

dy的生信学习笔记

- 一.Linux
- 二.python&R 常见数据处理
- jupyter
- 单细胞分析
- GWAS&QTL

dy的生信学习笔记

一.Linux

常见中值得注意的

```
1 top （查看cpu使用情况）
2 ssh -L localhost:[number]:[remoteip]:[number] 服务器地址 账号 （可在本地端口运行服务器端口的程序）
3 rm -rf （强制删除文件夹）
4 conda search + conda install 很好用!!!
5 ls -hl （可以以K,M,G的单位显示当前文件夹大小）#没法显示子文件夹里面的内存
```

补：

```
1 #conda是个p, mamba才是王道
2 mamba search + mamba install!
```

装R包的tips（我放到了python&R 常见数据处理那一部分）

处理文本的工具awk

```
1 这里用一个例子来说明
2 cat pruned_coatColor_maf_geno.vcf | awk 'BEGIN{FS="\t";OFS="\t";}/#{next;}
  {{if($3==".")$3=$1:"$2;}print $3,$5;}' > alt_alleles
```

看起来很复杂，一步一步看

"|"代表左边的输出结果直接作为右边的输入，即cat读取文件后传入右边的awk，故awk的最后一个参数（输入文件）省去了

awk实际是逐行读取文件，awk -F 可以选择分隔的符号。awk -F [,.]可以先空格分隔，再','分隔一次

awk 'BEGIN{' 可以填写在读取文件前的操作，如print。此处设置FS与OFS都等于"\t"（读取文件分隔符和输出所用分隔符）

/#{代表正则表达式含"#"的部分，若该行满足条件，则运行后面的next，跳过该次循环（即进入下一行），否则继续运行后面的部分

if(\$3 == "."),若第三列是".",则 使第三列等于第一列加": "加第三列，然后输出第三列和第五列

最后的>,非常常用的重定向，写入文件

如果报错backslash not last character on line，记得删掉\$3这种变量或引号前面的反斜杠（直接运行不要加，提交作业可能要加，巨sb）

一种选择多文件操作的方法

```
1 files="./set*.vcf"    #注意shell语言中'='的两边不要有空格！
2 file=`ls files | head -n $id | tail -n 1`
```

ls files会将满足正则的**文件**全部列出，随head -n 次数的累加与tail，每轮依次选择不同文件。id需是可以循环累加或类似于 \$SLURM_ARRAY_TASK_ID这种变量

文本工具sed

```
1 sed 's/.gz//g'
```

s 为更改，即：将.gz更改为/g'(应该是空格)

使用软件的方法（软链接）

以存放在/home/path/路径的tool软件为例

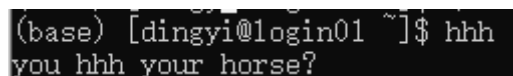
```
1 /home/path/tool 即可运行
2 可使用软链接放入bin目录中
3 cd xxx/bin
4 ln -s /home/path/tool atool #软链接，后面是快捷用法
5 atool #即可
6
7 ll可查看目录下的软链接
8 which可查看命令存储的位置（类似这里软链接）
```

自定义命令的方法

例：想自定义一个叫“work”的命令，作用是切换到“/home/work”目录并检索目录，同时输出一段话

```
1 vim ~/.bashrc
2 #在.bashrc中写入
3 alias work="cd /home/work && ls && echo ""It is time for work!"" " #两个双引号
  是为了区别单个双引号（也可使用单引）
4 #退出.bashrc
5 source ~/.bashrc
```

一个自定义命令的截图（don't mind the details）



```
(base) [dingyi@login01 ~]$ hhh
you hhh your horse?
```

二.python&R 常见数据处理

当读取很大的datafram文件时

```

1 | scipy.io.mmread
2 | scipy.io.mmwrite("xxx.mtx") (写为稀疏矩阵)
3 |
4 | df.to_hdf('xxx.h5', key='df')
5 | pd.read_hdf('xxx.h5', key='df')

```

```

1 | data.table::fwrite() 速度较快
2 |
3 | as(data.matrix(df), 'dgCMatrix')
4 |
5 | library(Matrix)
6 | writeMM(obj, file="xxx.mtx")
7 | readMM()

```

R值得注意的

```

1 | keeplist <- c()
2 | for (i in 1:20){
3 |     if (xxx){
4 |         keeplist <- c(keeplist,i)
5 |     }
6 | }
7 | #python的append在R中类似的用法
8 |
9 | list <- 1:5
10 | paste0("x_", list)
11 | #输出
12 | > x_1 x_2 x_3 x_4 x_5
13 |
14 | 想输出字符串的长度，请用nchar!! 不要用length
15 | length(c('aaa', 'bbcd'))#输出字符串的个数
16 | > 2
17 | nchar(c('aaa', 'bbcd'))#字符串长度
18 | > 3 4

```

R装包tips (保你能装上)

以R包 plink2R 为例 (阴间包)

```

1 | 1.按惯例试试 install.packages("plink2R") , 不行就用
   | BiocManager::install("plink2R")
2 | 2.上面两个要是都不行, 就只能用conda/mamba search +conda/mamba install
3 | 3.conda也找不到的话, 那基本就是小众到离谱的阴间R包了。去你看到这个包的网站(rcran、github
   | 等等), 把R包文件夹手动下载下来, 然后用devtools::install_local("path/xxx") (两个
   | 都行)
4 | 在这里, 我从github上下载了master.zip, 解压后发现其中的Plink2R子文件夹是所需的R包, 于是
   | devtools::install_local("xxx/master/plink2R"), 成功安上

```

📁 R	Added imputation of missing genotypes by sampling proportional to the
📁 man	initial commit
📁 src	Removed old dependency on mmap
📄 DESCRIPTION	Removed old dependency on mmap
📄 NAMESPACE	initial commit

这是子文件夹plink2R里面的模样，R包一般需具有R代码文件、description（必需）、其他文件。

运行截图：

```
devtools::install_local("/storage/yangjianLab/dingyi/tools//plink2R-master/plink2R/")
```

Skipping 2 packages not available: RcppEigen, Rcpp

— R CMD build —

```
* checking for file '/tmp/RtmpTGDBNP/file375441688a1c2c/plink2R/DESCRIPTION' ... OK
* preparing 'plink2R':
* checking DESCRIPTION meta-information ... OK
* cleaning src
* checking for LF line-endings in source and make files and shell scripts
* checking for empty or unneeded directories
* building 'plink2R_1.1.tar.gz'
Warning in sprintf(gettext(fmt, domain = domain), ...) :
  one argument not used by format 'invalid uid value replaced by that for user 'nobody'
Warning: invalid uid value replaced by that for user 'nobody'
```

jupyter

在新的账号上安装jupyter（base环境）

```
1 conda/pip install jupyter
2 conda/pip install jupyterlab
3 jupyter notebook --generate-config #生成 ~/.jupyter/jupyter_notebook_config.py 文件
4 jupyter notebook password # 然后输入想设置的密码并重复密码，之后即可打开文件
  ~/.jupyter/jupyter_notebook_config.json，复制文件中保存的密钥xxxxxxx（是一串字符）
5 #打开 ~/.jupyter/jupyter_notebook_config.py 文件，加入
6 c.NotebookApp.allow_remote_access = True #允许远程访问
7 c.NotebookApp.ip = '*' #似乎 '*' 或者 '0.0.0.0' 效果一样
8 c.NotebookApp.password = u'xxxxxxx' #这里是刚才的哈希密码（那一串字符）
9 c.NotebookApp.open_browser = False #不打开浏览器
10 c.NotebookApp.port = 8888 #这个端口是只运行命令 jupyter notebook 时的默认端口，可以通过
   命令 jupyter notebook --port XXX 来设置端口
```

运行命令改为 `jupyter lab` 即可使用 `lab`，运行 `notebook` 后，在本地浏览器链接后加上 `lab` 同样可以 如 `localhost:2000/lab`

创建多个kernel

R

```
1 install.packages("IRkernel") #不行就用conda
2 IRkernel::installspec(name = 'ir33', displayname = 'R 3.3')
```

python

```
1 pip install ipykernel #conda 也行
2 python -m ipykernel install --user --name py_38
```

单细胞分析

一些细胞种类的marker

B cells:CD19,MS4A1

plasma cells(浆细胞):IGHG1,CD79A

CD4+T cells :CD3D,CD4

CD8+T cells:CD3D,CD8A

natural killer cells(NK): NVR1, FGFBP2

myeloid cells(髓细胞):CD14,CD68

mast cells(肥大细胞):TPSAB1,TPSB2

endothelial cells(内皮细胞):RAMP2,PECAM1

fibroblasts(成纤维细胞):DCN,LUM

mural cells(壁细胞):PDGFRB,ACTA2

glial cells(胶质细胞):PLP1,SOX10

epithelial cells(上皮细胞):PGC,PGA3

GWAS&QTL

文件格式

0代表缺失

*.ped

FID	IID	PID	MID	Sex	P	rs1	rs2	rs3
1	1	0	0	2	1	CT	AG	AA
2	2	0	0	1	0	CC	AA	AC
3	3	0	0	1	1	CC	AA	AC

*.map

Chr	SNP	GD	BPP
1	rs1	0	870000
1	rs2	0	880000
1	rs3	0	890000

*.fam

FID	IID	PID	MID	Sex	P
1	1	0	0	2	1
2	2	0	0	1	0
3	3	0	0	1	1

*.bed

Contains binary version of the
SNP info of the *.ped file.
(not in a format readable for
humans)

*.bim

Chr	SNP	GD	BPP	Allele 1	Allele 2
1	rs1	0	870000	C	T
1	rs2	0	880000	A	G
1	rs3	0	890000	A	C

Covariate file

FID	IID	C1	C2	C3
1	1	0.00812835	0.00606235	-0.000871105
2	2	-0.0600943	0.0318994	-0.0827743
3	3	-0.0431903	0.00133068	-0.000276131

Legend

FID	Family ID	rs{x}	Alleles per subject per SNP
IID	Individual ID	Chr	Chromosome
PID	Paternal ID	SNP	SNP name
MID	Maternal ID	GD	Genetic distance (morgans)
Sex	Sex of subject	BPP	Base-pair position (bp units)
P	Phenotype	C{x}	Covariates e.g. Multidimensional Scaling (MDS) components

Plink

--geno 筛选variants

--mind 筛选samples

osca

--eqtl:

- --bfile: *reads individual-level SNP genotype data (in PLINK binary format), i.e. .bed, .bim, and .fam files.*
- --bfile: *to input molecular phenotypes (e.g. DNA methylation or gene expression measures in BOD format).*
- --qcovar : *reads quantitative covariates from a plain text file*
- something else

--sqt (THESTLE 工具): 与 eqtl类似, 看说明文档就好

也可直接输入qtl summary文件

--meta 用来合并来自同一cohort的多个qtl结果文件

--Mecs 合并不同cohort的qtl文件