ゲノム支援情報解析講習会
（国立遺伝学研究所；2015年11月19～20日）

Linux基本コマンド

かずさDNA研究所
平川英樹

参考資料

新Linux/UNIX入門 第3版
林 晴比古（著）

出版社: ソフトバンククリエイティブ
ISBN-10: 4797369841
ISBN-13: 978-4797369847

内容

- Linuxの特徴
- カーネルとシェル
- ディレクトリ
- 講習会用のデータの準備
- ユーザ、ファイルのアクセス制御
- パーミッションの変更
- ファイル操作（１～２）
- テキスト処理（１～５）
- ディレクトリ操作
- その他
- 実践（１）
- エディタ
- 実践（２）

Linuxの特徴

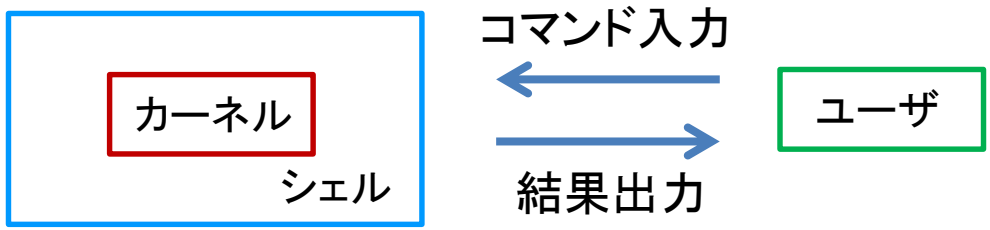
- マルチユーザ
  一台のコンピュータを複数のユーザで同時に使用できる。
- スーパーユーザと一般ユーザ
  スーパーユーザ：管理者。システムメンテナンスやユーザ管理ができる。root権限。
  一般ユーザ：アカウントを取る必要がある。
- ドライブ名が無い（C:ドライブなど）
- マウントが必要
  外部記憶装置（ハードディスクなど）をマウントしてファイルシステムに組み込む必要がある。
- 小文字と大文字は区別される
  コマンドlsとLSは別物
- ドットファイル
  隠しファイル。通常は表示されない。ls –aコマンドで表示される。
- コマンドの入力
  一般に「コマンド名　[オプション]　[引数]」で指定する。
  ls、ls mydir、ls –a mydir、ls –al mydir、cat –n file1
- ディレクトリ
  「/」で区切る
- コマンド実行の中断：Ctrl + C

など

カーネルとシェル

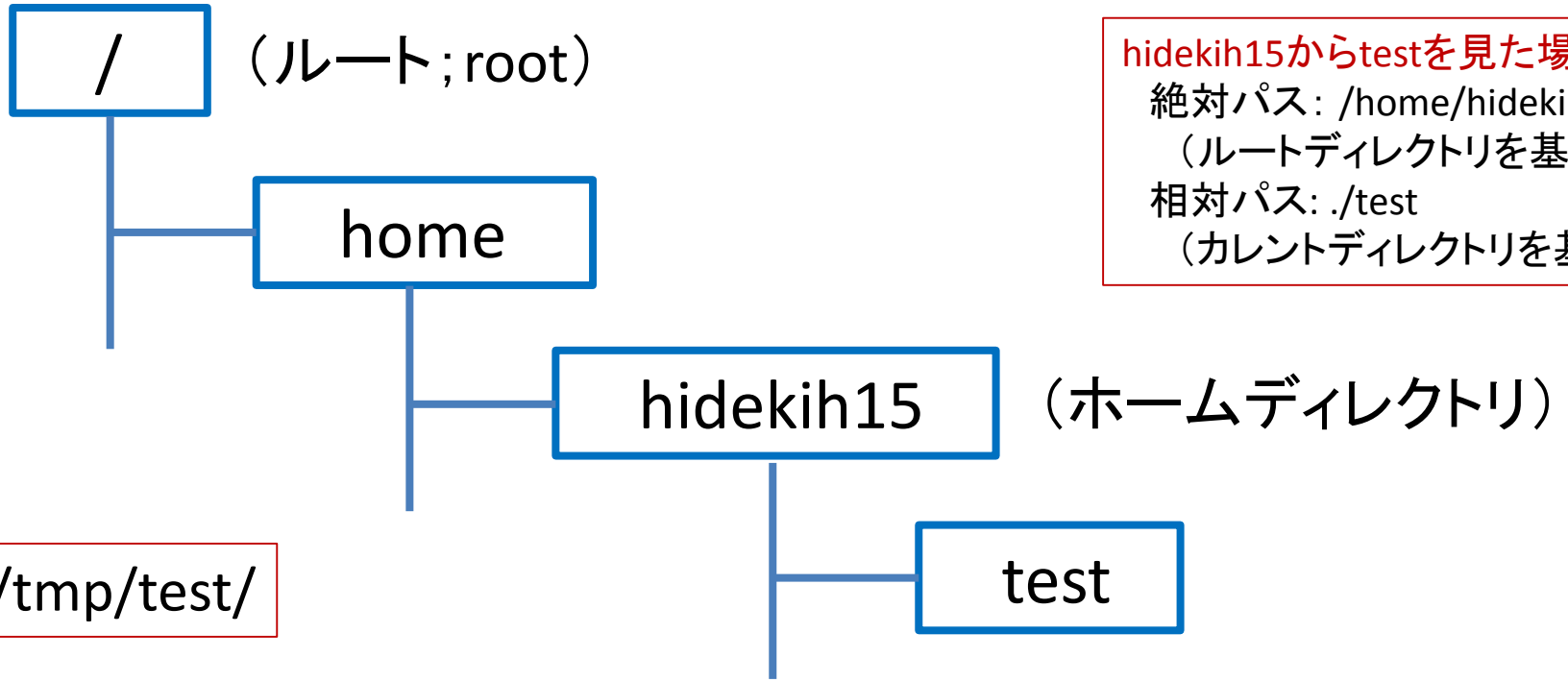
カーネル（kernel；核）
　メモリ、周辺装置、ファイルシステムを管理。
　指定されたコマンドを実行。
　Linuxというときには、カーネル部分のソフトウェアを指す。

シェル（shell；貝殻。カーネルを包む）
　ユーザとカーネルとの橋渡し役。
　ユーザと対話し、その入力を解釈し、指示されたコマンドをカーネルに渡して実行。
　（ユーザの入力したコマンドを通訳してカーネルに渡す）



幾つかのシェル（bash（Linuxの標準シェル）、csh、tcsh、zsh、shなど）

ディレクトリ



/tmp/test/

hidekih15からtestを見た場合
絶対パス： /home/hidekih15/test/
（ルートディレクトリを基点）
相対パス: ./test
（カレントディレクトリを基点）

ディレクトリ記号

/	ルートディレクトリ
.	カレントディレクトリ
..	親ディレクトリ（ひとつ上のディレクトリ）
~	ホームディレクトリ（チルダ）
~-	直前にいたディレクトリ（チルダとマイナス）
~hidekih15	指定ユーザのホームディレクトリ

```
cd /home/hidekih15/test/ #移動
pwd # カレントディレクトリを見る
cd ../
pwd

man # コマンドのマニュアル表示
man pwd
```

ログイン後、プロンプト（$）が表示されていたらコマンドを入力できる。
$ コマンド　オプション　引数　（例：　ls -l（コマンド：ls）、オプション（l））

講習会用のデータ準備

まずは以下のコマンドを実行して講習会用のデータを準備してください。

```
cd  # ホームディレクトリに移動
pwd  #  自身のホームディレクトリにいることを確認
mkdir test  # testというディレクトリ名を作成
cp /home/hidekih15/test/* ./test  #すべてのファイルをコピー
cp /tmp/test/* ./test
cd
mkdir sample
cp /tmp/sample/* ./sample
cd test  # testというディレクトリに移動
ls
```

/home/ユーザ名/testの下にtest1.faなどがあればファイルがコピーされています。

test1.fa

```
>test1
MVTALSDVNNTDNYGAGQIQVLEGLEAVRKRPGMYIGSTSERGLHHLVWEIVDNSIDEALAGYANQIEVV
IEKDNWIKVTDNGRGIPVDIQEKMGRPAVEVILTVLHAGGKFGGGGYKVSGGLHGVGSSVVNALSQDLEV
YVHRNETIYHQAYKKGVPQFDLKEVGTTDKTGTVIRFKADGEIFTETTVYNYETLQQRIRELAFLNKGIQ
ITLRDERDEENVREDSYHYEGGIKSYVELLNENKEPIHDEPIYIHQSKDDIEVEIAIQYNSGYATNLLTY
ANNIHTYEGGTHEDGFKRALTRVLNSYGLSSKIMKEEKDRLSGEDTREGMTAIISIKHGDPQFEGQTKTK
LGNSEVRQVVDKLFSEHFERFLYENPQVARTVVEKGIMAARARVAAKKAREVTRRKSALDVASLPGKLAD
CSSKSPEECEIFLVEGDSAGGSTKSGRDSRTQAILPLRGKILNVEKARLDRILNNNEIRQMITAFGTGIG
GDFDLAKARYHKIVIMTDADVDGAHIRTLLLTFFYRFMRPLIEAGYVYIAQPPLYKLTQGKQKYYVYNDR
ELDKLKSELNPTPKWSIARYKGLGEMNADQLWETTMNPEHRALLQVKLEDAIEADQTFEMLMGDVVENRR
```

ユーザ、ファイルのアクセス制御

＊ユーザ

hidekih15

グループの元で管理（hh-kdri；ls -alコマンドで確認）

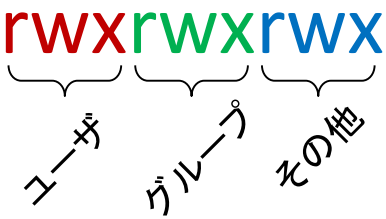
＊パーミッション（アクセス権）

1）ファイルに対するアクセス許可

単位：　ユーザ（自分自身（u））、グループ（自分自身が属する（g））、その他（o）

パーミッション（許可）：　「読み出し（r）」、「書き込み（w）」、「実行（x）」の可否

可能（r, w, xの文字）、不可（-）



rw-r----- : ユーザ（read/write可）
グループ（read可）
その他（不可）

```
ls -l /home/hidekih15/test/test1.fa # パーミッションの確認
```

2）ディレクトリに対するアクセス許可

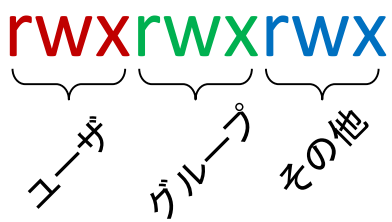
drwxrwxrwx

drwxrwxr-x : ユーザ（read/write可）
グループ（read/write可）
その他（read/実行可）

先頭文字
- :　通常ファイル
d :　ディレクトリ
l : シンボリックリンク

パーミッションの変更

パーミッションの変更：chmod



ユーザ：
　u（ユーザ）、g（グループ）、o（その他）、a（全ユーザ）
オペレータ：
　+（モード追加）、-（モード削除）
モード：
　r（読み込み）、w（書き込み）、x（実行）

```
chmod o+w test1.fa # その他ユーザの書き込み許可を与える（他の設定はそのまま）
chmod g-w test1.fa # グループユーザの書き込み許可を不可にする
chmod a+w test1.fa # 全ユーザの書き込み許可を与える
```

オーナーの変更：chown

```
chown hh test1.fa # hidekih15からhhに変更（hhは例です）
chown -R hh ./test # test以下の全てのファイルをhhに変更
```

ファイル操作（1/2）

＊ファイル内容の表示：cat [オプション] [ファイル名]
cat -n test1.fa # test1.faに行数を表示

＊ファイルをコピー：cp
cp test1.fa test2.fa # test1.faをtest2.faにコピー
cp test1.fa test_dir1 # test1.faをtest_dir1にコピー
（test_dir1が事前に必要。mkdir test_dir1で作成）
cp -r test_dir1 test_dir2 # ディレクトリごとコピー

＊ファイルの移動・ファイル名の変更：mv
mv test1.fa test_dir1 # test1.faをtest_dir1に移動
mv test1.fa test11.fa # test1.faの名前をtest11.faに変更

＊ファイルの削除：rm [オプション] [ファイル名]
rm -i test11.fa # 削除確認メッセージの表示
rm -r test_dir1 # ディレクトリを丸ごと削除（注意）
rm –ri test_dir1 # ディレクトリを丸ごと削除（確認しながら）

＊ファイルの簡易作成：1) echo, 2) cat, 3) touch
1) echo abcde > test_echo.txt
2) cat > test_cat.txt
aaa
bbb
Ctrl+D
3) touch test_touch.txt

エイリアス定義
rmコマンドはシェルの設定（.bashrc）でrm –iになっていることが多い。mvも同様。

ドットファイル（隠しファイル、「.」が先頭）
.emacs： emacsの隠しファイル
設定ファイルが多い

リダイレクト（入出力の切り替え）
>： コマンド実行結果をファイルに出力
>>： 上書き
<: コンピュータへの入力

パイプ機能（コマンドの連続実行）
|： コマンドを繋げる（キーボードの右上）
ls –al | wc -l # ls –alの実行結果の行数を数える（wcコマンド：ファイルの単語数、バイト数、行数を数える）

ワイルドカードを用いたファイル指定
*： 任意の文字列
?： 任意の一文字
[...]： 括弧内に含まれる任意の一文字
[!...]： 括弧内に含まれない任意の一文字

ファイル操作（2/2）

＊ファイルの圧縮・解凍：
compress / uncompress、gzip / gunzip、bzip / bunzip / bzcat
compressとgzipで圧縮されたファイルはlessかzcatで見ることが出来る。
gzip test1.fa # 実行後test1.fa.gzを作成
gunzip test1.fa.gz # 実行後test.faを作成

＊リンクを張る： ln
ln -s test1.fa test_ln.fa # test_ln.faからtest1.faを見ることができる
ln -s test_ln test1 # test1ディレクトリからtest_lnディレクトリを見ることができる
利点：ディレクトリのリンクもできる、別のファイルシステム上のファイルともリンクできる。

＊ディスク容量を調べる： du [オプション] [ディレクトリ名（ファイル名）]
du # キロバイト単位で表示
du -s # 総計サイズのみ表示
du -sh # 合計サイズとファイルサイズを単位（M、K）付きの短い数字で表示

テキスト処理（1/5）

＊ファイルの先頭部分の表示：head [オプション] ファイル名

head test1.fa # ファイルの先頭10行を表示（デフォルトでは10行）

head -20 test1.fa # ファイルの先頭20行を表示

＊ファイルの終端部分を表示：tail [オプション] ファイル名

tail test1.fa # ファイルの最後10行を表示

tail -10 test1.fa # ファイルの最後の10行を表示

tail +5 test1.fa # ファイルの5行目からを表示

tail +6 test1.fa | head -4 # 6行目から4行分（6～9行目）を表示

cat -n test1.fa | head -4 # 行番号付きで表示

＊ファイル内容の並び替え：sort [オプション] ファイル名

sort test2.txt # 先頭文字から並べる

sort -r test2.txt # 逆順に並べる

sort -n test2.txt # 数値で順に並べる（デフォルトは文字列）

sort -nr test2.txt # 数値の逆順で並べる

sort -k 5 test2.txt # 5番目以降のカラムで並べる

sort -k 2,5 test2.txt # 2番目から5番目のカラムで並べる

sort -k 2,2 test2.txt # 2番目のカラムのみで並べる

test2.txt（右側を省略）

```
合計 20
drwxrwxrwx  3 hidekih15 hh-kdri 4096  2月 27
drwxrwxrwx 12 hidekih15 hh-kdri 4096 11月 26
drwxrwxrwx  4 hidekih15 hh-kdri 4096 11月 23
-rwxr-xr-x  1 hidekih15 hh-kdri  661  2月 27
-rwxr-xr-x  1 hidekih15 hh-kdri  663  2月 27
-rw-r--r--  1 hidekih15 hh-kdri    0  2月 27
```

＊ファイル内の重複行を削除する：uniq [オプション] 入力ファイル名 主力ファイル名

uniq test_uniq1.txt # 重複行を削除

uniq test_uniq2.txt #重複行が続いていない場合はsortコマンドを実施

sort test_uniq2.txt | uniq

uniq -u test_uniq1.txt # 重複していない行のみ表示

uniq -d test_uniq1.txt # 重複している行のみ表示

uniq -c test_uniq1.txt # 重複している行数を表示

test_uniq1.txt

```
abcde
abcde
ABCDE
vwxyz
VWXYZ
```

test_uniq2.txt

```
abcde
ABCDE
abcde
vwxyz
VWXYZ
```

テキスト処理（2/5）

＊文字を置換： tr [オプション] 文字列1 文字列2

```
tr abcd ABCD < test_tr.txt   # abcdをABCDに置換
tr a-z A-Z < test_tr.txt   #小文字を大文字に変換
tr 0-9 . < test_tr.txt  # 数値をピリオドに変換
tr -c 0-9 . < test_tr.txt   # 数値以外はピリオドに変換
tr -d " " < test_tr.txt   # 空白を削除
tr -d "¥n" < test_tr.txt   # 改行文字を削除
tr -s " " < test_tr.txt   # 連続する空白を削除（1つの空白になる）
```

test_tr.txt

```
abcde
abcde
ABCDE
01234
012AB
5    6
vwxyz
VWXYZ
ab1ef
ab5ef
```

＊テキストをページ単位で表示： more [オプション] ファイル名

```
more test_R1.fq
more -2 test_R1.fq   # 一度にスクロールする行数を指定（ここでは2行ずつ）
more +2 test_R1.fq   # 指定行から表示開始（ここでは2行目から）
ls  -al | more   # ls -alの出力結果をmoreで見る
```

・内部コマンド

- Enter: 次の1行を表示
- Space: 次の一画面を表示
- b: 前の一画面を表示
- h: ヘルプ表示
- v: viエディタの起動（:qでmoreに戻る）
- =: 現在の行番号を表示（Backspageで消える）
- /文字列：文字列の検索
- q: moreを終了

test_R1.fq(右側を省略)

```
@MRSA_SRR583008.1 HISEQ:73:B00NUACXX:3:1101:
ATTTTTTAAAATAAGCTGCTTTATTAGTTATTAATCTTGCATTG
+MRSA_SRR583008.1 HISEQ:73:B00NUACXX:3:1101:
@CCFFFFFHHHHGGHGJJIFHGIJDIIHIJIJIJJGIIIGGIJJ
@MRSA_SRR583008.2 HISEQ:73:B00NUACXX:3:1101:
GTATTTCCTAGGACATATAGAGGAACATTCAACGTAAAATTCCC
+MRSA_SRR583008.2 HISEQ:73:B00NUACXX:3:1101:
CBBFFFFFHHHHFHIJJJJJJJIJJJJJJJJJJJIJJIJJJJJJ
@MRSA_SRR583008.3 HISEQ:73:B00NUACXX:3:1101:
ATTACGGATAAACTATATTCTGACTCATCTACCGTTATAATGCC
+MRSA_SRR583008.3 HISEQ:73:B00NUACXX:3:1101:
##########################################
```

テキスト処理（3/5）

＊テキストをページ単位で表示：less　[オプション]　ファイル名

less test_R1.fq

less -2 test_R1.fq # 一度にスクロールする行数を指定（ここでは2行ずつ）

less +2 test_R1.fq # 指定行から表示開始（ここでは2行目から）

ls -alh | less

・内部コマンド

↓: 次の一行を表示

↑: 前の一行を表示

b: 次の一画面を表示

/文字列：文字列を検索（見つかった行を先頭行にする）

?文字列：文字列を先頭方向に検索

v: エディタを起動（:qで戻る）

n: 同じ方向に検索を繰り返す

N: 逆さ方向に検索を繰り返す

q: lessを終了

less ファイル名1 ファイル名2　# 複数ファイルの表示

:pで前のファイルを表示

:nで次のファイルを表示

test_R1.fq （右側を省略）

```
@MRSA_SRR583008.1 HISEQ:73:B00NUACXX:3:1101:
ATTTTTTAAAATAAGCTGCTTTATTAGTTATTAATCTTGCATTG
+MRSA_SRR583008.1 HISEQ:73:B00NUACXX:3:1101:
@CCFFFFFHHHHGGHGJJIFHGIJDIIHIJIJIJJGIIIGGIJJ
@MRSA_SRR583008.2 HISEQ:73:B00NUACXX:3:1101:
GTATTTCCTAGGACATATAGAGGAACATTCAACGTAAAATTCCC
+MRSA_SRR583008.2 HISEQ:73:B00NUACXX:3:1101:
CBBFFFFFHHHHFHIJJJJJJJIJJJJJJJJJJJIJJIJJJJJJ
@MRSA_SRR583008.3 HISEQ:73:B00NUACXX:3:1101:
ATTACGGATAAACTATATTCTGACTCATCTACCGTTATAATGCC
+MRSA_SRR583008.3 HISEQ:73:B00NUACXX:3:1101:
##########################################
```

テキスト処理(4/5)

＊文字列を検索：grep [オプション] 文字列 ファイル名

```
grep BLAST test2.txt   # BLASTがある行を表示
grep -v BLAST test2.txt   # BLASTが無い行を表示
grep -i blast test2.txt   # 大文字と小文字を区別しない
grep -c BLAST test2.txt   # 一致する行数を表示
grep  BLAST test1.fa test2.txt   # 複数ファイルの時はファイル名も表示
grep -w BL test2.txt   # 単語として検索(BLには一致するがBLASTには一致しない)
```

・正規表現を使った検索

```
grep ^AB test_tr.txt   # 先頭にAB
grep “ab[123]ef” test_tr.txt   # abと123の中の一文字とefで構成される文字列
grep “ab.ef” test_tr.txt   # abと任意の一文字とefで構成される文字列
```

＊より多くの正規表現(+や?など)を使用できる：egrep

＊正規表現解釈をせずに文字列そのままの検索：fgrep

test_tr.txt

```
abcde
abcde
ABCDE
01234
012AB
5    6
vwxyz
VWXYZ
ab1ef
ab5ef
```

test2.txt

```
合計 20
drwxrwxrwx  3 hidekih15 hh-kdri 4096  2月 27 15:33 2015 .
drwxrwxrwx 12 hidekih15 hh-kdri 4096 11月 26 17:22 2013 ..
drwxrwxrwx  4 hidekih15 hh-kdri 4096 11月 23 20:51 2013 BLAST
-rwxr-xr-x  1 hidekih15 hh-kdri  661  2月 27 09:04 2015 test1.fa
-rwxr-xr-x  1 hidekih15 hh-kdri  663  2月 27 09:03 2015 test1.fa~
-rw-r--r--  1 hidekih15 hh-kdri    0  2月 27 15:33 2015 test2.txt
```

正規表現

^：行の先頭

$：行の末尾

.：任意の一文字

*：直前の文字の0回以上の繰り返し

¥：正規表現文字のエスケープ

[…]：[]の任意の一文字

[abc]：a, b, cのいずれか一文字

[^abc]：a, b, c以外のいずれか一文字

[a-n]：a～nのいずれか一文字

テキスト処理（5/5）

＊ファイル内容の相違を調べる：diff　[オプション]　ファイル名　ファイル名2

diff -i test_diff1.txt test_diff2.txt　# 大文字と小文字の違いを無視

diff -s test_diff1.txt test_diff1.txt　# ファイルの内容が同じだった場合にも報告

・結果

3c3　# file1の3行目とfile2の3行目で相違。Changeで正しくなる。

< ccccc　# file1の3行目

> YYYYY # file2の3行目

6d5 # file1の6行目とfile2の5行目で相違。Deleteで正しくなる

< 00000 # file1の6行目

7a7,9　# file1の7行目とfile2の7～9行目で相違。Addで正しくなる。

> ggggg # file2の7行目

> hhhhh # file2の8行目

> iiiii # file2の9行目

test_diff1.txt

```
aaaaa
bbbbb
ccccc
ddddd
eeeee
00000
fffff
```

test_diff2.txt

```
aaaaa
bbbbb
YYYYY
ddddd
eeeee
fffff
ggggg
hhhhh
iiiii
```

＊テキストをエコー表示する：echo

echo This is sample　# This is sampleと表示

echo This is sample > test_echo.txt # test_echo.txtにThis is sampleを出力

・コマンドの実行結果をテキストにする

echo List: `ls`　# lsコマンドの出力結果を表示（バッククオーツ（`））（List: test1.fa test2.txt test_R1.fq test_R2.fq test_diff1.txt test_diff2.txt test_tr.txt test_uniq1.txt test_uniq2.txt）

ディレクトリ操作

＊ディレクトリ内容を表示する：ls [オプション] [パス名]
-a:ドットファイルも表示
-l: ロング形式で表示
-R: サブディレクトリ内容も表示
-1: 一行に一つずつ表示
-h: 合計サイズとファイルサイズを単位（M、K）付きの短い数字で表示

＊ディレクトリの作成：mkdir [オプション] ディレクトリ名
mkdir mydir1
mkdir mydir1 mydir2
mkdir -p mydir1/mysubdir1 # 途中のディレクトリを同時に作成

＊ディレクトリの削除：rmdir [オプション] ディレクトリ名
rmdir mydir1
rmdir -p mydir1/mysubdir1 #途中のディレクトリを同時に削除
rmdir -r mydir1 #ディレクトリを丸ごと削除

＊ディレクトリの移動と名前変更：mv ディレクトリ名1 ディレクトリ名2
mv mydir1 mydir2 # mydir2が既存の場合は移動、新規の場合は名前変更

＊カレントディレクトリの表示：pwd

＊ディレクトリの移動：cd ディレクトリ名
cd # ディレクトリを指定しない場合、ホームディレクトリに移動（cd ~と同じ）

その他

＊コマンドのヒストリ（履歴）を表示：history
history # 過去のコマンドを表示
history 10 # 過去のコマンドを新しいものから10個だけ表示
history -c # コマンド履歴をクリア
＊コマンドのヒストリの番号を指定して実行： !n
historyの結果の例
1162 pwd
1163 cd /home/hidekih15/test
1164 cd /home/hidekih15/test/
1165 ls
1166 history
!1162と入力するとpwdが実行される。
＊コマンドに別名（エイリアス）を付ける：alias
alias hs=history # hsと入力するとhistoryが実行される
alias # エイリアス定義一覧を見る（以下は一部）
alias cp='cp -i'
alias emacs='emacs -nw'
alias hs='history'
alias mv='mv -i'
alias rm='rm -i'
alias vi='vim‘
＊エイリアス定義を削除： unalias
unalias hs #エイリアスhsの削除

＊ディスクドライブ容量の表示
df –k # キロバイト表示
df –kh # キロ以上
（メガ、ギガ、ペタ）

実践（1）

① ファイルのコピー

1．ホームディレクトリにディレクトリraw_dataを作成
2．ディレクトリ（/home/hidekih15/test/（/tmp/test/））にあるtest_R1.fqとtest_R2.fqを1）で作成した ディレクトリ（raw_data）にコピー
3．test_R1.fqとtest_R2.fqのファイル名をユーザ名_R1.fqとユーザ名_R2.fqに変更
（ユーザ名がhidekih15の場合、hidekih15_R1.fqとhidekih15_R2.fq）
4．ユーザ名_R1.fqとユーザ名_R2.fqのファイル容量を調べる
5．ユーザ名_R1.fqとユーザ名_R2.fqをそれぞれgzip形式で圧縮し容量を調べる

② ファイルのマージ

1．ディレクトリ（/home/hidekih15/sample/lecture/data/BLAST/query（/tmp/sample））にあるtest_aa.faとtest_nt.faを合わせてtest_aa_nt.faを作り、ディレクトリraw_dataに保存
2．作成されたtest_aa_nt.faの内容を見る
3．test_aa_nt.faの行数を数える
4．test_aa_nt.faの配列名を抽出しtest_aa_nt.listに保存
配列名は>test_aaと>test_ntとする。

③ 文字列抽出

1．test_R1.fqの配列名（先頭が@MRSAで始まる行）を抽出してtest_R1.fq.listを作る
2．そのリード数を数える

エディタ

Emacs（Linuxでは標準のエディタ）、vi、nanoなどのエディタがある。

＊起動

Emacs　# emacsを起動

Emacs test1.fa　# test1.faを編集

＊ファイル操作

Ctrl+x Ctrl+s （ファイルを保存）

Ctrl+x Ctrl+w （ファイルに新しい名前で保存）

Ctrl+x Ctrl+f （ファイルを開く）

Ctrl+x Ctrl+c （Emacsの終了）

＊カーソル移動

Ctrl+p （1行上へ）

Ctrl+n （1行下へ）

Ctrl+b （1文字左へ）

Ctrl+f （1文字右へ）

META+< （ファイル先頭へ）

META+> （ファイル末尾へ）

META+v （全ページへ）

Ctrl+v （全ページへ）

Ctrl+a （行頭へ）

Ctrl+e （行末へ）

META+ goto line （指定行番号の行へ）

＊文字削除

Ctrl+d （カーソル位置の1文字を削除）

Ctrl+k （カーソル位置から行末まで削除）

＊テキストのカットアンドペースト

Ctrl+Space （マーク（リージョンの先頭）の指定）

カーソル位置 （ポイント（リージョンの末尾）の指定）

Ctrl+w （リージョンのカット）

META+w （リージョンのコピー）

Ctrl+y （保存内容のペースト）

＊検索

Ctrl+s （終端方向へのサーチ）

Ctrl+r （先頭方向へのサーチ）

Esc （検索終了）

＊置換

META+x replace string （確認なしの全置換）

META+% （確認つきの置換）

＊処理の中断

Ctrl+g （困った場合に実行すると戻れる）

emacs ファイル名

METAはEscキーのこと
Perlプログラミングやスクリプトの作成などで重要

実践（2）

＊perlプログラムの作成

1．test.plという名前のファイルをemacsで開く
2．以下のperlプログラムをtest.plに書き込む

```
#!/usr/bin/perl
print “Perl programming¥n”;
```

3．内容を保存する
4．emacsを閉じる
3．chmod +x ./test.pl　#perlへの実行権の付与
4．./test.pl　#perlの実行

実践（1）解答例

① ファイルのコピー

1．ホームディレクトリにディレクトリraw_dataを作成

mkdir raw_data

2．ディレクトリ（/home/hidekih15/test/（/tmp/test/））にあるtest_R1.fqとtest_R2.fqを1）で作成した ディレクトリ（raw_data）にコピー

cp /home/hidekih15/test/test_R1.fq ./raw_data

cp /home/hidekih15/test/test_R2.fq ./raw_data

（ cp /home/hidekih15/test/test_*.fq ./raw_dataでも可（ワイルドカード使用））

3．test_R1.fqとtest_R2.fqのファイル名をユーザ名_R1.fqとユーザ名_R2.fqに変更

（ユーザ名がhidekih15の場合、hidekih15_R1.fqとhidekih15_R2.fq）

mv test_R1.fq hidekih15_R1.fq

mv test_R2.fq hidekih15_R2.fq

4．ユーザ名_R1.fqとユーザ名_R2.fqのファイル容量を調べる

ls -alh hidekih15_R1.fq hidekih15_R2.fq #それぞれ33K

5．ユーザ名_R1.fqとユーザ名_R2.fqをそれぞれgzip形式で圧縮し容量を調べる

gzip hidekih15_R1.fq

gzip hidekih15_R2.fq

ls –alh hidekih15_R1.fq.gz #8.9K

ls –alh hidekih15_R2.fq.gz #9.7K

② ファイルのマージ

1．ディレクトリ（/home/hidekih15/sample/lecture/data/BLAST/query）にあるtest_aa.faとtest_nt.faを合わせてtest_aa_nt.faを作り、ディレクトリraw_dataに保存

cat /home/hidekih15/sample/lecture/data/BLAST/query/test_aa.fa /home/hidekih15/sample/lecture/data/BLAST/query/test_nt.fa > ./raw_data/test_aa_nt.fa

2．作成されたtest_aa_nt.faの内容を見る

less test_aa_nt.fa　#cat, moreなどでも可

3．test_aa_nt.faの行数を数える

wc -l test_aa_nt.fa　#23行

4．test_aa_nt.faの配列名を抽出しtest_aa_nt.listに保存

配列名は>test_aaと>test_ntとする。

grep '^>' test_a_nt.fa > test_aa_nt.list

③ 文字列抽出

1．test_R1.fqの配列名（先頭が@MRSAで始まる行）を抽出してtest_R1.fq.listを作る

grep '^@MRSA' test_R1.fq > test_R1.fq.list

2．そのリード数を数える

wc –l test_R1.fq.list　# 100行