of available packages, sorted by name</a> ←</p> <p>Installation of Packages</p> <p>Please type <code>help("INSTALL")</code> or <code>help("install.packages")</code> in R for information on how to install packages from this repository. The manual <a href="#">R Installation and Administration</a> (also contained in the R base sources) explains the process in detail.</p> <p><a href="#">CRAN Task Views</a> allow you to browse packages by topic and provide tools to automatically install all packages for special areas of interest. Currently, 36 views are available.</p> <p>Package Check Results</p>  |
| <p><a href="#">XRSCC</a></p> <p><a href="#">xseq</a></p> <p><a href="#">xslt</a></p> <p><a href="#">xsp</a></p> <p><a href="#">xSub</a></p> <p><a href="#">xtable</a> ←</p> <p><a href="#">xtal</a></p> <p><a href="#">xtermStyle</a></p> <p><a href="#">xtractomatic</a></p> <p><a href="#">xts</a></p>   | <p>Statistical Quality Control Simulation</p> <p>Assessing Functional Impact on Gene Expression of Mutations in Cancer</p> <p>Extensible Style-Sheet Language Transformations</p> <p>The Chi-Square Periodogram</p> <p>Cross-National Data on Sub-National Violence</p> <p>Export Tables to LaTeX or HTML</p> <p>Crystallization Toolset</p> <p>Terminal Text Formatting Using Escape Sequences</p> <p>Accessing Environmental Data from ERD's ERDDAP Server</p> <p>eXtensible Time Series</p>   |

|                               |  |
|-------------------------------|--|
| <a href="#">Packages</a>      | Downloads:   |
| <a href="#">Other</a>         |  |
| <a href="#">Documentation</a> | Reference manual: <a href="#">xtable.pdf</a>   |
| <a href="#">Manuals</a>       | Vignettes: <a href="#">xtable List of Tables Gallery</a>   |
| <a href="#">FAQs</a>          | <a href="#">xtable List of Tables Gallery</a>  |
| <a href="#">Contributed</a>   | <a href="#">xtable margintable</a>   |
|                               | <a href="#">xtable Gallery</a>   |
|                               | Package source: <a href="#">xtable 1.8-2.tar.gz</a>  |
|                               | Windows binaries: r-devel: <a href="#">xtable 1.8-2.zip</a> , r-release: <a href="#">xtable 1.8-2.zip</a> , r-oldrel: <a href="#">xtable 1.8-2.zip</a> |

2. 下载完成后，切换至高性能帐号中，在编译节点，进入/R-3.1.3/install 文件夹（原来在/R-3.1.3/install/bin 文件夹中）：**cd ../**
3. 创建 packages 文件夹：**mkdir packages**

```
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3/install> mkdir packages
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3/install> ls
bin  lib64  packages  share
```

4. 用 Xftp 客户端将已下载的压缩包上传至用户自己帐号中（前面创建的 packages 文件夹中）。



5. 进入 packages 文件夹：**cd packages /**
6. 用 R 语言来安装：**R CMD INSTALL xtable\_1.8-2.tar.gz**

```
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3/install/packages> R CMD INSTALL xtable_1.8-2.tar.gz
* installing to library '/home-gk/users/nsc731_TEST/Softwares/R/R-3.1.3/install/packages'
* installing *source* package 'xtable' ...
** package 'xtable' successfully unpacked and MD5 sums checked
** R
** data
** inst
** preparing package for lazy loading
** help
*** installing help indices
** building package indices
** installing vignettes
** testing if installed package can be loaded
* DONE (xtable)
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3/install/packages>
```

7. 完成后会生成，xtable 文件夹。

```
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3/install/packages> ls
xtable  xtable_1.8-2.tar.gz
```

8. 输入命令：**R**
9. 会有如下显示：

```

nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3/install/packages/xtable> R

R version 3.1.3 (2015-03-09) -- "Smooth Sidewalk"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-unknown-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>

```

10. 即 R 和 xtable 安装成功。

11. 根据上面提示，输入 **q()** 可以退出。

```

> q()
Save workspace image? [y/n/c]: n
nsc731_TEST@gk0110:~/Softwares/R/R-3.1.3/install/packages/xtable>

```

12. 可以选择打印上面的输出，也可以选择打印。

13. 还可以根据自己的需要安装其他的包。

## 五、测试运行

1. 用 FTP 工具将准备好的任务文件上传至预先创建的 TEST/R/ 目录中，需提前写 r 程序，\*.r 或 \*.R。

2. 切换至 Xshell 界面，进入 TEST/R 文件夹：**cd ~/TEST/R**

```

nsc731_TEST@gk0110:~> cd TEST/R/
nsc731_TEST@gk0110:~/TEST/R>

```

3. 用 vi 编辑器编写提交作业脚本文件 r\_serial.lsf( 以 **GK 分区** 为例；也可在 Windows 系统中写好以后用 FTP 工具上传)：**vim r\_serial.lsf**

4. 输入字母 “i” 进入编辑模式，将提交作业脚本的内容写入 r\_serial.lsf 文件中。脚本内容如下（**注：**脚本中红色部分需根据情况进行修改：

**第一处：**APP\_NAME=intelk\_small: 在不同的分区，需要选择对应分区的队列名称（其他队列其分区见附表 2）；



第二处: NP=1: 可以根据任务的需要进行修改( NP 与 APP\_NAME 的关系见附表 2 ), 若需要使用 R 并行, 则需要使用 parallel 包里的并行化方式函数;

第三处: /home-gk/users/nscc731\_TEST/...bin/R: R 软件安装时的绝对路径;

第四处: try1.r: 为输入文件; try1.output: 为输出文件。需根据计算任务名称进行修改。本指南安装的 GK 分区的测试脚本如下:

```
#!/bin/bash
APP_NAME=intelk_small
NP=1
NP_PER_NODE=12
RUN="RAW"

/home-gk/users/nscc731_TEST/Softwares/R/R-3.1.3/install/bin/R -f try1.r > try1.output
```

5. 按下键盘上的“Esc”键后, 输入“:wq”保存文件, 并退出。
6. 将脚本文件转换为 UNIX 格式(如从 Windows 系统上传的话必须要转换, 不然提交作业时会上报错误, 若提交作业的脚本名称不为 r\_serial.lsf, 则需要修改为对应的脚本名称): **dos2unix r\_serial.lsf**

```
nscc731_TEST@gk0110:~/TEST/R> dos2unix r_serial.lsf
dos2unix: converting file r_serial.lsf to UNIX format ...
nscc731_TEST@gk0110:~/TEST/R>
```

7. 赋予脚本文件可执行权限: **chmod +x r\_serial.lsf**

```
nscc731_TEST@gk0110:~/TEST/R> chmod +x r_serial.lsf
nscc731_TEST@gk0110:~/TEST/R>
```

8. 中心系统中用 bsub 命令提交作业脚本(在编译节点或者登录节点都可以提交): **bsub r\_serial.lsf**

```
nscc731_TEST@gk0110:~/TEST/R> bsub r_serial.lsf
Job <2370363> is submitted to queue <intelk_small>.
```

9. 如果提交正确, 则会出现如下内容(其中 Job 后面的数字为: JobID, 在计算出现问题时, 可以根据 JobID 查看该任务情况)。
10. 查看任务是否计算完成, 可以使用 bjobs -l +JobID 命令(当出现: Done successfully. The CPU time used is xxxx seconds 说明计算结束): **bjobs -l 2370363**

```
nscc731_TEST@gk0110:~/TEST/R> bjobs -l 2370363
Job <2370363>, User <nscc731_TEST>, Project <default>, Application <gkapp>, Sta
```

11. 任务计算结束后, 可以查看输出文件, 检查任务是否计算成功: **more try1.output**



```
nssc731_TEST@gk0110:~/TEST/R> more try1.output
```

12. 如果计算成功，在输出文件 try1.output 的中会出现需要的结果：

```
> re
[1] 7.185942 -173.213454 -3.587471 -1.363450 4.431254 -1.385386
[7] -1.274224 -2.505717 -2.815654 2.405442 4.547630 3.842978
[13] -2.305223 -45.332782 -1.834273 -5.662550 -2.103937 -1.176640
[19] -60.435547 -1.795964 -1.297844 4.723706 -2.599209 3.040167
[25] 7.302232 -40.353779 2.073382 -50.735549 -1.463356 26.709204
[31] -6.443010 -3.177243 -5.841788 4.052410 -1.604754 -2.445886
[37] -1.725536 2.832670 -2.068708 -16.957299 -1.620822 -2.629291
[43] -1.773400 -3.967022 -4.547971 -1.745436 2.131180 -6.715766
[49] -2.018349 -2.266462 -6.700263 -1.494345 -10.501844 5.983390
[55] 2.480157 2.741893 14.240726 -4.880854 -2.503574 -18.906056
[61] -1.349937 9.882885 -8.062321 20.720399 -2.602065 6.690047
[67] -6.651516 -1.846415 2.592088 -1.451201 25.544723 -1.563664
[73] 34.691162 21.607583 -6.783233 -1.406541 -1.479437 -33.280541
```

13. 计算完成后，用 FTP 工具将文件下载后，进行分析。

附：

表 2 各分区计算队列（APP\_NAME）及核数（NP）

| 计算分区 | 计算队列（APP_NAME） | 队列说明（NP）      | NP_PER_NODE |
|------|----------------|---------------|-------------|
| GG   | intelg_small   | 每节点核数<12      | 小于等于 12     |
|      | intelg_mid     | 核数 12~120     | 建议为 12      |
|      | intelg_long    | 核数大于 120      | 建议为 12      |
| GK   | intelk_small   | 每节点核数<12      | 小于等于 12     |
|      | intelk_mid     | 核数 12~120     | 建议为 12      |
|      | intelk_long    | 核数大于 120      | 建议为 12      |
| GS   | intels_small   | 每节点核数<12      | 小于等于 12     |
|      | intels_mid     | 核数 12~120     | 建议为 12      |
|      | intels_long    | 核数大于 120 核的作业 | 建议为 12      |
| FN   | AMD_small      | 每节点核数<32      | 不指定或≤32     |
|      | AMD_mid        | 核数 32~256     | 建议为 32      |

|    |              |            |         |
|----|--------------|------------|---------|
|    | AMD_long     | 256<=核数    | 建议为 32  |
| YW | intelw_small | 每节点核数<12   | 小于等于 12 |
|    | intelw_mid   | 核数 24~120  | 建议为 12  |
|    | Intelw_long  | 核数 120-960 | 建议为 12  |
|    | intelw_exc   | 核数 12~6000 | 建议为 12  |
| YS | intels_small | 每节点核数<12   | 小于等于 12 |
|    | intels_mid   | 核数 12~120  | 建议为 12  |
|    | Intels_long  | 核数 96-960  | 建议为 12  |
|    | intels_exc   | 核数 12~6000 | 建议为 12  |
| YG | intels_small | 每节点核数<12   | 小于等于 12 |
|    | intels_mid   | 核数 12~120  | 建议为 12  |
|    | Intels_long  | 核数 120~960 | 建议为 12  |

#### 脚本参数含义:

**APP\_NAME:** 指定作业运行使用的队列名称，需根据帐号所在分区及 NP 的值进行选择；

**NP:** 指定作业运行需要的核数，如果该参数不指定，则缺省值为 1 个核，最大不能超过所开帐号的最大权限核数；

**NP\_PER\_NODE:** 指定每个节点上最多分配给作业运行的核数。除 FN 分区最大值为 32，其他分区最大值为 12。FN 分区如果不指定此参数，系统默认会对 AMD\_long 队列以 NP\_PER\_NODE=32 来分配资源，对提交到 AMD\_small 和 AMD\_mid 队列中的计算作业，则会搜寻所有可用的资源，每节点上的核数有可能不相等。其他分区必须指定该参数。

**注:** 如有问题请发邮件至: [service@nscsz.gov.cn](mailto:service@nscsz.gov.cn)