The 1st. D-STEP & The 36th DDBJing /

第1回 D-STEP 講習会 & 第36回 DDBJing講習会

2018 (H30) Jan. 26th. 10:00 - 12:00

Takeshi Kawashima / 川島武士 / かわしまたけし

本日の方針

参加者の皆さんには

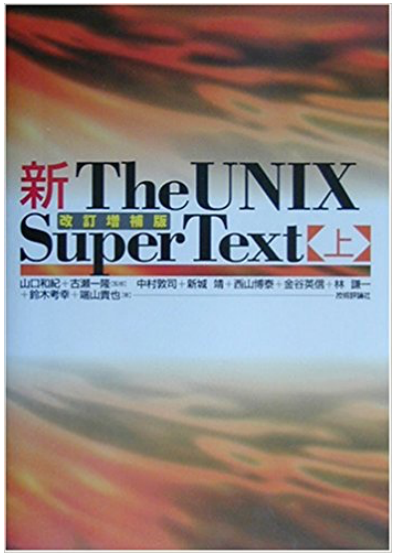
1. NIG (国立遺伝学研究所)の大型計算機にアクセスし、
2. Unix[系]のコマンドを利用し
3. 塩基配列データを対象とした配列解析を行う

　　ための基礎を身につけてもらいます。

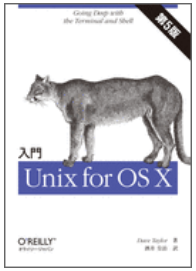
ほとんどの内容は、NIGの大型計算機に特化した内容ではなく、他の多くの大型計算機センターでの利用時に通用するものだと思います。実際に使うジョブ管理システム(キューイングシステム)は、NIGの大型計算機で採用しているUGE (Univa Grid Engine)を前提に説明しますが、他の大型計算機センターにおいても概ね同じようなシステムが動いていますので、参考になるはずです。

おすすめテキスト

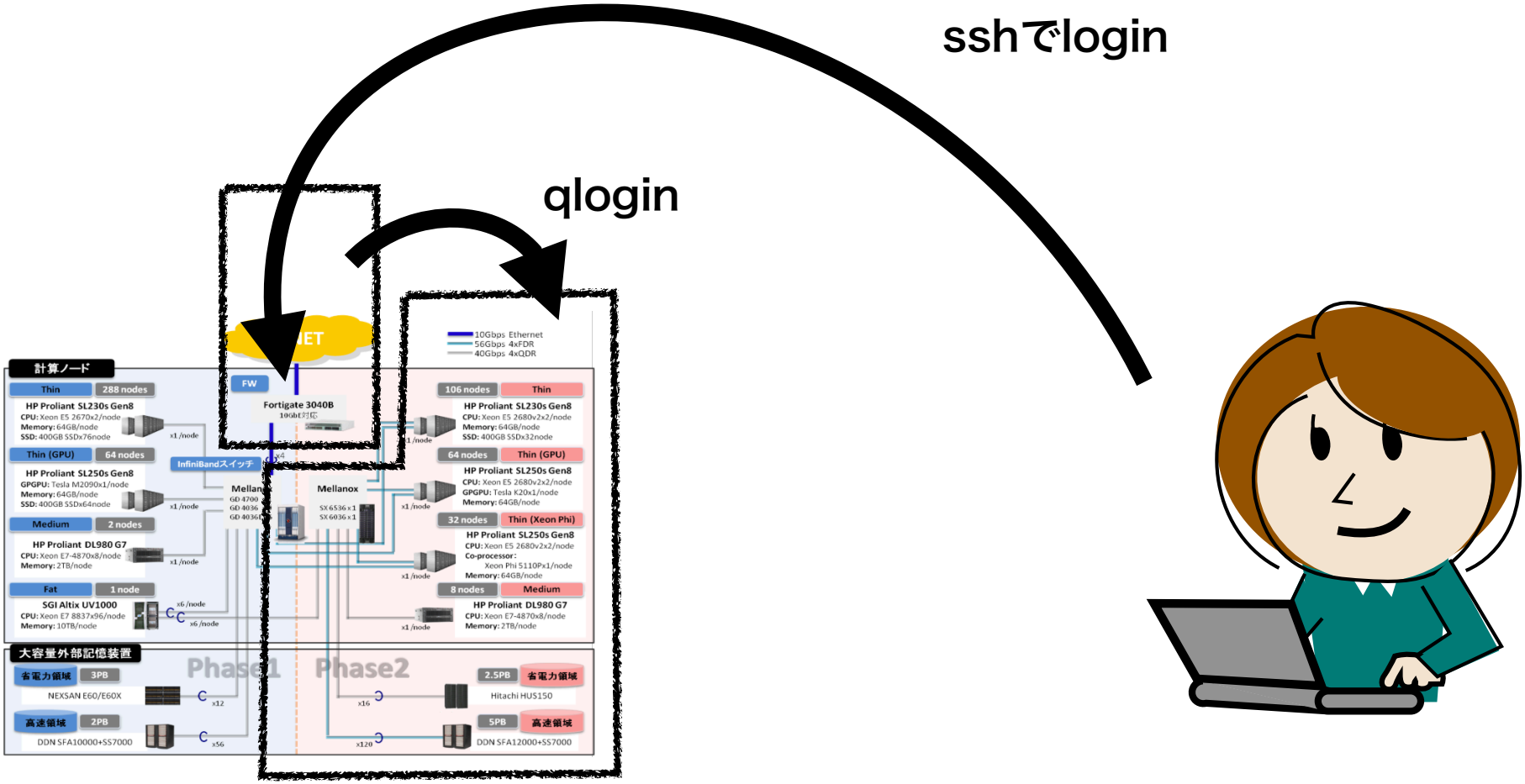
入門Unixオペレーティングシステム
絶版かも。。。



新The UNIX Super Text 上下巻
絶版かも。。。



入門Unix for OS X
上記とほぼ同じ内容です。
（ただしMacOSXユーザ用）



NIGスパコンへのloginは、2ステップ。

まずssh、次にすぐにqlogin!!

本日の目標、7項目

関連キーワード

1. **自分が利用している環境について知る** — **NIG Super Computer**
   - ls, lfs, pwd, /usr/local/seq, etc.
2. **キューイングシステムを利用する** — **ジョブ管理システム**
   - qlogin, qsub, qstat, qdel, qreport, etc.
3. **欲しいデータを取得する** — **生物データベース**
   - DDBJ, DRA / ftp, wget, tar, gzip, bzip2, etc.
4. **自分の環境を使いやすくする** — **UNIX**
   - .bashrc, ローカルへのインストール, 環境変数 etc.
5. **マッピング作業** — **Genome Informatics**
   - bowtie2, etc.
6. **キューイングシステム利用のための簡単なBASHプログラミング** — **ジョブ管理システム**
   - bash, アレイジョブ（SGE_TASK_ID）
7. **（時間に余裕があれば）描画のための簡単なRプログラミング** — **Visualize**
   - R

1. 自分が利用している環境について知る

大型計算機へのログインの方法と、
Unixの基本的なコマンドを、おさらいしましょう。

1-1. 自分が利用している環境について知る (1/8)

この辺りに書いてあります

© 2014 DBCLS, Licensed under Creative Commons Attribution 4.0 International license (CC-BY 4.0)

1-2-1. スパコンへのログインの方法 (2/8)

自分のパソコンのターミナルにて

```
ls                                          #  エルエス
ls -l
# ssh <your-login-id>@gw2.ddbj.nig.ac.jp
```

```
DSTEPMac:TOSHIBA dstep$ ssh dstep@gw2.ddbj.nig.ac.jp
Last login: Thu Jan 11 13:08:03 2018 from xxx.xx.xx.xx
-----------------------------------------------------------------------
Thank you for using supercomputer system.
This node is in use for login service only. Please use 'qlogin'.
-----------------------------------------------------------------------
[dstep@gw2 ~]$ qlogin
```

自分のパソコンのターミナルに戻る

```
[dstep@gw2 ~]$ exit
[dstep@gw2 ~]$ exit
```

id_rsa.pubの設定がややこしくなってきた時、id_rsaのファイルをあえて指定する

```
# ssh -i <your-login-id>@gw2.ddbj.nig.ac.jp
DSTEPMac:TOSHIBA dstep$ ssh -i ~/.ssh/id_rsa dstep@gw2.ddbj.nig.ac.jp
Enter passphrase for key '/Users/dstep/.ssh/id_rsa.nig':
Last login: Thu Jan 11 13:08:03 2018 from xxx.xx.xx.xx
-----------------------------------------------------------------------
Thank you for using supercomputer system.
This node is in use for login service only. Please use 'qlogin'.
-----------------------------------------------------------------------
[dstep@gw2 ~]$ qlogin
```

© 2014 DBCLS, Licensed under Creative Commons Attribution 4.0 International license (CC-BY 4.0)

1-2-2. スパコンへのログイン (3/8)

自分のパソコンのターミナルに戻る

```
[dstep@gw2 ~]$ exit
[dstep@gw2 ~]$ exit
```

X window システムを自分のターミナルに飛ばす

```
# ssh -X <your-login-id>@gw2.ddbj.nig.ac.jp
# ssh -Y <your-login-id>@gw2.ddbj.nig.ac.jp
```

```
DSTEPMac:TOSHIBA dstep$ ssh -Y dstep@gw2.ddbj.nig.ac.jp
Last login: Thu Jan 11 13:08:03 2018 from xxx.xx.xx.xx
-----------------------------------------------------------------------
Thank you for using supercomputer system.
This node is in use for login service only. Please use 'qlogin'.
-----------------------------------------------------------------------
```

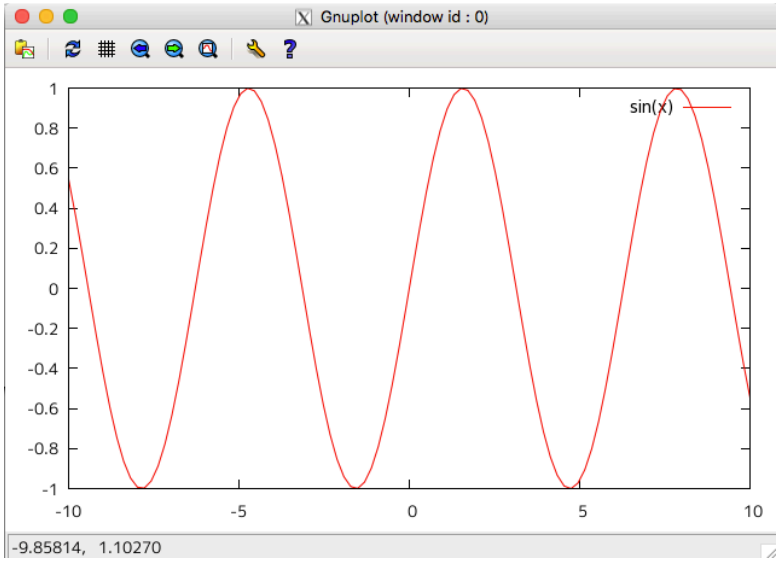
ここからはgnuplot

```
[dstep@nt098 ~]$ gnuplot

   G N U P L O T
   Version 4.4 patchlevel 4

Terminal type set to 'wxt'
gnuplot> plot sin(x)

gnuplot> quit
```



© 2014 DBCLS, Licensed under Creative Commons Attribution 4.0 International license (CC-BY 4.0)

1-2-3. スパコンへのログイン (4/8)

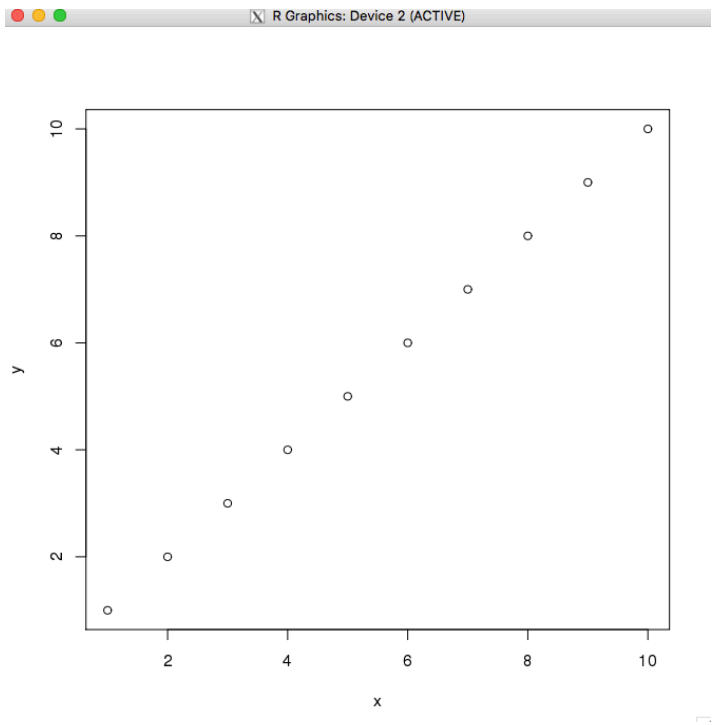
ここからはR

```
[dstep@nt094 ~]$ R

R version 2.14.1 (2011-12-22)

> x <- 1:10
> y <- 1:10
> plot(x, y)

# Rからでる
> q()
Save workspace image? [y/n/c]: n
```



© 2014 DBCLS, Licensed under Creative Commons Attribution 4.0 International license (CC-BY 4.0)

1-3. 利用可能なディスクスペース (5/8)

qloginしたターミナルにて

```
ls
ls -l
ls -l /home
ls -l /home | grep dstep
```

```
[dstep@nt096 ~]$ ls -l /home |grep <your-login-id>
lrwxrwxrwx 1 dstep        ma-nig          22  4月  8 10:43 2016 dstep -> /
lustre3/home/dstep
[dstep@nt096 ~]$
```

この場合、自分のホームディレクトリが /lustre3 にあることがわかる。

なおphase2の利用ユーザーは、lustre1からlustre5までのどれかに振り分けられている。

```
[dstep@nt096 ~]$ lfs quota -u <your-login-id>  /lustre3
Disk quotas for user dstep (uid 3907):
     Filesystem  kbytes   quota   limit   grace   files   quota   limit   grace
       /lustre3 200000000       0 1000000000       - 7000000       0       0       -
```

この場合、自分には /lustre3の元で1000000000 kbyte (=1Tbyte)割り当てられており

そのうち200000000 kbyte (=200Gbyte)利用していることが分かる。

© 2014 DBCLS, Licensed under Creative Commons Attribution 4.0 International license (CC-BY 4.0)

1-4. 環境変数 (6/8)

```
# qloginしたターミナルにて
echo “TEST”

printenv
printenv | wc
printenv | sed -e “s/=.*//"

echo $HOME
echo $SHELL
echo $PWD
echo $USER
echo $PATH

time
which time
```

```
[dstep@nt098 ~]$ which time
/usr/bin/time
```

time コマンドは　/usr/binの下においてある

```
echo $PATH
```

/usr/binというディレクトリは確かに環境変数PATHの中に記載されている

```
[dstep@nt098 ~]$ echo $PATH
/home/geadmin2/UGER/bin/lx-amd64:/usr/lib64/qt-3.3/bin:/opt/pgi/linux86-64/current/bin:/
usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/pkg/java/current/bin:/opt/intel/xe_2016/
compilers_and_libraries_2016.1.150/linux/bin/intel64:/opt/intel/xe_2016/
compilers_and_libraries_2016.1.150/linux/mpi/intel64/bin:/opt/intel/xe_2016/debugger_2016/
gdb/intel64_mic/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/
ibutils/bin:/opt/intel/itac/8.1.3.037/bin:/home/dstep/local/bin:/home/dstep/bin
```

© 2014 DBCLS, Licensed under Creative Commons Attribution 4.0 International license (CC-BY 4.0)

1-5. 利用可能OSS (7/8)

	0.5.9	/usr/local/pkg/bwa/0.5.9	○	○	○	○	2012/02/07 11:07
canu	1.6	/usr/local/pkg/canu/v1.6	○	○	○	○	2017/10/19 14:22
	1.5	/usr/local/pkg/canu/v1.5	○	○	○	○	2017/05/18 15:42
		/usr/local/pkg/canu/current	○	○	○	○	2017/05/18 15:42
	0.69-6	/usr/local/pkg/circos/0.69-6	○	○	○	○	2017/10/19 14:36

/usr/local/pkg/canu/current

```
[dstep@nt094 ~]$ /usr/local/pkg/canu/current/bin/canu --version
```

```
Canu snapshot v1.5 +43 changes
(r8243 f05c81531c19887459ca53ac3702509ee70eac22)
```

full pathを打たないで使えるようにしたい

```
[dstep@nt094 ~]$ canu
```

```
-bash: canu: コマンドが見つかりません
```

環境変数 PATHを設定することで、利用可能になります

© 2014 DBCLS, Licensed under Creative Commons Attribution 4.0 International license (CC-BY 4.0)

1-6. 自分が利用している環境について知る (8/8)

ここまでで、

適切にログインし、
自分の環境を確認する

ことができるようになりました。

環境変数についても学びました。

© 2014 DBCLS, Licensed under Creative Commons Attribution 4.0 International license (CC-BY 4.0)

2. キューイングシステムを利用する

大型計算機利用時に特有のジョブ管理システム（キューイングシステム）
について学びます。

ただし**最低限**知っていなければならないことは、すでに1.で学んだように、
ログイン時にすぐqloginコマンドを打つことです。

とはいえ、ゲノム解析などを実際に行うには、**キューイングシステム**について、
正しい理解が必要です。ここでは実際にqsubコマンドを利用してみましょう。

2. キューイングシステムを利用する

大型計算機利用時に特有のジョブ管理システム（キューイングシステム）
について学びます。

ただし**最低限**知っていなければならないことは、すでに1.で学んだように、
ログイン時にすぐqloginコマンドを打つことです。

とはいえ、ゲノム解析などを実際に行うには、**キューイングシステム**について、
正しい理解が必要です。ここでは実際にqsubコマンドを利用してみましょう。

2. キューイングシステムを利用する　(1/6)

この後利用するいくつかのファイルをダウンロードしてください。

(gitを普段利用している方へ：

ここでは説明のため、あえてgitの機能を利用せずにwgetで取得しています。)

以下を、スパコンにログインした後行ってください

```
cd ~/
mkdir myprojects
mkdir myprojects/DSTEP20180126
cd ~/myprojects/DSTEP20180126
wget https://raw.githubusercontent.com/tkwsm/DSTEP20180126/master/time.qsub
wget https://raw.githubusercontent.com/tkwsm/DSTEP20180126/master/time.sh
wget https://raw.githubusercontent.com/tkwsm/DSTEP20180126/master/bowtie2.qsub
wget https://raw.githubusercontent.com/tkwsm/DSTEP20180126/master/bowtie2_multi.qsub
wget https://raw.githubusercontent.com/tkwsm/DSTEP20180126/master/sra2fastq_loop.sh
wget https://raw.githubusercontent.com/tkwsm/DSTEP20180126/master/trimmomatic.sh
wget https://raw.githubusercontent.com/tkwsm/DSTEP20180126/master/hemoglobin_b.fa
```

```
[dstep@nt096 tmp]$ wget https://raw.githubusercontent.com/tkwsm/DSTEP20180126/master/time.qsub

2018-01-11 16:55:38 (24.8 MB/s) - `time.qsub' へ保存完了 [236/236]
```

gitを普段利用されている方は下記の方が楽です。

```
DSTEPMac:tmp dstep$ git clone git@github.com:tkwsm/DSTEP20180126.git
Cloning into 'DSTEP20180126'...

DSTEPMac:tmp dstep$ ls
DSTEP20180126
DSTEPMac:tmp dstep$ cd DSTEP20180126/
DSTEPMac:DSTEP20180126 dstep$ ls
README.md time.qsub time.sh
```

2. キューイングシステムを利用する　(2/6)

time.sh というプログラムが取得できたかどうかをlsで確認する

```
[dstep@nt096 tmp]$ ls
time.sh
```

time.sh というプログラムの中身をcatで確認する

```
[dstep@nt096 tmp]$ cat time.sh
```

```
#!/bin/sh

echo "start program"
echo "start sleep 15 sec"
sleep 15
echo "end sleep"
echo "program done"
```

→ # shebang行

→ # プログラム本体

→ # time.sh
というプログラムの中身

time.sh というプログラムを実行してみる

```
[dstep@nt094 dstep20180126]$ bash time.sh
start program
start sleep 15 sec
end sleep
program done
[dstep@nt094 dstep20180126]$
```

2. キューイングシステムを利用する　(3/6)

time.qsub というプログラムが取得できたかどうかをlsで確認する

```
[dstep@nt096 tmp]$ ls
time.qsub
```

time.qsub というプログラムの中身をcatで確認する

```
[dstep@nt096 tmp]$ cat time.qsub
```

```
#!/bin/sh
#$ -S /bin/sh
#$ -cwd
#$ -l s_vmem=1G,mem_req=1G
#$ -l short
### #$ -l debug
#$ -pe def_slot 1
#$ -o ./
#$ -e ./

source ~/.bashrc

echo "start program"
echo "start sleep 15 sec"
sleep 15
echo "end sleep"
echo "program done”
```

→ # shebang行

→ # qsubコマンド設定

→ # 普段利用している
設定ファイルの読み込み

→ # プログラム本体

→ # time.qsub
というプログラムの中身

```
[dstep@nt096 tmp]$
```

2. キューイングシステムを利用する　(4/6)

time.qsub というプログラムを実行してみる

```
[dstep@nt094 dstep20180126]$ qsub time.qsub
Your job 10384431 ("time.qsub") has been submitted

[dstep@nt094 dstep20180126]$ qstat

job-ID     prior   name       user         state submit/start at     queue                          jclass           slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------------------------
  10384220 0.25036 QLOGIN     dstep        r     01/16/2018 11:36:44 login.q@nt094i                                                1
  10384431 0.00000 time.qsub  dstep        qw    01/16/2018 12:35:25                                                               1

[dstep@nt094 dstep20180126]$ qstat
job-ID     prior   name       user         state submit/start at     queue                          jclass           slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------------------------
  10384220 0.25040 QLOGIN     dstep        r     01/16/2018 11:36:44 login.q@nt094i                                                1

[dstep@nt094 dstep20180126]$

[dstep@nt094 dstep20180126]$ ls

time.sh
time.qsub

time.qsub.e10384431
time.qsub.pe10384431
time.qsub.o10384431
time.qsub.po10384431
```

→ # 標準出力と標準エラー出力

2. キューイングシステムを利用する　(5/6)

time.qsub というプログラムの中身をcatで確認する

```
[dstep@nt096 tmp]$ cat time.qsub
```

```
#!/bin/sh
#$ -S /bin/sh
#$ -cwd
#$ -l s_vmem=1G,mem_req=1G
#$ -l short
### #$ -l debug
#$ -pe def_slot 1
#$ -o ./
#$ -e ./

source ~/.bashrc

echo "start program"
echo "start sleep 15 sec"
sleep 15
echo "end sleep"
echo "program done”
```

→ # shebang行

→ # qsubコマンド設定

→ # time.qsub
というプログラムの中身

利用可能バイオツール
利用可能OSS
利用可能DB

システム使用方法

基本的利用方法
その他UGE利用方法
ファイル転送方法
システム利用TIPS
稼働スケジュール
アドバンスドリザベーション　コマンド利用方法
アドバンスドリザベーション

計算ジョブの投入

計算ノードに計算ジョブを投入するには、qsubコマンドを利用します。qsubコマンドでジョブを投入する為には、投入するジョブスクリプトを作成する 必要が有ります。簡単な例ですが以下のように記述します。

```
#!/bin/sh
#$ -S /bin/sh
pwd
hostname
date
sleep 20
date
echo  “to stderr”  1>&2
```

この時、行の先頭に、"#$"が指定されている行が、UGEへのオプション指示行になります。オプション指示行をシェルスクリプトにする、またはqsub コマンド 実行時のオプションとして指定することでUGEに動作を指示します。主なオプションとしては以下のものがあります。

指示行の記述	コマンドラインオプション指示	指示の意味
#$ -S インタプリタパス	-S インタプリタパス	コマンドインタプリタをパス指定する。シェル以外のスクリプト言語も指定可能。このオプションは指定必要
#$ -cwd	#$ -cwd	ジョブ実行のカレントワーキングディレクトリを指定する。これによりジョブの標準出力、エラー出力もcwd上に出力される。 指定しない場合はホームディレクトリがカレントワーキングディレクトリとなりジョブ実行される。
#$ -N ジョブ名	-N ジョブ名	ジョブの名前を指定する。この指定が無ければスクリプト名がジョブ名になる。
#$ -o ファイルパス	-o ファイルパス	ジョブの標準出力の出力先を指定する。
#$ -e ファイルパス	-e ファイルパス	ジョブの標準エラー出力の出力先を指定する。

qsubには上記の他にも指定可能なオプションが多数ありますので、詳細についてはシステムにログイン後"man qsub"として、qsubコマンドの オンラインマニュアルを参照して確認してください。

NIG スパコンのシステム使用方法のあたりに詳しく書いてあります。

2. キューイングシステムを利用する　(6/6)

ここまでで、

適切にログインし、
UGEキューイングシステム(qsubコマンドなど)を使って、
計算機を利用できるようになりました。

3. 欲しいデータを取得する

大型計算機によるゲノム解析の特徴の一つは、事前に大量の生物学データを自分の計算機環境にアップロードする必要があることです。そのような生物学データは多くの場合、既存の配列データベースに登録されているものや利用者自身が取得したNGS等による配列データでしょう。NIGの大型計算機では、主要な生物学データベースがすでに相当数整備され、常時アップデートされています。これらのデータベースの利用方法を学びましょう。またNIGの大型計算機に整備されていないデータベースについてはご自身で最新のデータをダウンロードしてこなくてはなりません。そのために必要な知識としてftp, wgetなどのコマンドの利用方法を学びましょう。

3-1. NIGのスパコンにすでに整備してあるデータベースを利用する

3-2. その他のゲノムの基礎データの登録サイト

3-2. DRAから取得

3. 欲しいデータを取得する　(1/14)

NIGのスパコンにすでに整備してあるデータベースを利用する

スーパーコンピュータシステム　利用可能ＤＢ

スーパーコンピュータシステムでは、各計算ノード、各ログインノードから各種バイオ系DBが利用可能です。

1.DDBJ,NCBI,EBI等の公共DBを利用したい場合

スーパーコンピュータシステムにて利用可能なDBおよびパスは 利用可能DB一覧をご覧下さい。

2.DRAを含むその他のDDBJ DBを利用したい場合

上記利用可能DB以外のDDBJ DBについては下記方法にてデータをコピーしてご利用下さい。

利用可能DB一覧

DB名	パス (/usr/local/seq/)		設置されているファイルの詳細	更新頻度
DDBJ-unified-all	-	ddbj-unified-all/	-	毎日
	fasta/		ftp://ftp.ddbj.nig.ac.jp/ddbj_database/ddbjnew/unified-all/fasta/以下を解凍したFASTA形式ファイル	
	blast/		ftp://ftp.ddbj.nig.ac.jp/ddbj_database/ddbjnew/unified-all/blastdb/以下を解凍したBLASTデータベース	
DDBJ-unified-new	-	ddbj-unified-new/	-	毎日
	fasta/		ftp://ftp.ddbj.nig.ac.jp/ddbj_database/ddbjnew/unified-new/fasta/以下を解凍したFASTA形式ファイル	
	blast/		ftp://ftp.ddbj.nig.ac.jp/ddbj_database/ddbjnew/unified-new/blastdb/以下を解凍したBLASTデータベース	
GenBank	flat/	genbank/	ftp://ftp.ncbi.nih.gov/genbank/以下の*.seqファイル、およびWGSファイル	随時
	fasta/		上記のFASTA形式ファイル	
	blast/		上記FASTA形式ファイルのBLASTデータベース	
GenBank-UPD	flat/	genbank-upd/	ftp://ftp.ncbi.nih.gov/genbank/daily-nc/以下の全ファイル	毎日
	fasta/		上記ファイルの内*.flatのFASTA形式ファイル	
	blast/		上記FASTA形式ファイルのBLASTデータベース	
GenPept	-	genpept/	-	随時
	fasta/		GenBankのseqファイル内の翻訳可能なエントリを翻訳したFASTA形式ファイル	
	blast/		上記FASTA形式ファイルのBLASTデータベース	
GenPept-UPD	-	genpept-upd/	-	毎日
	fasta/		GenBank-UPDの*.flatファイル内の翻訳可能なエントリを翻訳したFASTA形式ファイル	
	blast/		上記FASTA形式ファイルのBLASTデータベース	

3. 欲しいデータを取得する (2/14)

NIGのスパコンにすでに整備してあるデータベースを利用する

```
[dstep@nt098 dstep20180126]$ ls /usr/local/seq/
blast  chemicaldb  entity  fasta  flat  igenome  old  taxonomy  work
[dstep@nt098 dstep20180126]$ ls /usr/local/seq/blast
ddbj-unified-all  embl       genbank       genpept       ncbi    refseq-upd
ddbj-unified-new  embl-upd   genbank-upd   genpept-upd   refseq  uniprot
```

```
[dstep@nt098 dstep20180126]$ ls
hemoglobin_b.fa
DRR016430.fastq  README.md  time.qsub  time.sh
```

```
[dstep@nt098 dstep20180126]$ cat hemoglobin_b.fa
>NP_000509.1 hemoglobin subunit beta [Homo sapiens]
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLG
AFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVAN
ALAHKYH
```

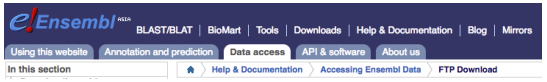
```
# blastp -query hemoglobin_b.fa
#        -db /usr/local/seq/blast/uniprot/swissprot
#        -num_alignments 1
#        -outfmt 6
```

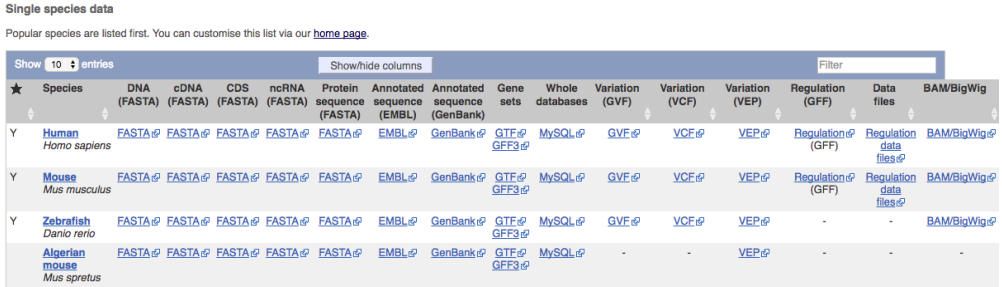
```
[dstep@nt098 dstep20180126]$ blastp -query hemoglobin_b.fa -db /usr/local/seq/blast/
uniprot/swissprot -num_alignments 1 -outfmt 6
```

```
NP_000509.1  sp|P68873|HBB_PANTR100.00014700114711473.25e-104301
```

3-3. その他のゲノムの基礎データの登録サイト (オススメいくつか)

https://asia.ensembl.org/index.html





★	Species	DNA (FASTA)	cDNA (FASTA)	CDS (FASTA)	ncRNA (FASTA)	Protein sequence (FASTA)	Annotated sequence (EMBL)	Annotated sequence (GenBank)	Gene sets	Whole databases	Variation (GVF)	Variation (VCF)	Variation (VEP)	Regulation (GFF)	Data files	BAM/BigWig
Y	Human *Homo sapiens*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF GFF3	MySQL	GVF	VCF	VEP	Regulation (GFF)	Regulation data files	BAM/BigWig
Y	Mouse *Mus musculus*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF GFF3	MySQL	GVF	VCF	VEP	Regulation (GFF)	Regulation data files	BAM/BigWig
Y	Zebrafish *Danio rerio*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF GFF3	MySQL	GVF	VCF	VEP	-	-	BAM/BigWig
	Algerian mouse *Mus spretus*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF GFF3	MySQL	-	-	VEP	-	-	-

https://phytozome.jgi.doe.gov/pz/portal.html

https://gold.jgi.doe.gov/

http://www.genome.jp/kegg/catalog/org_list.html

3. 欲しいデータを取得する (ftp経由) (4/14)

EnsemblからcDNAファイルを取得する方法を試しましょう

https://asia.ensembl.org/info/data/ftp/index.html

danio

Show 10 entries　Show/hide columns　danio

★	Species	DNA (FASTA)	cDNA (FASTA)	CDS (FASTA)	ncRNA (FASTA)	Protein sequence (FASTA)	Annotated sequence (EMBL)	Annotated sequence (GenBank)	Gene sets	Whole databases	Variation (GVF)	Variation (VCF)	Variation (VEP)	Regulation (GFF)	Data files	BAM/BigWig
Y	Zebrafish *Danio rerio*	FASTA	FASTA	FASTA	FASTA	FASTA	EMBL	GenBank	GTF GFF3	MySQL	GVF	VCF	VEP	-	-	BAM/BigWig

Showing 1 to 1 of 1 (filtered from 112 total entries)　<< < 1 > >>

/pub/release-91/fasta/danio_rerio/dna/ のインデックス

[親ディレクトリ]

名前	サイズ	更新日
CHECKSUMS	4.7 kB	2017/11/29 22:32:00
Danio_rerio.GRCz10.dna.chromosome.1.fa.gz	16.8 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.10.fa.gz	13.0 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.11.fa.gz	12.9 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.12.fa.gz	14.0 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.13.fa.gz	14.8 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.14.fa.gz	14.9 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.15.fa.gz	13.6 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.16.fa.gz	15.8 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.17.fa.gz	15.3 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.18.fa.gz	14.6 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.19.fa.gz	14.0 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.2.fa.gz	17.1 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.20.fa.gz	15.8 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.21.fa.gz	13.1 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.22.fa.gz	11.2 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.23.fa.gz	13.3 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.24.fa.gz	12.1 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.25.fa.gz	10.5 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.3.fa.gz	17.9 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.4.fa.gz	21.1 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.5.fa.gz	20.5 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.6.fa.gz	17.3 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.7.fa.gz	21.2 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromoso	15.5 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromoso	16.3 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromoso gz	5.4 kB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.nonchrom .gz	8.5 MB	2017/11/22 19:14:00

Click!!

今回は

Danio_rerio.GRCz10.dna.chromosome.9.fa.gz

だけをダウンロード

Danio rerio / ゼブラフィッシュ

3. 欲しいデータを取得する (ftp経由) (5/14)

EnsemblからanonymousFTPで、cDNAファイルを取得する方法を試しましょう

ftp://ftp.ensembl.org/pub/release-91/fasta/drosophila_melanogaster/cdna/ の一覧

上位のディレクトリーへ移動

名前	サイズ	最終更新日時
CHECKSUMS	1 KB	2017/11/29 13:33:00 JST
Drosophila_melanogaster.BDGP6.cdna.all.fa.gz	16973 KB	2017/11/20 23:37:00 JST
README	3 KB	2017/11/20 23:37:00 JST

ftpコマンド

```
[dstep@nt093 dat]$ ftp -i ftp.ensembl.org
Name (ftp.ensembl.org:dstep): anonymous
331 Please specify the password.
Password: <your-email-address>
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd pub/release-91/fasta/drosophila_melanogaster/cdna
ftp> ls
227 Entering Passive Mode (193,62,193,8,246,104)
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp            76 Nov 29 13:33 CHECKSUMS
-rwxrwxr-x    1 ftp      ftp      17379879 Nov 20 23:37 Drosophila_melanogaster.BDGP6.cdna.all.fa.gz
-rwxrwxr-x    1 ftp      ftp          2521 Nov 20 23:37 README
226 Directory send OK.
ftp> get Drosophila_melanogaster.BDGP6.cdna.all.fa.gz
local: Drosophila_melanogaster.BDGP6.cdna.all.fa.gz remote: Drosophila_melanogaster.BDGP6.cdna.all.fa.gz
227 Entering Passive Mode (193,62,193,8,94,215)
150 Opening BINARY mode data connection for Drosophila_melanogaster.BDGP6.cdna.all.fa.gz (17379879 bytes).
226 Transfer complete.
17379879 bytes received in 5.3 seconds (3.2e+03 Kbytes/s)
ftp> bye
221 Goodbye.
[dstep@nt093 dat]$
```

3. 欲しいデータを取得する (ftp経由) (5/14)

Ensemblからanonymous FTPで、cDNAファイルを取得する方法を試しましょう

/pub/release-91/fasta/danio_rerio/dna/ のインデックス

[親ディレクトリ]

名前	サイズ	更新日
CHECKSUMS	4.7 kB	2017/11/29 22:32:00
Danio_rerio.GRCz10.dna.chromosome.1.fa.gz	16.8 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.10.fa.gz	13.0 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.11.fa.gz	12.9 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.8.fa.gz	15.5 MB	2017/11/22 19:14:00
Danio_rerio.GRCz10.dna.chromosome.9.fa.gz	16.3 MB	2017/11/22 19:14:00

ftpコマンド

```
[dstep@nt093 dat]$ ftp -i ftp.ensembl.org
Name (ftp.ensembl.org:dstep): anonymous

331 Please specify the password.
Password: <your-email-address>
230 Login successful.
ftp> cd pub/release-91/fasta/danio_rerio/dna

ftp> ls
227 Entering Passive Mode (193,62,193,8,246,104)
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp            76 Nov 29 13:33 CHECKSUMS
-rwxrwxr-x    1 ftp      ftp      17379879 Nov 20 23:37 Danio_rerio.GRCz10.dna.chromosome.9.fa.gz
…..
-rwxrwxr-x    1 ftp      ftp          2521 Nov 20 23:37 README
226 Directory send OK.
ftp> get Drosophila_melanogaster.BDGP6.cdna.all.fa.gz
local: Drosophila_melanogaster.BDGP6.cdna.all.fa.gz remote: Drosophila_melanogaster.BDGP6.cdna.all.fa.gz
227 Entering Passive Mode (193,62,193,8,94,215)
150 Opening BINARY mode data connection for Drosophila_melanogaster.BDGP6.cdna.all.fa.gz (17379879 bytes).
226 Transfer complete.
17379879 bytes received in 5.3 seconds (3.2e+03 Kbytes/s)

ftp> bye
221 Goodbye.
[dstep@nt093 dat]$
```

3. 欲しいデータを取得する (ftp経由) (6/14)

```
ftp> bye
221 Goodbye.
[dstep@nt093 dat]$
```

```
[dstep@nt093 dat]$ ls
DRR016430.fastq  Danio_rerio.GRCz10.dna.chromosome.9.fa.gz
```

```
[dstep@nt093 dat]$ gunzip Danio_rerio.GRCz10.dna.chromosome.9.fa
[dstep@nt093 dat]$ ls
Danio_rerio.GRCz10.dna.chromosome.9.fa
```

今回は拡張子が *.gzだったので、gunzipで解凍しました。
拡張子が*bz2だった場合は、bunzip2で解凍しましょう