

平成28年度NGSハンズオン講習会
補足資料
「パスの理解を深める」

2016年7月28日

amelieff

「パスを通す」って何？

「パスを通す」って何？

- **パス**：ファイルやディレクトリの位置
- **絶対パス**：ディレクトリ階層構造の頂点から目的のファイルやディレクトリの位置までの道筋を省略せず記述する方式。コマンドを実行する位置によらず、「/」から始まり、あるファイル・ディレクトリについては必ず同じパスになる
- **相対パス**：コマンドを実行する位置から見た目的のファイルのディレクトリの位置を記述する方式
- **コマンド検索パス**：**シェル**（ユーザーが打ち込んだコマンドを解釈して実行するプログラム）が**実行ファイル**を探しに行くパス（コマンド置き場）

「パスを通す」って何？

■ コマンドラインは

行頭の **コマンド** と **サブコマンド、オプション、ファイル名など** から成る

【例】

```
$ ls -l
```

```
$ samtools view -b handson.sam > handson.bam
```

「パスを通す」って何？

- シェルは **行頭** に書かれた文字列を **コマンド** として解釈し、**コマンド検索パス**（**コマンド置き場**）に探しに行き、コマンド置き場にあったプログラムを実行する
- コマンド検索パスの一覧

```
$ echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/cd-hit:/usr/lib/cd-hit:/home/iu/Downloads/FastQC
```

「パスを通す」って何？

パスが通っている

= 「コマンド検索パスになっている」
(コマンド置き場)

パスを通す

= 「コマンド検索パスに追加する」
(コマンド置き場)

「パスを通す」って何？

- 備考：コマンド検索パスには優先順位がある

```
$ echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/cd-hit:/usr/lib/cd-hit:/home/iu/Downloads/FastQC
```



それぞれのディレクトリに、同じ名前のプログラムがあった場合、優先度が高い場所にあったプログラムが呼び出されて実行される。

「パスを通す」って何？

- 備考：使っているコマンドがどこに置いてあるか確認できる

```
$ which samtools
```

```
/usr/local/bin/samtools
```

```
$ which R
```

```
/usr/bin/R
```

コマンド置き場には、システムにとって重要なコマンドが置かれていることもある。

誤操作で重要なコマンドを削除すると、最悪OSの再インストールなどが必要になるため、一般ユーザは操作権限を与えられていない。

仮に与えられたとしても、**非常に慎重に操作しなくてはならない**。

プログラムのインストール

プログラムのインストール

■ インストール：

**パスが通っているディレクトリに、
プログラム（の本体、またはリンク）を置いて、
コマンドとしてプログラムを呼び出せるようにする**

プログラムのインストール

■ パスが通っていない場所にあるプログラム

```
$ /usr/local/src/samtools-1.3/samtools index input.bam
```

- 明示的にプログラムの場所を指定して実行する
(相対パスでも絶対パスでもよい)

■ パスが通っている場所 (/usr/local/bin) にあるプログラム

```
$ samtools index.bam
```

- パスを指定せず、プログラム名だけで実行できる

JARファイルは？

JARファイルは？

- java コマンドを使ってJARファイルを実行している。

```
$ java -jar trimmomatic-0.36.jar
```

JARファイルをコマンド置き場に置いて、コマンドとして**呼び出す**ことはできますが、javaコマンドを使っていないので**実行**はできません。

→ パスを指定してJARファイルを呼び出さなくてはいけないので、パスが通ったディレクトリにJARファイルを置く意味はあまりない

JARファイルは？

JARファイルを使いやすくする方法① 呼び出しやすい場所に置く

■ 例

```
$ mkdir /home/iu/jar  
  
$ ln -s /usr/local/src/Trimmomatic-0.36/trimmomatic-0.36.jar ¥  
  /home/iu/jar  
  
$ java -jar ~/jar/trimmomatic-0.36.jar
```

~ : ユーザーのホームディレクトリを意味する省略記号
(ここでは /home/iu)

JARファイルは？

JARファイルを使いやすくする方法② **置いてある場所を呼び出しやすくする**

■ 例

```
$ mkdir /home/iu/jar  
  
$ ln -s /usr/local/src/Trimmomatic-0.36/trimmomatic-0.36.jar ¥  
  /home/iu/jar  
  
$ export JARDIR=/home/iu/jar  
  
$ java -jar $JARDIR/trimmomatic-0.36.jar
```

「\$JARDIR」と書くだけで「/home/iu/jar/」を意味するよう
に設定しました。

※一時的な設定です。永続的な設定にするには別の設定が必要です

JARファイルは？

JARファイルを使いやすくする方法③ JARファイル起動プログラムを作る

■ 例

```
$ cat trimmomatic.sh
```

```
#!/bin/sh
```

```
FILE1=$1
```

```
FILE2=$2
```

```
OUT=$3
```

```
java -jar /usr/local/src/Trimmomatic-0.36/trimmomatic-0.36.jar ¥
```

```
PE -threads 2 -phred33 -trimlog log.txt ¥
```

```
$FILE1 $FILE2 ¥
```

```
$OUT¥_paired_1.fastq $OUT¥_unpaired_1.fastq ¥
```

```
$OUT¥_paired_2.fastq $OUT¥_unpaired_2.fastq ¥
```

```
SLIDINGWINDOW:5:20 LEADING:20 TRAILING:20 MINLEN:80
```

何を書いてあるか詳しく知りたい人は
シェルスクリプトを勉強しましょう！

JARファイルは？

JARファイルを使いやすくする方法③ **JARファイル起動プログラムを作る**

■ 例

```
$ sh trimmomatic.sh input_R1.fastq input_R2.fastq output
```

→ Trimmomaticが実行されます

JARファイルを起動するプログラムを、コマンド置き場に置きます

```
$ chmod +x trimmomatic.sh  
$ ln -s /path/to/trimmomatic.sh /usr/local/bin  
$ trimmomatic.sh
```

→ Trimmomaticが実行されます