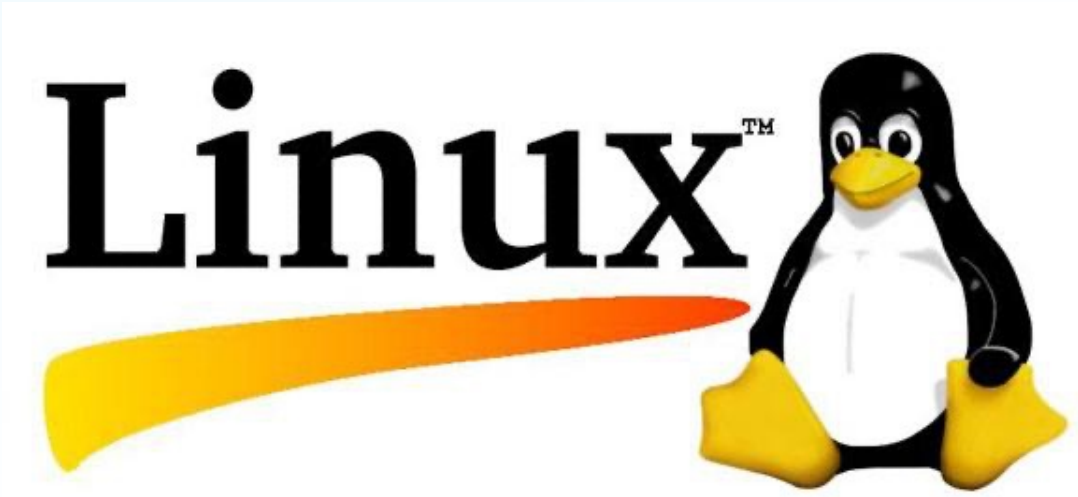
易生信——最懂你的生信培训，学习生信更容易

Linux服务器软件安装



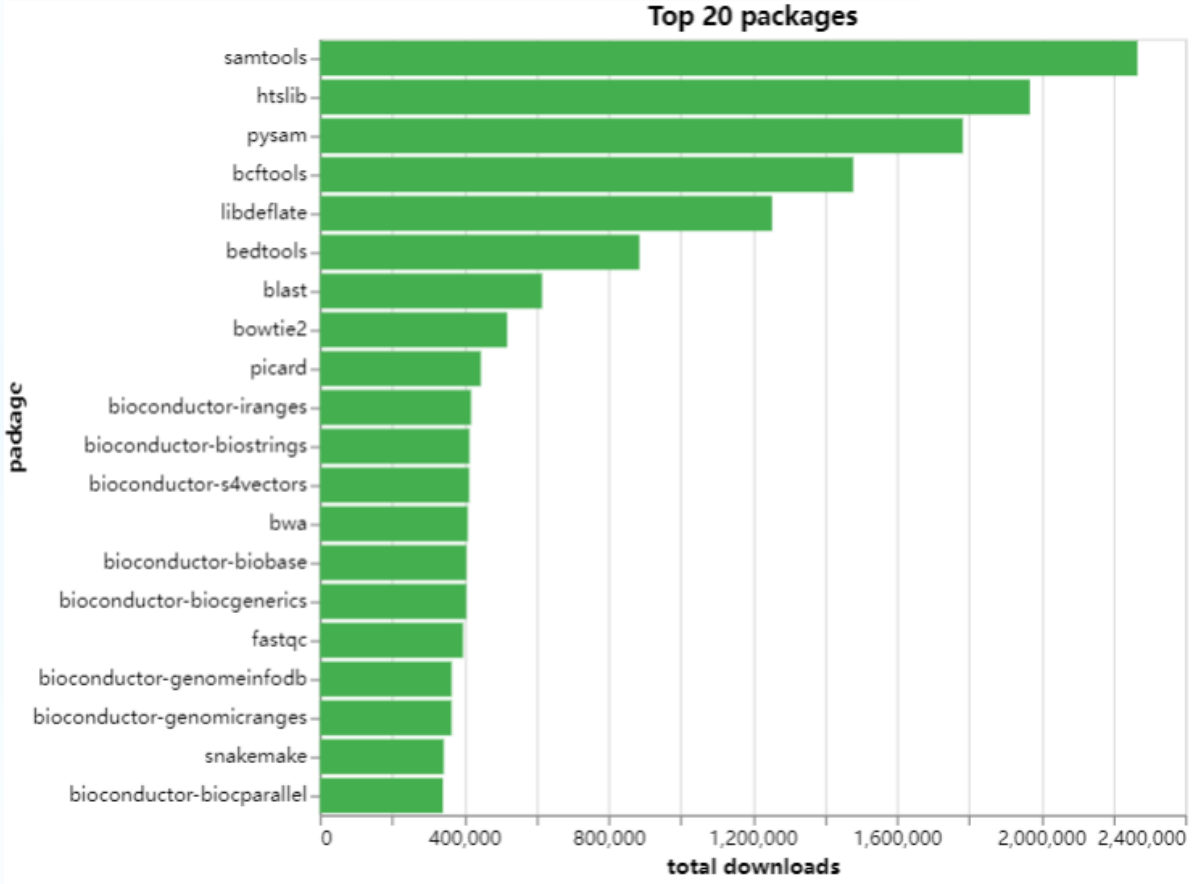
Linux服务器软件列表

- Fastq-dump (SRA toolkit): 公共数据下载
- Fastqc：质量评估
- MultiQC：结果整合
- Trimmomatic/Fastp：reads预处理
- Salmon：快速准确不比对定量工具
- Conda：包管理器
- Samtools：sam文件处理工具
- UCSC toolkit: 文件格式转换工具集合
- STAR：转录组比对工具
- RSeQC：转录组比对质量评估工具
- Stringtie：转录本拼装工具
- rMATS：可变剪接分析工具
- Rmats2sashimiplot：可变剪接可视化工具
- CPC2: lncRNA预测
- htseq: reads count 计算
- bedtools: 生产igv batch script

Conda安装

Conda软件包管理神器

- conda：任意语言的软件包、环境、依赖关系的开源管理系统。
- Anaconda：集合了常用Python包的数据科学平台
- Miniconda：精简版Anaconda，只包含conda和Python
- bioconda：conda的一个通道，含数万生信分析软件和版本收录，文章发表于Nature Method



推荐使用miniconda，可以获得最新版

- wget -c https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
- # -p 指定安装路径
- # -b -f 不做提示，直接安装
- bash Miniconda3-latest-Linux-x86_64.sh -b -f -p ${HOME}/miniconda3

Conda增加通道

- conda config --add channels defaults
- conda config --add channels bioconda # 增加软件支持
- conda config --add channels conda-forge # Highest priority

- conda config --set show_channel_urls yes

Conda增加国内通道（可选）

- conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/msys2/# Anocanda清华镜像，国内镜像，加速下载
- conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
- conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
- conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
- conda config --set show_channel_urls yes

创建新的运行环境

- 创建transcriptome环境，指定使用的R和python版本

  conda create -y -n transcriptome

- 激活新环境transcriptome

  source activate transcriptome

- 退出transcriptome环境

  source deactivate transcriptome

激活环境、按顺序安装软件

- 在新环境transcriptome中安装软件

```
# 先激活环境，注意安装顺序
conda install -y samtools multiqc fastqc star
conda install -y stringtie trimmomatic
conda install -y rmats
conda install -y rnasamba
conda install -y rseqc
conda install -y salmon
conda install -y bedtools htseq
```

如果conda慢，可用Mamba提速

- conda install –y mamba -c conda-forge
- mamba install -y -q samtools multiqc rseqc fastqc salmon star stringtie sra-tools trimmomatic rmats rmats2sashimiplot

[一文掌握Conda软件安装：虚拟环境、软件通道、加速solving、跨服务器迁移]

如何提速Conda

- 采用最新版的 `conda` (Conda4.7相比Conda4.6提速**3.5**倍, Conda 4.8应该不会比4.7慢)
- 安装时指定版本减少搜索空间 `conda install python=3.7.4`
- 安装R包时指定R的版本也会极大减小搜索空间 (R包因其数目众多，也是生物类软件依赖解析较慢的原因之一) `conda install r-base=4.0.2 r-ggplot2=3.3.2`
- 采用 `mamba` 加速软件依赖解析 [mamba采用 `c++` 重写了部分解析过程，这个提速效果是很明显的] (安装好 `mamba` 后就可以用 `mamba` 替换 `conda` 进行安装了)

```
conda install mamba -c conda-forge
mamba install python=3.7.4
```

判断可执行文件的位置和当前所在环境

- which python返回python的位置
- 激活conda的环境在终端会有标识

```
(metagenome_env) amplicon@localhost:~$ which python
/anaconda2/envs/metagenome_env/bin/python
(metagenome_env) amplicon@localhost:~$ source deactivate metagenome_env
amplicon@localhost:~$ which python
/usr/bin/python
amplicon@localhost:~$ source activate metagenome_env
(metagenome_env) amplicon@localhost:~$
```

Conda-pack加载我们构建好的环境

```
# bash /mnt/c/Miniconda3-latest-Linux-x86_64.sh -b -f
# 加载环境
~/miniconda3/condabin/conda init
source ~/.bashrc
# Unpack已有镜像    # 从QQ群下载transcriptome.env.tar.gz    # 如果是Win10+ubuntu则 放置在 C盘根目录下
# 新建文件夹存放transcriptome环境
mkdir -p ~/miniconda3/envs/transcriptome
# 解压环境
tar -vxzf /mnt/c/transcriptome.env.tar.gz -C ~/miniconda3/envs/transcriptome
# 激活环境
source ~/miniconda3/envs/qiime2/bin/activate
conda-unpack
```

几个需要注意的概念

Linux下的软件

- 软件类型
  - 脚本
  - 二进制程序
  - Java包
- 可执行属性
  - 软件或脚本需要有执行权限 `chmod a+x soft_name`
- 环境变量
  - 告诉系统软件可能在的位置或使用完整路径

什么样的软件？

- **脚本型**

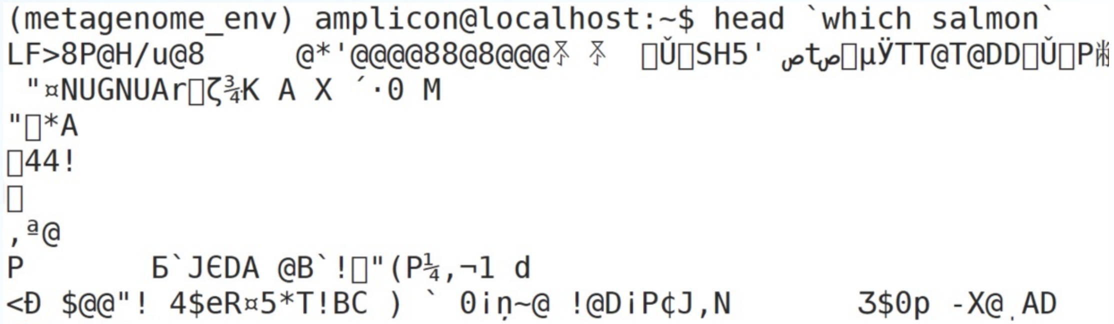
  解释型语言写作，如Bash，R，Python，Perl等，源代码可直接查看

```
(metagenome_env) amplicon@localhost:~$ head `which kraken`
#!/usr/bin/env perl

# Copyright 2013-2015, Derrick Wood <dwood@cs.jhu.edu>
#
# This file is part of the Kraken taxonomic sequence classification system.
#
# Kraken is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
```

- **二进制型**

  源代码编译成机器语言，直接打开查看乱码，如bwa, salmon等



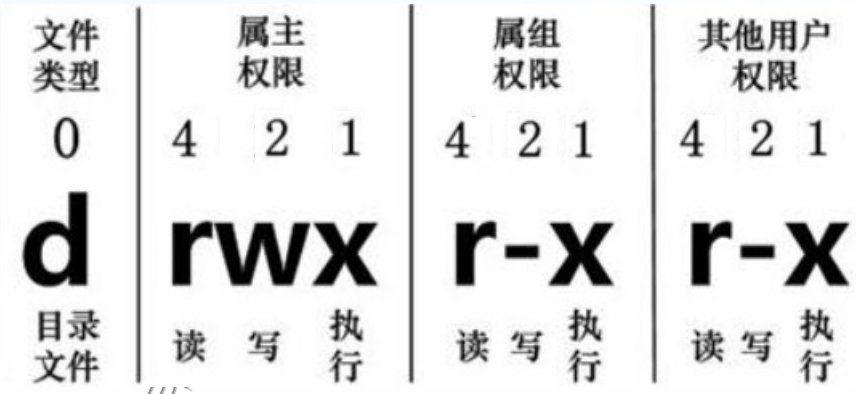
- **Java程序**

  java -jar trimmomatic-0.36.jar

可执行属性 – 软件的必须属性

- ls –l 查看文件的属性（文件夹的可执行属性是可读属性）
- 一般在终端不同颜色对应不同属性
- chmod修改属性
- chmod a+x file # 所有人增加可执行属性
- chmod 755 file # 所有人可执行，自己可写

```
drwxr-xr-x  4 ct ct    32 10月 14 10:21 binning
-rw-r--r--  1 ct ct 15702 10月 18 16:33 metagenome_softinstall_ln_wgt.sh
-rwxr-xr-x  1 ct ct 19272 10月 14 10:21 pipeline.sh
drwxr-xr-x 12 ct ct   190 10月 14 10:21 result
drwxr-xr-x  2 ct ct   258 10月 14 10:21 seq
-rw-r--r--  1 ct ct  5804 10月 14 10:21 soft_db.sh
drwx------  2 ct ct     6 10月 18 13:48 temp
lrwxrwxrwx  1 ct ct     4 10月 18 20:24 temp2 -> temp
```



环境变量PATH – 软件所在目录的集合

- 环境变量PATH是一堆目录，一堆**存放有软件的目录。**
- 在系统接到命令输入比如”cd”后，会去环境变量PATH存储的目录中从前向后查找，在哪个目录发现存在输入的命令同名”cd”的文件视为找到程序，然后判断是否有可执行属性，如果有则执行。
- echo $PATH
- export PATH=$PATH:~/soft

```
ct@localhost:/db/meta$ echo $PATH
/self_bin:/disk2/bin:/anaconda2/bin/:/usr/lib64/qt-3.3/bin:/disk2/home/ct/perl5/bin:/usr/local/bin:/usr/bi
n:/usr/local/sbin:/usr/sbin:/disk2/bin/:/anaconda2/bin:/opt/Cytoscape_v3.5.1/:/disk2/soft/rsem/bin:/disk2/
soft/bin:/disk2/home/ct/bin
ct@localhost:/db/meta$ export PATH=$PATH:~/soft
ct@localhost:/db/meta$ echo $PATH
/self_bin:/disk2/bin:/anaconda2/bin/:/usr/lib64/qt-3.3/bin:/disk2/home/ct/perl5/bin:/usr/local/bin:/usr/bi
n:/usr/local/sbin:/usr/sbin:/disk2/bin/:/anaconda2/bin:/opt/Cytoscape_v3.5.1/:/disk2/soft/rsem/bin:/disk2/
soft/bin:/disk2/home/ct/bin:/disk2/home/ct/soft
```

PATH和path，傻傻分不清

```
YSX@ehbio:~/train/single_cell$ pipeline_metagenome.sh
-bash: pipeline_metagenome.sh: 未找到命令
```

`pipeline_metagenome.sh` 命令去哪儿了？上面我们都看到了，就在 `metagenome` 目录下，为啥电脑（操作系统）这么笨却找不到？另外为什么运行 `head` 就可以找到？难道有一些黑魔法在里面？

确实是有一些黑魔法的，不过我们一般称之为**规则**。

操作系统为了便捷性和安全性，定义了一系列环境变量，存储常用信息，`PATH` (注意全是大写)是其中一个。

`PATH`：是存放有(可执行)命令和程序的目录集合；在操作系统接到用户输入的命令时，会对PATH**存储的目录**进行查找，看下是否有与用户输入的命令同名的文件存在，而且是**从前到后**一个个查找，而且是**查到就停**，最后查不到就报错。（从这几个**加粗**的文字，可以看到操作系统很懒，当然懒是好的程序员的必备属性。）

环境变量 – 永久设置

- 服务器自己用户的家目录下一般有2个隐藏文件：.bashrc和.bash_profile。
- .bashrc本地登录时读取。
- .bash_profile远程登录时读取。
- .bashrc和.bash_profile是bash脚本，可以写任何bash命令。
- 需要把环境变量设置命令写入.bash_profile中。

不同类型的“环境变量”

- 环境变量PATH：定义可执行程序的目录
- LD_LIBRARY_PATH：定义动态库的目录 (.so文件not found)
- PYTHONPATH：定义Python包的目录
- PERL5LIB：定义Perl模块的目录

传统软件安装方法

编译好的二进制文件

- 编译好的多平台通用二进制文件或特定平台可用二进制文件，下载，解压，增加可执行属性，放入环境变量，直接调用。
- 认真看软件说明手册，如果提供了二进制版本，尽量使用二进制版本，简单方便，把时间多放在数据上，而不是软件安装上。
- 一般可执行程序放置在 bin 目录下。

```
wget ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/ncbi-blast-2.7.1+-x64-lir
tar xvzf ncbi-blast-2.7.1+-x64-linux.tar.gz
cd ncbi*
cd bin
# 直接进入 bin 目录，找到对应可执行文件，链接到在环境变量的目录中去。
# 具体可看视频的操作  http://bioinfo.ke.qq.com
ln -s `pwd`/* ~/bin
```

经典的源码安装

- ./configure && make && make install
- ./configure是检测系统的库文件、头文件、依赖的软件是否存在及版本是否兼容，并根据检测结果生成Makefile文件。这一步是软件安装是否能成功的关键，检测通过安装一般没问题。检测不通过，缺什么补什么。如果非根用户，这一步通常也会配置下软件安装的路径--prefix=/home/ct/soft/specific_name。
- make具体的编译过程，根据Makefile中的规则把程序语言转换为机器语言。
- make install拷贝make编译出的可执行文件或依赖的动态库到prefix指定的目录。
- 置入环境变量即可使用。

```
wget https://jaist.dl.sourceforge.net/project/samtools/samtools/1.7/samtools-1.7.tar.bz2
tar xvzf samtools-1.7.tar.bz2
cd samtoo*
./configure --prefix=/home/ct/soft/samtools
make
make install
cd /home/ct/soft/samtools/bin
ln -s `pwd`/* ~/bin
```

Python包的安装

- Python包管理器安装

  easy_install package_name

  pip install package_name -i https://pypi.tuna.tsinghua.edu.cn/simple/

- Python包手动源码安装

  python setup.py build

  python setup.py install

- Conda安装

  conda install package_name

软件和数据库下载

- wget -c soft_url/database_url # -c断点续传
- wget -c ftp://ftp.ncbi.nlm.nih.gov/blast/db/nt.*.tar.gz 支持通配符
- wget -cr -np -nd ftp://ftp.ncbi.nlm.nih.gov/blast/db/ 递归下载
- 同类工具还有curl和axel，都可以用

SRA toolkit安装

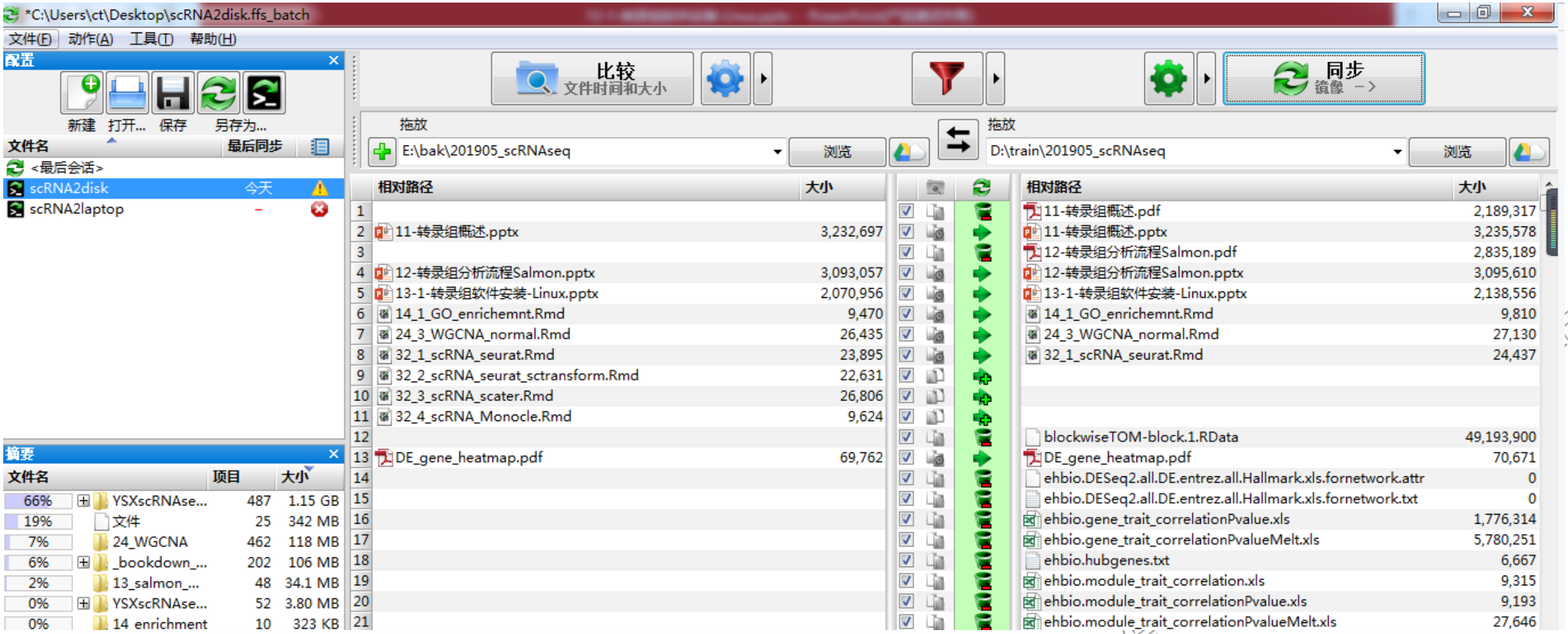
- wget -c https://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/2.9.0/sratoolkit.2.9.0-centos_linux64.tar.gz
- tar xvzf sratoolkit.2.9.0-centos_linux64.tar.gz
- ln -s `pwd`/sratoolkit.2.9.0-centos_linux64/bin/fastq-dump ~/transcriptome/soft
- # 若运行成功，则输出为 ~/transcriptome/soft/fastq-dump
- which fastq-dump

数据备份 rsync

- rsync则是一个增量备份工具，只针对修改过的文件的修改过的部分进行同步备份，大大缩短了传输的文件的数量和传输时间。

```
# 把本地project目录下的东西备份到远程服务器的/backup/project目录下
# 注意第一个project后面的反斜线，表示拷贝目录内的内容，不在目标目录新建project文件夹。
# -a: archive mode, quals -rlptgoD
# -r: 递归同步 -p: 同步时保留原文件的权限设置
# -u: 若文件在远端做过更新，则不同步，避免覆盖远端的修改
# -L: 同步符号链接链接的文件，防止在远程服务器出现文件路径等不匹配导致的软连接失效
# -t: 保留修改时间 -v: 显示更新信息 -z: 传输过程中压缩文件，对于传输速度慢时适用
rsync -aruLptvz --delete project/ user@remoteServerIP:/backup/project
```

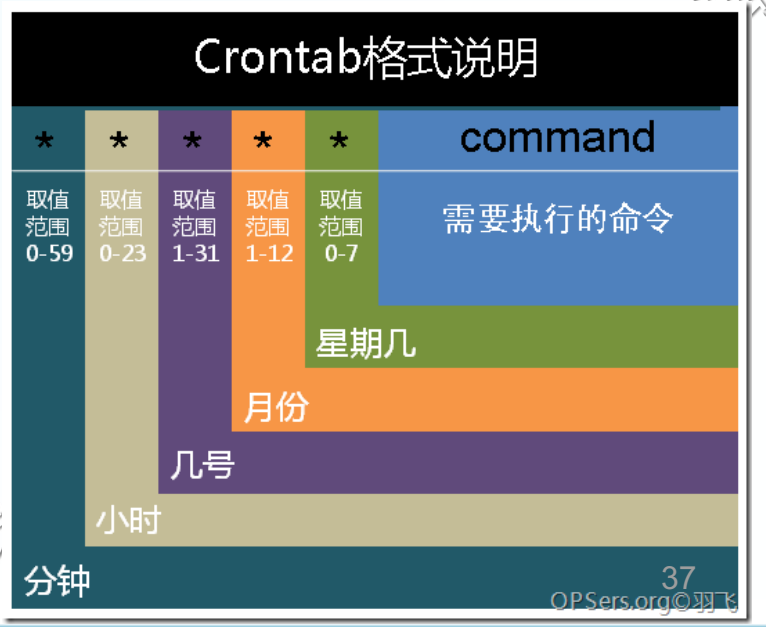
Freefile sync 跨平台的数据同步工具



定时备份

- crontab -e # 打开下面的编辑vim界面

```
#minute hour    day     month   week    command
0       0       */3     *       *       rsync.sh at 00:00 every 3 days
0       6       *       *       *       Call me at 6:00 everyday
*/20    6-18    *       *       *       Run every 20 minutes in 6-18
```



扫码关注生信宝典，学习更多生信知识

扫码关注宏基因组，获取专业学习资料

易生信，没有难学的生信知识