

WindowsでDRY解析する奴～www

DRY解析教本、まずmacを買わないと先に進ませないみたいな雰囲気だしてるからWindowsでDRY解析したったwww

1. Ubuntu で Linux を入れる

1-1. Windows を Ubuntu が利用可能なバージョンまでアップデートしていることを確認する（最新なら大丈夫）。バージョンは 設定 > システム > （左のタブの）詳細情報 > Windowsの仕様 > バージョン から（20H2とかならOK）。

1-2. WLS (Windows Subsystems for Linux) を可能にする。
設定 > アプリ > （アプリと機能の凄く下の方にある）プログラムと機能 > （左のタブの）Windowsの機能の有効化または無効化（管理者権限が必要） > Linux用 Windowsサブシステム にチェックを入れて”OK”を押す。

1-3. Ubuntu をインストールする。
Microsoft Store（買い物かごみたいなやつ）を起動する。検索窓にUbuntuと入れて、シンプルな”Ubuntu”を見に行く。”入手”して（マイクロソフトアカウントはなくても大丈夫なはず）、”インストール”、”起動”。しばらく待つ。

1-4. UNIXのアカウントづくり
Enter new UNIX username：好きな名前
New password：好きなパスワード（入力しても表示されないので注意） Retype
成功すれば晴れてLinuxが使えるようになります。パスワードは忘れないように（結構使うっぽいので）

2. Linux コマンドを使ってみる（Level 1 準備編 の内容）

2-1. Windowsのファイルシステムと同じ場所の方がいろんな意味で分かりやすいので、Ubuntuを起動したらまず

```
cd /mnt/c
```

でCドライブ（もしDドライブがいいなら/mnt/d）へ作業してる場所を移す。cdはchange directoryのことで、/mnt/cはUbuntuが用意してくれたWindowsのファイルシステムへのマウント（つまり、LinuxがWindowsCドライブを一つのディレクトリとして認識できるようにしてくれているということ）。ちなみにDRY解析は多分凄く大量のデータを扱うので、容量の大きい場所にしたほうがいいです。
次に

```
ls
```

と打つと、現在のフォルダ（Cドライブ）の中のファイルやディレクトリが表示される。

dry解析用に新しくディレクトリを作ったほうがいいので、

```
mkdir 好きなディレクトリ名(以降は dry とする)
```

mkdir = make directory　そしてそのまま新しく作ったディレクトリへと cd する。

ここから先毎回Ubuntuを開いて作業を始めるときは

```
cd /mnt/c/dry
```

としてから始めるようにすればよい。

2-2. 後はdryの本と同様に種々のコマンドに触れてみる

```
mkdir kihondir
cd kihondir
export > testfile
cp testfile newfile
mv newfile newfile.txt
curl http://togows.org/api/ucsc/hg38/refGene/1,100 > newfile.txt
less newfile.txt　(qで終了する)
rm newfile.txt
rm testfile
cd -　(このコマンドで、cdしてきたディレクトリに戻る事が出来る)
rmdir kihondir
```

スペースの打ち間違いに注意

2-3. シェルスクリプト

Windowsのファイルシステムからも開けることが利用できます。つまり、Windowsの側で c://dry/ というディレクトリが存在するはずで、そこにWindowsで新しいファイルを作ってもLinux側に反映されます（lsなどで確認できる）。ただし、両方で起動や読み書きを混線させると重大なインシデントが起きる可能性があるので、やめてください。片方でいじり終えてからもう片方を動かすように心がける！

Win で dry/test.txt を作成し、教科書に書いてあるスクリプトを写して保存、名前を test.sh へ書き換える (Linux上でmvを使って！Winでやるとダメ)。これをUbuntuで

```
./test.sh
```

とすれば、ICHIBAN NIBAN が表示されるはず。

3. 諸々をインストールする (Level 1 準備編 の内容)

3-1. テキストエディタ

教科書は mac についた Xcode とかを入れるみたいですが、Windows なら Visual Studio Code か 類似した軽めのソースコード書けるタイプのソフト入れてください。Qiitaの記事とか見れば大体どうにかなるはず (vim, emacs など vscode で代替できる)

3-2. 諸々のインストールの前に

Ubuntu のパッケージのインストールの際には、アメリカのサーバーへ取りに行くようで、それだと遅いらしいので、

```
sudo sed -i -e  
's%http://.*.ubuntu.com%http://ftp.jaist.ac.jp/pub/Linux%g'  
/etc/apt/sources.list
```

とUbuntuで打って（改行ナシ！表示の都合で改行になってるだけ）、パスワード入れて日本のサーバーに変えましょう。まあ後で出てくるHomebrewを使うようになったらあんまり関係ないんですけどね

参考：<https://qiita.com/rubytomato@github/items/fdfc0a76e848442f374e>

3-3. パッケージをたくさん入れよう

Homebrewを入れるために、まず 諸々のパッケージを入れてしまいます。

```
sudo apt update && sudo apt upgrade
```

これにより、大量になんかいい感じのパッケージが入ります。途中で [Y/N] 的なのを聞かれると思うので、Y で返しておいてください。黄金のコマンド的な奴なので、多分これからもお世話になるかもしれないですね。

sudo は superuser として do する ということらしく、簡単にいえば何でもできるモードとしてやります、という意味（スーパーユーザモード）。apt は Ubuntu に付属した Advanced Packing Tool というソフトの略で、パッケージ（Windows の”ソフト”的な）を入れることが出来ます。

多分普通に長時間かかると思うので、休んでいてください。でもこれが終わればいろんな機能が追加されるので、多分いろんなことに使えます。知らんけど。

3-4. Homebrew 入れよう

後は<https://tech-blog.cloud-config.jp/2019-07-19-homebrew-on-wsl/> に書いてあることをやるだけです。

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

実際にインストールするコマンドはこれです。Homebrewの公式ページに行かないとそのときの正しいコマンドは得られないらしいです。

https://brew.sh/index_ja.html

あとは、以下の3コマンドを分けて入れればHomebrewが入るはずです

```
test -d ~/.linuxbrew && eval "$( ~/.linuxbrew/bin/brew shellenv)

test -d /home/linuxbrew/.linuxbrew && eval
$(/home/linuxbrew/.linuxbrew/bin/brew shellenv)

echo "eval '$(brew --prefix)/bin/brew shellenv'" >> ~/.profile
```

SLを走らせるパッケージを入れてみます。

```
brew install sl
```

```
sl
```

SLが右から走ってきたら成功です。

Homebrewが更新されるように、毎回使う前に

```
brew doctor
brew update
```

をするといいらしいです。

3-5. wget, rmtrash を入れる

homebrew を使ってソフトを入れるには、brew install を使います

```
brew install wget
```

しばらくして完了したら、次を打ってそれっぽいurlが入れば wget のインストールに成功しています。

```
which wget
```

rmtrash は最近のhomebrewから残念ながら削除されてしまったそうです。なので代替品を使ってみようと思います。trash-cli が Windows (Linux) でも使えて、rmtrashと同じような機能を持つパッケージになります。

```
brew install trash-cli
```

さて、問題はWindows上でLinuxがどこに置かれているかです。現在(2021/02/19)では、主に
C:\Users\(\各自のユーザー名)
\AppData\Local\Packages\CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\LocalState\rootfs
にLinuxのホームがあります。

今やりたいのは、削除したら一生戻ってこれない rm の代わりに、ゴミ箱へ移す trash-cli を “rm” というコマンドで実行できるようにすることです。そこで、こういうコマンドの実決める .bashrc をいじることにします。

...\LocalState\rootfs\home\ (Ubuntuの最初に決めたユーザー名) \.bashrc を Win のテキストエディタで開き、その最後の行の後に

```
alias rm='trash-put'  
alias rmdir='trash-put'
```

と入れて保存しておきます。これでUbuntuをいったん閉じて又開くと、rm/rmdirを打つと trash-put を打ったのと同様になります。

なお rootfs\home\○○\ は Ubuntu 上での ~ というパスになります。

さて、例えば /mnt/c/kihondir/a.txt を trash-put でゴミ箱に入れたとして、実際にはどこに入ったのでしょうか？その答えは、C:\.Trash-1000 です。なおこれは、/mnt/c/... での結果で、もし rootfs\... での操作の場合は ~/.local/share/Trash/ となります。つまり、trash-put は実際には消したつもりでほかのフォルダに移しただけであり、実際にあとで消すのは自分でできます (ゴミ箱を空にするのと同じ)。

4. Ubuntuで R Python Github⁵⁷⁵ (Level 1 準備編 の内容)

Ubuntu での操作に慣れるために、R, Python (Bioconda) を入れてみます。現在のディレクトリは、前に作ったゼミ用のディレクトリにしてください (cd で)

4-1. R を Homebrew で入れてみます。

```
brew install r
```

めちゃくちゃ簡単！ちなみにこんな感じでいろんなソフト入れてみたいなってなったら、Homebrewのホームページ(<https://formulae.brew.sh/formula-linux/>)から検索することで brew install が使えるか分かります。

ちなみに、Rというソフトは他のソフトの機能を用いて実現されています（大抵のソフトがそう）、これは依存関係と呼ばれるらしく、Rをインストールするとき同時に依存関係で必要なソフトもインストールしています。逆に、既に同じインストールされた依存関係はもう一度インストールせずに済むため、いろんなソフトを入れるうちにどんどんとインストールにかかる時間は減っていくと言えるでしょう。Rの場合だと、例えば GCC (C言語のコンパイル用ソフト) が入るのが見えると思います。

インストールが完了すれば、Rが起動できます

```
R
```

すると、何やらいろいろと出てきた後に

```
>
```

が出てきます。これはこの行でRのソースコードを入力することが出来ることになります。例えば次のコードを入力してみます（> は省略しています）。

```
d <- runif(10)
plot(d)
q()
```

すると、今開いているディレクトリ（dry）に“Rplots.pdf”が追加されます。これを開くと、1~10のx軸に0.0~1.0のyの値を持つ点がプロットされた図が表示されると思います。Rが正常に機能しているとわかります。

Rの拡張機能を利用するには、ソースコード中にてそれを読み込むようにします。たとえば CRAN という拡張は

```
install.packages("ggplot2")
```

Bioconductor は

```
source("http://bioconductor.org/biocLite.R")
biocLite("meshr")
```

のようにして使えます。また R のソースコードは Windows環境の vscode などからいじって、作業環境に置いておいて Ubuntu で実行、といったことが出来ます。詳しくはやりません。DRY解析のRの演習は各自で頑張ってください。

4-2. 次に、Python の環境である Miniconda と Bioconda を入れてみます。ここからは <http://www.aise.ics.saitama-u.ac.jp/~gotoh/MinicondaOnUbuntu1804.html> に書いてあることをやります。作業ディレクトリは各自で適当に作りましょう。そしたらまずは Miniconda で必要なファイルを wget で仕入れてきます。

```
wget
https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

一行です。これを次で起動します。

```
bash Miniconda3-latest-Linux-x86_64.sh
```

途中色々あるのは全部 yes で大丈夫です。インストール先は ~/miniconda3/ になるようです（変えられるようですが、変えた時の動作の保証はしかねます）。

一旦 Ubuntu を exit して再起動します。作業ディレクトリまで移動して、.bashrc に miniconda のことを登録します。3行です。

```
~/miniconda3/bin/conda init bash
source ~/.bashrc
which conda
```

それっぽいアドレスが出たら成功です。

```
conda config --add channels defaults
conda config --add channels conda-forge
conda config --add channels bioconda
```

これで設定に bioconda を加えました。

あとは本に書かれている必要そうなツールを conda を用いて導入します。

```
conda install -c bioconda sra-tools
conda install -c bioconda trim-galore
conda install -c bioconda kallisto
```

途中で、[y/n] が聞かれたら y を返して進めます。これで Bioconda 環境が整ったようです。

4-3. Github は重要ですが、本質的にはあまり関係ないので省略しちゃいます。アカウントの登録とかはしておいたらいと思います。

<http://www.aise.ics.saitama-u.ac.jp/~gotoh/IntroGitOnLinux.html>

ここに大量に情報があります。