

PBSの利用方法

How to use PBS

22/Nov./2022 Hiroyo NISHIDE

講習用スクリプトのコピー

1. `/bio/course/` 下の `bias2211` ディレクトリをホームディレクトリにコピー
2. `bias2211/pbs_1ec` ディレクトリに移動
3. `ls` コマンドでファイルを確認

```
$ cp -r /bio/course/bias2211 ~/
```

```
$ cd ~/bias2211/pbs_1ec
```

```
$ ls
```

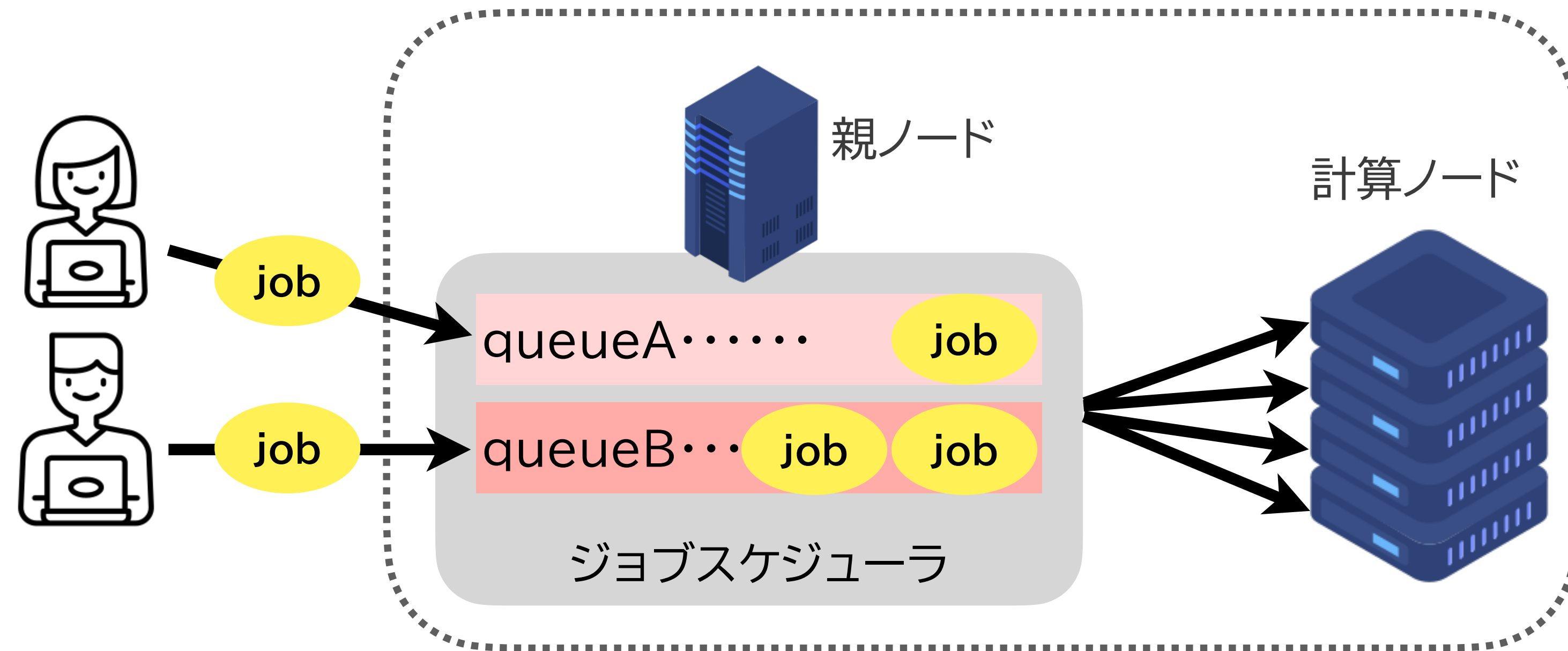
```
diamond.sh  sce_prot.fasta  spo.dmnd  test_qsub.sh
```

ジョブ管理システム: PBS

- 複数の人間が同じ計算機群を使う
- どのマシン/CPUが空いてるか？ どの計算を優先させるべきか？

ジョブ管理システム

- ▶ 計算機資源の割り当てを自動で行い効率を上げる
- ▶ ユーザはコマンドで親ノードにジョブを投げるだけ
- ▶ ジョブは「キュー」と呼ばれるリストに入って実行を待つ



ジョブ管理システム **PBS**

- PBSにおけるジョブ:
実行するコマンドやアプリケーションを記述したシェルスクリプト形式のタスク

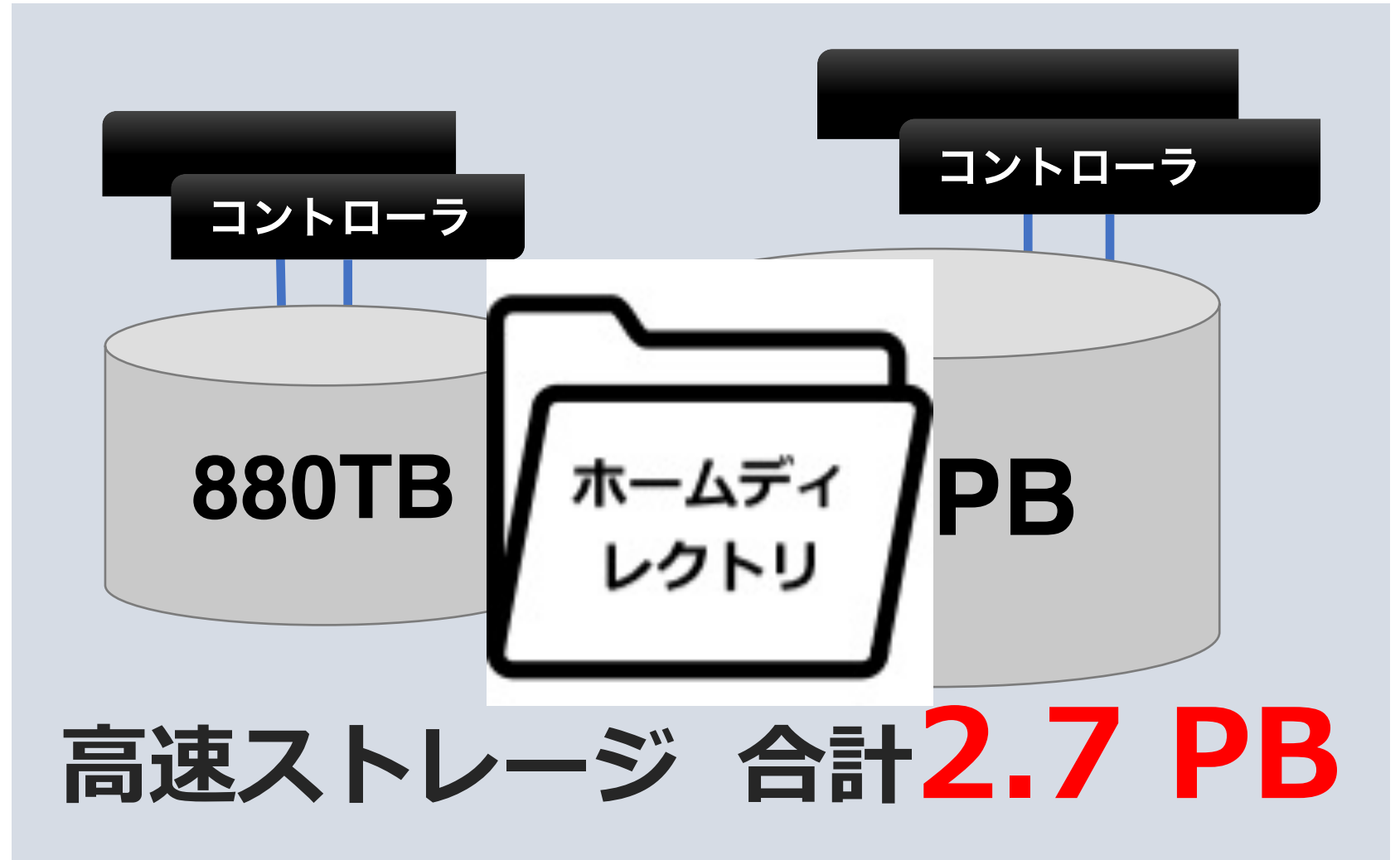
生物情報解析システム (bias5)

共有メモリ型計算サーバ
3TB Mem 72 cores
bias5-smp.nibb.ac.jp

共有メモリ型計算サーバ
4TB Mem 80 cores
ldas-smp.nibb.ac.jp
smpls, smpm, smpl

共有メモリ型計算サーバ
527GB Mem 64 cores
diaf-smp1.nibb.ac.jp

共有メモリ型計算サーバ
527GB Mem 64 cores
diaf-smp2.nibb.ac.jp



small, medium
large, blast

分散処理計算機クラスタ
800cores 4.8GB Mem/core
bias5-node[01-20].nibb.ac.jp



ログインノード
768GB Mem
bias5.nibb.ac.jp

生物情報解析システム (bias5)

共有メモリ型計算サーバ

3TB Mem 72 cores
bias5-smp.nibb.ac.jp

共有メモリ型計算サーバ

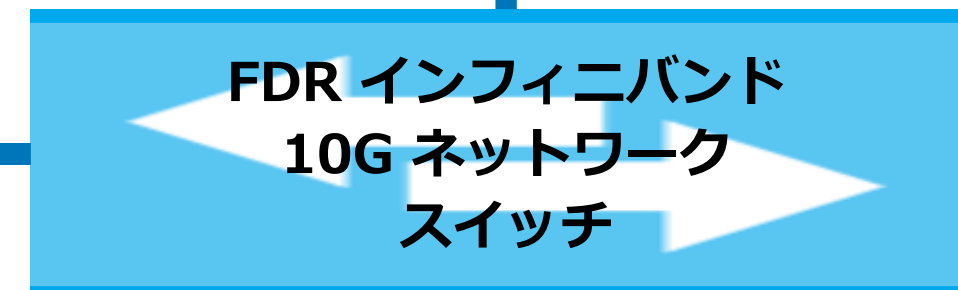
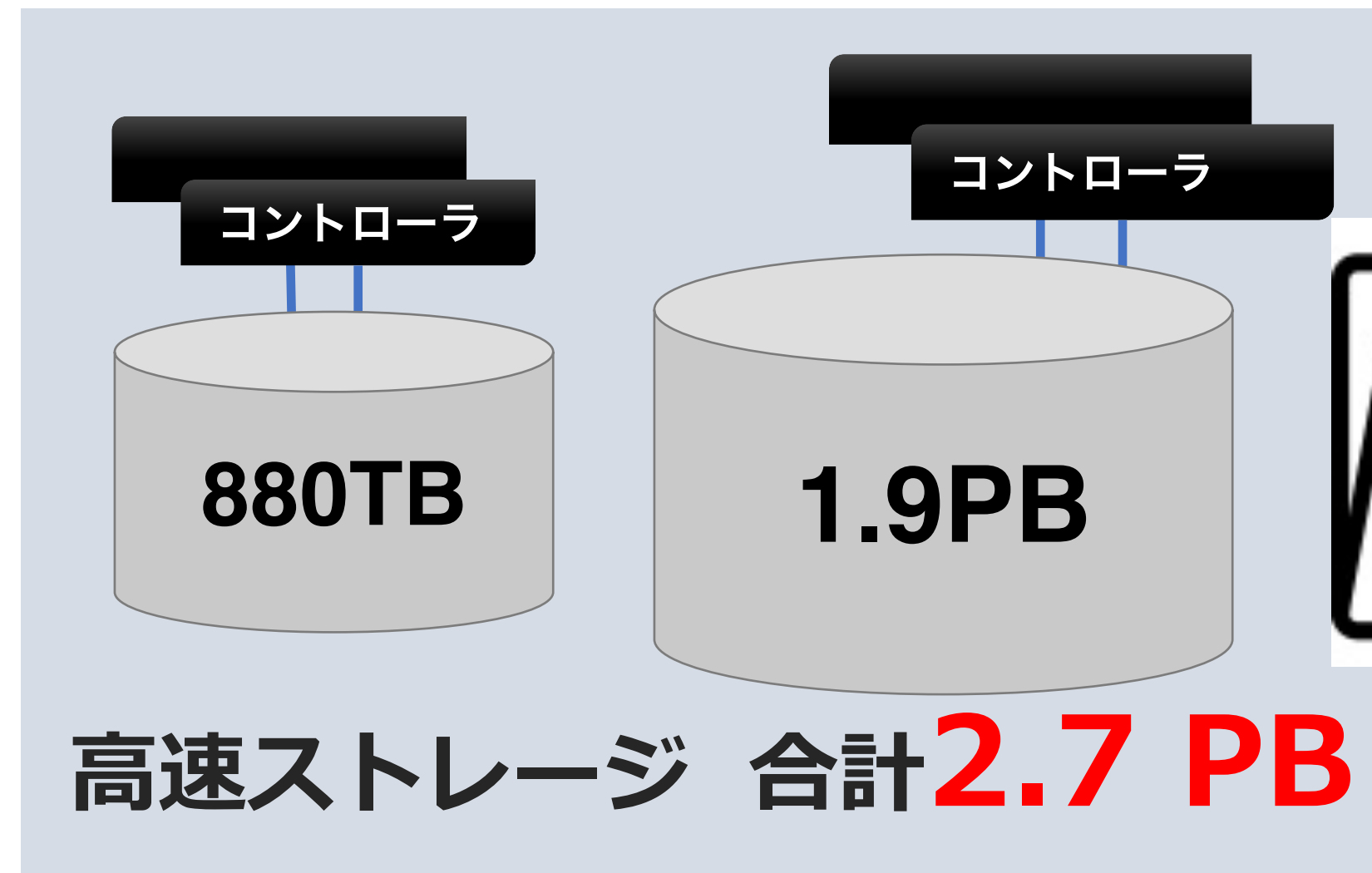
4TB Mem 80 cores
ldas-smp.nibb.ac.jp

共有メモリ型計算サーバ

527GB Mem 64 cores
diaf-smp1.nibb.ac.jp

共有メモリ型計算サーバ

527GB Mem 64 cores
diaf-smp2.nibb.ac.jp



PBSによるジョブ実行の流れ

- 実行したいコマンドをシェルスクリプトに書く

test_qsub.sh

```
#!/bin/bash
cd ${PBS_O_WORKDIR}

diamond blastp --db spo.dmnd --out sce.tab --outfmt 6 --query sce_prot.fasta
```

- シェルスクリプトのファイルをqsubコマンドに渡して実行

```
$ qsub test_qsub.sh
```

```
5812.bias5-adm
```

- qstat でジョブの実行状況を確認

```
$ qstat
```

実習: qsubの実行

- ディレクトリの内容、qsub_test.sh ファイルの内容を確認

```
$ ls
```

```
$ less test_qsub.sh
```

- qsubコマンドに qsub_test.sh ファイルを渡して実行

```
$ qsub test_qsub.sh
```

- qstat で実行状況を確認

```
$ qstat
```


実習：結果の確認

- 結果ファイルの確認

```
$ ls  
$ less sce.tab
```

- 標準出力ファイル、標準エラー出力ファイルの確認

```
$ less test_qsub.ejobID  
$ less test_qsub.ojobID
```



jobID は、
qsubした時についた
ジョブの番号

- ジョブ実行状況の確認

```
$ tracejob jobID
```


ジョブを投入するコマンド qsub

- PBSにおけるジョブ:
実行するコマンドやアプリケーションを記述したシェルスクリプト形式のタスク

```
$ qsub [オプション] スクリプトファイル
```

```
$ qsub diamond.sh  
5812.bias5-adm
```

- ジョブを投入するとジョブにID(**jobID**)が付与される
- **jobID** の後ろに「.bias5-adm」が表示される(当PBSの親ノード名)

PBSジョブのためのシェルスクリプトファイル（ジョブファイル）

diamond.sh

```
#!/bin/bash
#PBS -q small
#PBS -l ncpus=4
#PBS -l mem=12gb

cd ${PBS_O_WORKDIR}

diamond blastp --threads ${NCPUS} --db /bio/db/diamond/swissprot/swissprot \
--outfmt 6 --query ./sce_prot.fasta --out ./sce2.tab --sensitive
```

←①シェルの指定

←② qsubオプション

←③ジョブ投入ディレクトリへの移動

←④実行するコマンド

① シェルの指定 : ジョブスクリプトのシェルの種類を指定

② qsubオプション

行の先頭を #PBS で始めると、qsubコマンドのオプション指示として処理される

③ ジョブ投入ディレクトリへの移動

qsubコマンドを実行したディレクトリに移動（変数 `${PBS_O_WORKDIR}` に qsub実行ディレクトリ名が自動で設定される）

これをしないとホームディレクトリにいる状態で続くコマンドを実行しようとする

④ 実行するコマンド : コマンド、パスの設定、変数のセット等スクリプト本体を記述

qsubのオプション

- #PBS に続けてジョブスクリプト内に書いても、qsubコマンドに付けてもよい（コマンド付けの方が優先）

オプション	説明	指定なしのデフォルト動作
-q <i>queue_name</i>	キューを指定	smallキューに投入される
-N <i>job_name</i>	ジョブの名前を指定	ジョブファイル名が使われる
-e <i>file_name</i>	標準エラー出力を <i>file_name</i> へ出力	<i>job_name.ejobID</i>
-o <i>file_name</i>	標準出力を <i>file_name</i> へ出力	<i>job_name.ojobID</i>
-j [oe, eo]	ジョブの標準出力ファイルと標準エラー出力ファイルを結合する oe : 標準出力ファイルに結合 eo : 標準エラー出力ファイルに結合	
-J X-Y[:Z]	ジョブアレイ範囲指定（アレイジョブの実行時に使用）	
-l <i>resource_list</i>	ジョブで使用するCPU数、メモリ、実行サーバを指定	
-v <i>variable_list</i>	環境変数等の設定をジョブに持たせる 例) -v var1=10, "var2='A,B'"	
-S <i>shell_name</i>	ジョブスクリプトのシェルを指定	□グインシェル
-V	ログイン環境の環境変数を全て設定した状態でジョブを実行	
-I	インタラクティブ・バッチジョブを投入	
-m <i>option</i>	メールを送信する条件を設定 a, b, e のうち1つ以上の文字を指定します。(デフォルト -m a) a : ジョブが中止(aborted)された場合にメールを送信します。 b : 実行開始(begin)時にメールを送信します。 c : メールを送信しません(not)。 e : ジョブの終了時にメールを送信します。	当システムでは未実装

当システムにおけるPBSのキュー構成

	分散処理用計算機クラスタ (800core)				共有メモリ型計算サーバ		
queue name	small (default)	medium	large	blast	smpls	smpm	smp
キューの特徴	短時間・並列多	中規模	長時間	for BLAST nr/nt	中メモリ	大メモリ	最大メモリ
Avairable hosts	bias5-node01～ bias5-node20	bias5-node01～ bias5-node20	bias5-node01～ bias5-node20	bias5-node01～ bias5-node20	bias5-smp ldas-smp diaf-smp1 diaf-smp2	bias5-smp ldas-smp	bias5-smp ldas-smp
最長実行時間 / ジョブ	6h	72h	no limit	no limit	no limit	no limit	no limit
最大同時実行ジョブ数 / キュー	no limit	no limit	no limit	no limit	16	6	2
最大同時使用CPU数 / キュー	580	200	20	120	176	96	72
最大同時実行CPU数 / ユーザ	480						
最大同時実行CPU数 / ユーザ/キュー	300	150	10	80	no limit	no limit	no limit
デフォルト使用CPU数 / ジョブ	1	1	1	20	1	1	1
最大同時使用CPU数 / ホスト	40				72/80/64/64	72/80	
最大利用メモリ / ジョブ	96GB	96GB	96GB	188GB	500GB	1TB	3TB
デフォルト使用メモリ量 / ジョブ	3.5GB	3.5GB	3.5GB	98GB	250GB	500GB	1500GB

- キュー指定がないとデフォルトキューである small に投入される
- ジョブの 同時実行数/キュー or 1ユーザ を超えて投入されたジョブは実行待ちとなる

同時実行数について

	分散処理用計算機クラスタ (800core)				共有メモリ型計算サーバ		
queue name	small (default)	medium	large	blast	smps	smpm	smpi
キューの特徴	短時間・並列多	中規模	長時間	for BLAST nr/nt	中メモリ	大メモリ	最大メモリ
Avairable hosts	bias5-node01～ bias5-node20	bias5-node01～ bias5-node20	bias5-node01～ bias5-node20	bias5-node01～ bias5-node20	bias5-smp ldas-smp diaf-smp1 diaf-smp2	bias5-smp ldas-smp	bias5-smp ldas-smp
最大同時実行ジョブ数 / キュー	no limit	no limit	no limit	no limit	16	6	2
最大同時使用CPU数 / キュー	580	200	20	120	176	96	72
最大同時実行CPU数 / ユーザ	480						
最大同時実行CPU数 / ユーザ/キュー	300	150	10	80	no limit	no limit	no limit
デフォルト使用CPU数 / ジョブ	1	1	1	20	1	1	1
最大同時使用CPU数 / ホスト	40				72/80/64/64	72/80	

smallキューでは...

- キューあたりの同時実行CPU数 580
- 1ユーザあたりの同時実行CPU数 300

キューをまたいだ1ユーザあたりの同時実行CPU数 480

メモリ制限について

- ジョブのメモリ利用量がメモリ指定量を超えると強制終了
- **small, medium, large, blast キューのみ**
- 標準エラー出力に出されるメッセージ:

=>> PBS: job killed: mem 3977304kb exceeded limit 3686400kb

	分散処理用計算機クラス (800core)			
queue name	small (default)	medium	large	blast
キューの特徴	短時間・並列多	中規模	長時間	for BLAST nr/nt
Avairable hosts	bias5-node01~ bias5-node20	bias5-node01~ bias5-node20	bias5-node01~ bias5-node20	bias5-node01~ bias5-node20
Max.memory usage / job	96GB	96GB	96GB	188GB
Default memory size / job	3.5GB	3.5GB	3.5GB	98GB

CPU数	総メモリ量
40/1台	198GB/1台

- small, medium, large キューの上限まで設定すると1台あたり2ジョブ
- blast キューの上限まで設定すると1台あたり1ジョブ

マシンのリソースとリソース指定方法

マシン名	CPU数	総メモリ量
bias5-node01~20	40/1台	198GB/1台
bias5-smp	72	3TB
ldas-smp	80	4TB
diaf-smp[1, 2]	64/1台	527GB/1台

- qsubオプション **-l**

`-l resource_name=value[,resource_name=value ...]`

リソース名=値

複数つける場合は一行ずつ書くか、カンマで区切る `#PBS -l ncpus=8,mem=48GB`

```
#PBS -l ncpus=8
```

CPU数を指定

この値は変数 `#{NCPUS}` に自動でセットされるので、CPU数を指定できるアプリケーションにはこれを使うこと

```
#PBS -l mem=48GB
```

最大利用メモリ量を指定

キューの選び方について

- mediumキュー(~72h)が混んでいることが多いが、smallの制限時間である6h以内に終わっているジョブがほとんど
- まずはsmallキューに、デフォルト条件で投入してみて、時間やメモリ利用量がオーバーするようなら他のキュー利用を検討いただきたい
- blastキューは最大188GBまでメモリを使えるので、smp[s, m, l]キューが混んでいる場合はこちらを利用してみてください

PBSジョブのためのシェルスクリプトファイル（ジョブファイル）

diamond.sh

```
#!/bin/bash
#PBS -q small
#PBS -l ncpus=4
#PBS -l mem=12gb

cd ${PBS_O_WORKDIR}

diamond blastp --threads ${NCPUS} --db /bio/db/diamond/swissprot/swissprot \
--outfmt 6 --query ./sce_prot.fasta --out ./sce2.tab --sensitive
```

←①シェルの指定

←② qsubオプション

←③ジョブ投入ディレクトリへの移動

←④実行するコマンド

① シェルの指定 : ジョブスクリプトのシェルの種類を指定

② qsubオプション

行の先頭を #PBS で始めると、qsubコマンドのオプション指示として処理される

③ ジョブ投入ディレクトリへの移動

qsubコマンドを実行したディレクトリに移動（変数 `${PBS_O_WORKDIR}` に qsub実行ディレクトリ名が自動で設定される）

これをしないとホームディレクトリにいる状態で続くコマンドを実行しようとする

④ 実行するコマンド : コマンド、パスの設定、変数のセット等スクリプト本体を記述

キューとジョブの状況を確認する

\$ qstat オプション

<code>qstat -Q</code>	全てのキューの状態を表示
<code>qstat -Q queue_name</code>	指定したキューの状態を表示
<code>qstat -Qf queue_name</code>	指定したキューの設定と状態を表示
<code>qstat</code>	全てのジョブを表示
<code>qstat -t</code>	全てのジョブを表示 (アレイジョブを展開)
<code>qstat -u user_name</code>	指定したユーザの全ジョブを表示
<code>qstat -i -u user_name</code>	指定したユーザの実行待ちジョブを表示
<code>qstat -f jobID</code>	実行待ちになっている原因を表示 (アレイジョブの場合は <code>jobID[]</code> と記述)
<code>qstat -r</code>	実行中の全てのジョブを表示
<code>qstat -nr1</code>	実行中の全てのジョブを実行ホストと共に1行で表示
<code>qstat -J</code>	アレイジョブを表示
<code>qstat -J -t -nr1</code>	実行中のアレイジョブを展開して詳細も表示
<code>qstat -x</code>	24時間以内のジョブ実行履歴を確認

qstatによる現在の状態の確認

```
$ qstat
```

```
bias5-adm:
```

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	S	Elap Time
362.bias5-adm	hiroyo	small	interpro	37276	1	1	3600mb	06:00	R	00:00
363.bias5-adm	hiroyo	small	interpro	145771	1	1	3600mb	06:00	R	00:00
364.bias5-adm	ideas	medium	dbupdate	145814	1	1	12gb	72:00	R	00:00
365.bias5-adm	sgiadm	small	testjob	145858	1	1	3600mb	06:00	R	00:00
366.bias5-adm	sgiadm	small	testjob	145902	1	1	3600mb	06:00	R	00:00
367.bias5-adm	nyacco	smps	myscript	145946	1	1	250gb	--	R	00:00
368[] bias5-a	hpe	blast	blastjob1	145990	1	1	3600mb	06:00	B	00:00
380.bias5-adm	sgiadm	small	TEST.sh	146033	1	1	8gb	06:00	R	00:00
381.bias5-adm	hogehe	smpm	Trinity_rn	--	1	1	500gb	--	Q	--

○ジョブID

○ジョブのステータス

R : 実行中

B : アレイジョブ

Q : 待機中

Eqw : 何らかのエラー

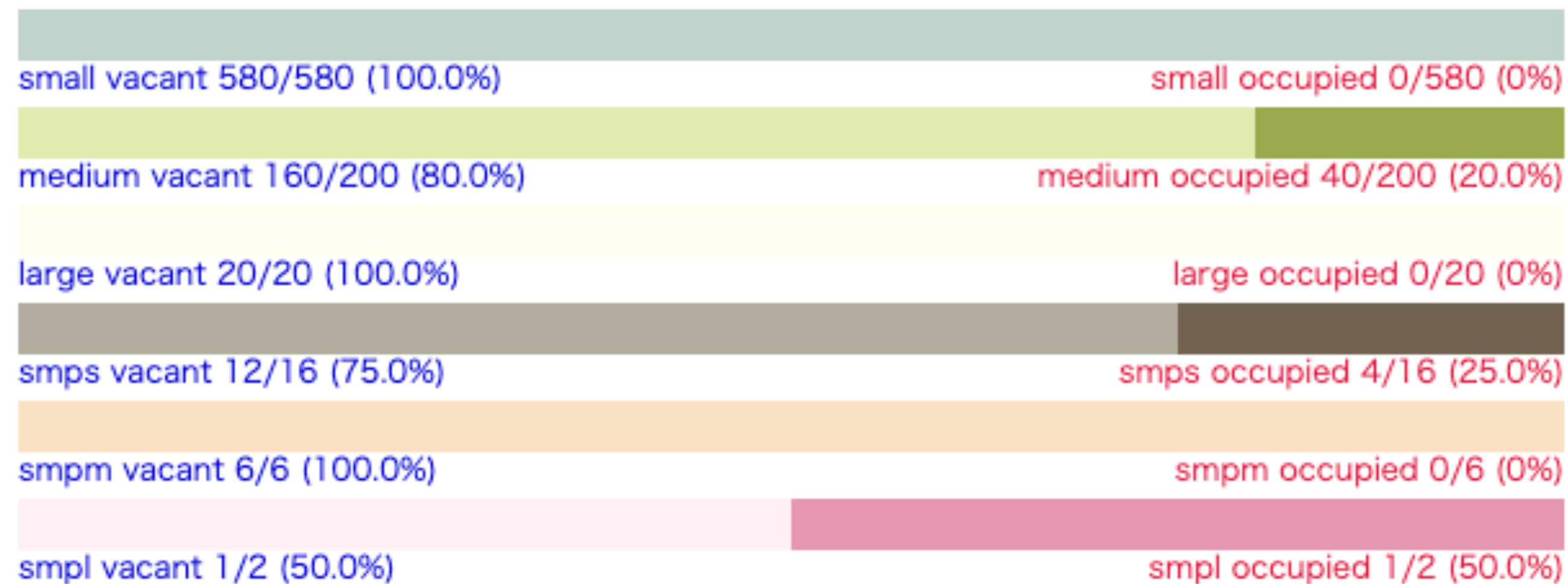
ノード/キュー稼働状況 (要VPN)

- 各ノード稼働状況

<http://bias5-web.nibb.ac.jp/job/index3.html>

- キューの投入状況

<http://bias5-web.nibb.ac.jp/job/index2.html>



node	status	#job	#Mem	Memory	run
node Total		78/800	1332/3768		
catm	free	0/32	0/504		
cats1	state-un	0/24	0/94		
cats2	free	0/24	0/94		
diaf-smp1	free	0/64	0/503		
diaf-smp2	free	48/64	500/503		
ldas-smp	free	0/80	0/4029		
node01	free	20/40	188/188		
node02	free	1/40	60/188		
node03	free	3/40	180/188		
node04	free	3/40	180/188		
node05	free	3/40	180/188		
node06	free	3/40	180/188		
node07	free	3/40	180/188		
node08	free	2/40	120/188		
node09	free	0/40	0/188		
node10	job-busy	40/40	64/188		
node11	free	0/40	0/188		
node12	free	0/40	0/188		
node13	offline	0/40	0/188		
node14	free	0/40	0/188		
node15	free	0/40	0/188		
node16	free	0/40	0/188		
node17	free	0/40	0/188		
node18	free	0/40	0/188		
node19	free	0/40	0/188		
node20	free	0/40	0/188		
smp	free	40/72	500/3023		
catl	free	0/32	0/1009		

キューの選び方について2

- qstatや稼働状況を見てリソース利用量を調整し、空いている/即実行されるキューにジョブを投入しましょう
- 分割できるものは分割してアレイジョブにし、全体の処理時間を短縮しましょう

ジョブを削除する

```
$ qdel jobId
```

- ジョブIDを指定して削除
- 自分のジョブのみ

```
$ qdel 5812
```

ジョブを選択する

```
$ qselect オプション
```

<code>qselect -u <i>user_name</i></code>	指定したユーザの全ジョブIDを表示
<code>qselect -u <i>user_name</i> -q <i>queue_name</i></code>	指定したユーザの指定キューに投入されたジョブIDを表示
<code>qselect -s Q -u <i>user_name</i></code>	指定したユーザの実行待ちジョブIDを表示
<code>qselect -s R -u <i>user_name</i></code>	指定したユーザの実行中ジョブIDを表示

ジョブを選択して一括削除する

```
$ qdel `qselect オプション`
```

- qdel + qselectで一括削除
- qdelを行う前にqselectで該当ジョブを確認する(想定外の削除をしないように)
- ``` バッククオート → 中に書かれたコマンドの実行結果で置換される
- **自分の全てのジョブを一括削除する**

```
$ qselect -u my_user_name  
$ qdel `qselect -u my_user_name`
```


.e ファイルと .o ファイル

- 標準エラー出力 `job_name.ejobID`
- 標準出力 `job_name.ojobID`
- ジョブが実行されたノード上で保持され、ジョブの終了後にqsubされたマシンに送られる
- 何らかの理由でこれらのファイルが送信できないこともあり、それを見たい場合はメールでジョブIDをご連絡ください

再実習：ジョブの投入と確認

```
$ less diamond.sh
```

diamond.sh

```
#!/bin/bash
```

←①シェルの指定

```
#PBS -q small
```

```
#PBS -l ncpus=2
```

```
#PBS -l mem=6gb
```

←② qsubオプション

```
cd ${PBS_O_WORKDIR}
```

←③ジョブ投入ディレクトリへの移動

```
diamond blastp --threads ${NCPUS} --db /bio/db/diamond/swissprot/swissprot \
```

```
--outfmt 6 --query ./sce_prot.fasta --out ./sce2.tab --sensitive ←④実行するコマンド
```

② qsubオプション

small キュー

2CPUを確保

6GBメモリを確保

再実習：ジョブの投入と確認

1. diamond.sh の内容を確認
2. ジョブ投入
3. ジョブの状況を確認
4. qstat に現れなくなったら(ジョブが終了したら)、tracejob の内容を確認

```
($ less diamond.sh)  
$ qsub diamond.sh  
$ qstat -u your_user_name  
$ qstat  
$ ls  
$ tracejob job_ID
```

ジョブの履歴を確認する

```
$ tracejob [オプション] jobID
```

- ジョブの履歴が見られるのは当日のみ
- 実行中の状態も見ることができる
- 所要時間、使った総メモリ量などがわかる

```
resources_used.walltime=  
resources_used_mem=
```

```
$ tracejob 6278
```

```
Job: 6278.bias5-adm
```

```
06/13/2018 11:26:30 L Considering job to run
```

```
06/13/2018 11:26:30 S enqueueing into small, state 1 hop 1
```

```
06/13/2018 11:26:30 S Job Queued at request of sgi@bias5-login, owner = sgi@bias5-login, job name = TB, queue = small
```

```
06/13/2018 11:26:30 S Job Run at request of Scheduler@bias5-adm on exec_vnode (bias5-node10:mem=3145728kb:ncpus=4)
```

```
06/13/2018 11:26:30 S Job Modified at request of Scheduler@bias5-adm
```

```
06/13/2018 11:26:30 L Job run
```

```
06/13/2018 11:36:30 S Obit received momhop:1 serverhop:1 state:4 substate:42
```

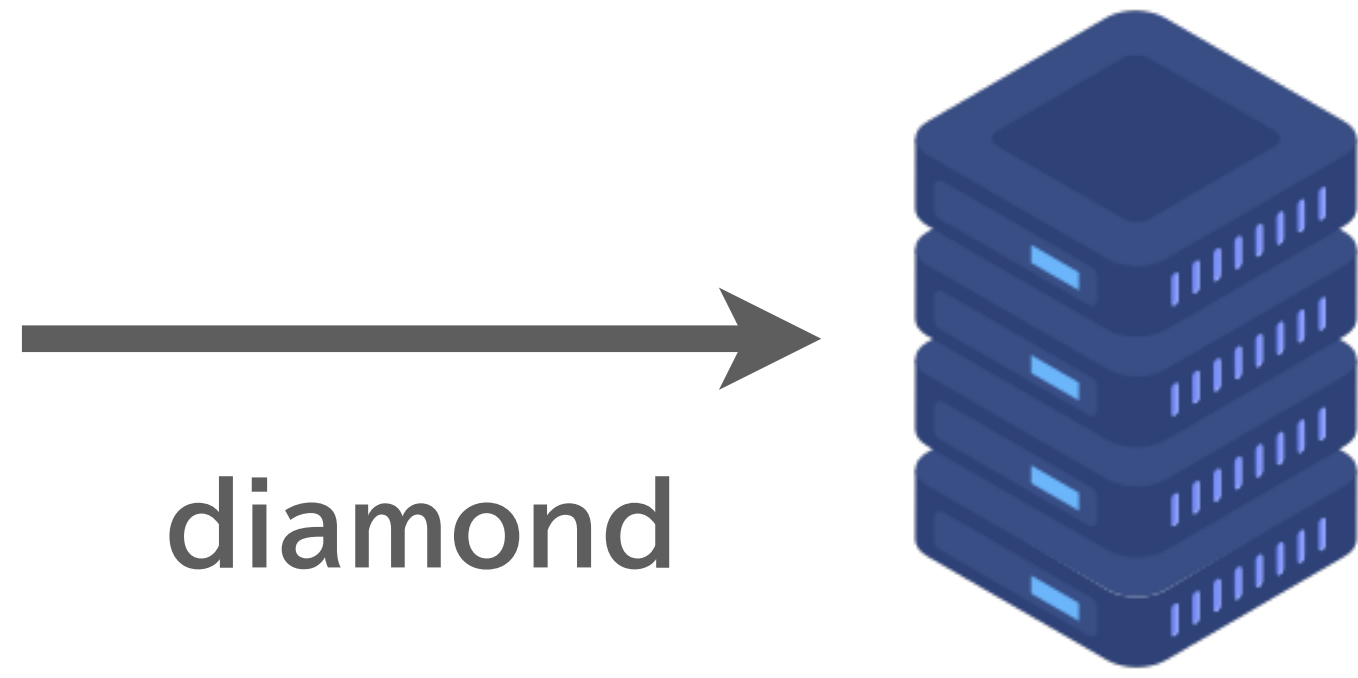
```
06/13/2018 11:36:30 S Exit_status=0 resources_used.cput=00:00:00
```

```
resources_used.mem=4000kb resources_used.ncpus=4 resources_used.vmem=357328kb resources_used.walltime=00:10:00
```

- tracejobでエラー原因がわかることもある
- 過去ジョブのtracejob結果を知りたい場合もメールでお知らせください

アレイジョブ

```
>spo:NP_001018179.1 hydroxymethylbilane synthase (predicted) [Schizosaccharomyce
MPSCTSFPIGTRKSKLAVIQSEI IREELEKHYPHLEFP IISRDTIGDEILSKALFEFKRQLAKSLWTRELEALLVTNQCR
ILVHSLKDLPEMPDGMVIACIPKRSCPLDAIVFKAGSHYKTVADLPPGSVVGTSIRRRALLARNFPHLRVFDIRGNVG
TRLAKLDAPDSQFDCLVLAAGLFRGLKDRIAQMLTAPFVYAVGQALAVEVRADDKEMIEMLKPLQHQETLYACLAE
RALMKRLQGGCAIPIGVQTDVLAISNSSYRISLLGTVLSADGLRAAFGNAEAVVSSEEEAEELGITVALALLKNGAGPIL
EEHQRSSDSEESLKNY
>spo:NP_001018181.1 poly(A) polymerase Cid14 [Schizosaccharomyces pombe 972h-]
MGKKSVSFNRRNYYKRNKNERTEPLRRIFKNDKPSKFKSKRKEKDKNSDAYDEMLLNNFTLLDQEEPMEIGSKSRND
NDSEGIRDKGGVEISNKNDPYIQFGKADPLEPLEKPDLPPEEAIKRGEP TILLGIPKREGRKTNPVHDKAVENNSDFIKFD
WNSDEDEDSVSNKSKNNE SLKSSKNEIPGFMQRGRFFHEANEKSDSNRKRKRQAYELDSQSCPWHRYKVEREVSRI
FHQDILHFIDYITPTPEEHAVRKTLSRINQAVLQKWPDVSLYVFGSFETKLYLPTSDLDLVIISPEHHYRGTKKDMFVL
AHHLKKLKLASEVQVITANVPIIKFVDPLTKVHVDISFNQPGGLKTCVVNGFMKKYPALRPLV I I KHFLNMRALNEV
FLGGLSSYAIVCLVVSFLQLHPRLSTGSMREEDNFGVLLLEFLELYGKQFYDVGIAVHNGGFYFSKKMGWLKPNQPY
LLSIQDPVDFQNDVSKSSRGLLRVKATFANGFDLLTSKLYALASRIEREGVNRVKDFPSILSTILSVDEGVRQHREHMLK
CYKNNPVPLEPLVEVDALASIDVDKLPQDVGLQYVEDESDETDAAKDDLKVNES IETNGHENFQKQALTSTGEQSS
SNSRANPSKLFNISSDSEDEVPIIEDTTASDEESRAKKIRKRF
>spo:NP_001018183.2 pre-ribosomal protein Loc1 (predicted) [Schizosaccharomyces
MVKKSKPKNQIRVEDLDLPLKLNSTSKNPQTKIQKGGKKGKIFAETKDDLQNILNQVTEYELDDKIKSKLQVAHEREAVFS
KQSDRKISNNKADKKTGRKNEK
>spo:NP_001018187.2 dual specificity protein kinase Lkh1 [Schizosaccharomyces
MHSKRRRNHAPDWQDFYKNGVPQEVIVIEDSASPRLTPLNPPFVSHVQLQSFVPPQPPSSSPSTGTVAVPINGANAV
YPSTNSVSLPQSYDPWLDANGVVPLPHDVASHPSYMQSPTS YHACSNNQSPFPHSHHPLHNPLPVSCQPVLRRPPVPQ
VPSHWYPVSLPSPNLPHQPI SKPPVIPNLPKLQVHPNRLPHPIHNHPYSSPTSYPPLCPATYCPSNPPQLAPATAIAPS
SQSSQHKS VNYSVTPSSINNHTAVPLSPTLAVWLPMTQPTFQPPSANVYQPASNANQVITPVVISDYRPPKRRKRAAWPP
YKKVDRVNVVVDHTTAFDPSTFDDDDGHYKVPVNSK FANRYTVVRL LGHGTFGKVIQCYDQSTGRHCAIKVTRAI PKYR
EASLIELRVLQTI AHS DPTNENKCIQLRDYFDYRKHCIVTDLFGWSVDFLKNNNYIPFPLKHIQMLSQQLFKSVAF LH
SLGLVHTDLKPENVLVSNASRTIRLPYRNYSQKVLNSCEIRLIDFGSATFEDEYHSSVVSTRHYRAPEIILGLGWSYPC
DVWSIGCILVELFTGQALFQTHEDESEHLMMKILGPFDRNMISRSRSTSQRFFKSDGKVRYPLSNTPKKSINYLQSLQT
LEQIFAVSSPEVALLDLKLVFVYDPKRRITAKEALWHPFFTQPISSNL
>spo:NP_001018191.1 homologous-pairing protein 2 [Schizosaccharomyces pombe 972h
MAKAKEVKAKPIKGEAEKLVYEYLRKTNRPYSATDV SANLKNVSKQVAQKALEQLRDTGLIHGKLYGKQSVFVCLQDD
LAAATPEELAEMEKQIQELKDEVS VVKTYKEKCIELQALNNSLSPA EIREKIQSIDKEIEETSSKLES LRNGTVKQISK
EAMQKTDKNYDFAKKGFSNRKMFYDLWHLITDSLENPKQLWEKLGFEETEGPIDLN
>spo:NP_001018193.1 Rho family GTPase Rho3 [Schizosaccharomyces pombe 972h-]
MSSCFGSKKPIYRKIVILGDGAAGKTSLLNVFTKGYFPQVYEPTIFENYIHDIFVDGNSIELSLWDTAGQEEYDQLRSL
SYSDTHVIMICFAVDSRDSLENVITKWLPEVSSNCPGVKLVLVALKCDLRGADEEQVDHSKIIDYEEGLAAAKKINAVRY
LECSAKLNRGVNEAFTEAARVALAAQPRGTDGADESHGTGCIIA
>spo:NP_001018195.1 UBA domain protein Mud1 [Schizosaccharomyces pombe 972h-]
MNNLTPENIRQTILATPFLNLRIRTEFPQLAAVLNDPNAFATTWQSINASQLLQIPSS TYSMGMPFS SEDDLFDVEVQRR
IEEQIRQNAV TENMQSAIENHPEVFGQVYMLFVNVEINGHKVKAFVDSGAQATILSADCAEKCGLTRLLDTRFQGVAKGV
GMAKILGCVHSAPLKIGDLYLPCRFTVIEGRDVMMLGLDMLRRYQACIDLENNVLR IHGKEIPFLGESEIPKLLANVEP
SANAHGLGIEPASKASASSPNPQSGTRLGTKESVAPNNEGSSNPPSLVNPPTDPGLNSKIAQLVSMGFDPLEAAQALDAA
NGDL DVAASFL
>spo:NP_001018218.1 SMN family protein Yip12 [Schizosaccharomyces pombe 972h-]
MPSKRKRNPLOYQTSGLDEETNQSAFPQIDNNSASESLEYDIPLDGLDYLATVREEARKLVFVAARREPETRETIPL
RKLEIEAGKKS FDFPFLRYLLNIIDKEGERLEQYMESSSLDASILPKNLQQWRVYIEHKAPCWAILAVVDLATVLEI LESL
SSWLEKDAIDLQSQWIFCFYKLPPELLNGEDISTLRSVLKSLRSTHFSFPALQMSASALQAVLVYRYGQKDLFQT
>spo:NP_001018219.2 ubiquitin-protein ligase E3 (predicted) [Schizosaccharomyce
MDSPLLEVLQVQVEKLI SPSLRFILAYFTHRYPRFLLRAYNSFDGIYLLVLLLEKSQLKKNATSVERRFQLKRVIA
VRDSSIIAEFPQESSESATSLNGIDVLKFLTYCIPYLLKCESLTTVKENHTAVSILSLQARDKQK GALSVFYSKIKI
LLVRLKILHFVFR LIRKSN TYLQWLYLLYALGKTPYTNLADHILRQRVIY NVENIHSRKLISTREKSSLLTSIADHSM
EGFLII IQLIDWWQSNNYESHKKGEVAFTELAPPKLPFEINVSTTDICKICGEKIKNPAVLSTGFVFCYPCIQVWLQRH
PFKCPVTNLELSRKGESFWRLMI
```



my.fa

アレイジョブ

同じような作業を
1つのスクリプトファイルで
独立して実行させる

```
>spo:NP_001018181.1 poly(A) polymerase Cid14 [Schizosaccharomyces pombe 972h-]  
MGKKSVSFNRNNYKRRKNERTEPLRRIFKNDKPSKFKSKRKEKDKNSDAYDEMLLNNNFLLDQEEPVEIGSKKSRND  
NDSEGIRDKGGVEISNKNDPYIQFGKADPLEPLEKPDLPPEEAIKRGEPTILLGIPKREGRKTNPVHDKAVENNSDFIKFD  
WNSDEDEDSVSNKSKNNESLKKSSKNEIPGFMQRGRFFHEANEKSDSNRKRKRQAYELDSQSCPWHRQYKVEREVSRI  
FHQDILHFIDYITPTPEEHAVRKTLSRINQAVLQKWPDVSLYVFGSFETKLYLPTSDDLVIISPEHHYRGTKKDMFVL  
AHHLKKLKLASEVQVITANVPIIKFVDPLTKVHVDISFNQPGGLKTCVNVNGFMKKYPALRPLVIIKHFLNMRALNEV  
FLGGLSSYAIIVCLVVSFLQLHPRSTGSMREEDNFGVLLLEFLELYGKQFYDAVGIAVHNGGFYFSKMKMGWLKPNQPY  
LLSIQDPVDFQNDVSKSSRGLLRVKATFANGFDLLTSKLYALASRIEREGVNRVKDFPSILSTILSVDEGVRQHREHMLK  
CYKNNPVPLEPLVEVDALASIDVDKLPQDVGLQYVEDESDSDTDAKDDLKFNES IETNGHENFQKQALTSTGSEQSS  
SNSRANPSKLFNISSDSEDEVPIIEDTTASDEESRAKKIRKRF
```

my.1.fa

```
>spo:NP_001018179.1 hydroxymethylbilane synthase (predicted) [Schizosaccharomyce  
MPSCTSFPIGTRKSKLAVIQSEIIREELEKHYPHLEFPIISRDTIGDEILSKALFEFKRQLAKSLWTRELEALLVTNOCR  
ILVHSLKDLPEMPDGMVIACIPKRSCPLDAIVFKAGSHYKTVADLPPGSVVGTSIRRRALLARNFPHLRFVDIRGNVG  
TRLAKLDAPDSQFDCLVLAAGLFRGLKDRIAQMLTAPFVYAVGQALAVEVRADDKEMIEMKPLQHQTLYACLAE  
RALMKRLQGGCAIPIGVQTDVLAISNYSYRISLLGTVLSADGLRAAFGNAEAVVSSEEEAEELGITVALALLKNGAGPIL  
EEHQSSDSEESLKNY  
>spo:NP_001018183.2 pre-ribosomal protein Loc1 (predicted) [Schizosaccharomyces  
MVVKKSKPKNQIRVEDLDLPKLNSTKNPQTKIQKGGKKGKIFAETKDDLQNILNQVTYELDDKIKSKLQVAHEREAVFS  
KQSDRKISNNKADKKTGRKNEKK  
>spo:NP_001018187.2 dual specificity protein kinase Lkh1 [Schizosaccharomyces
```

my.2.fa

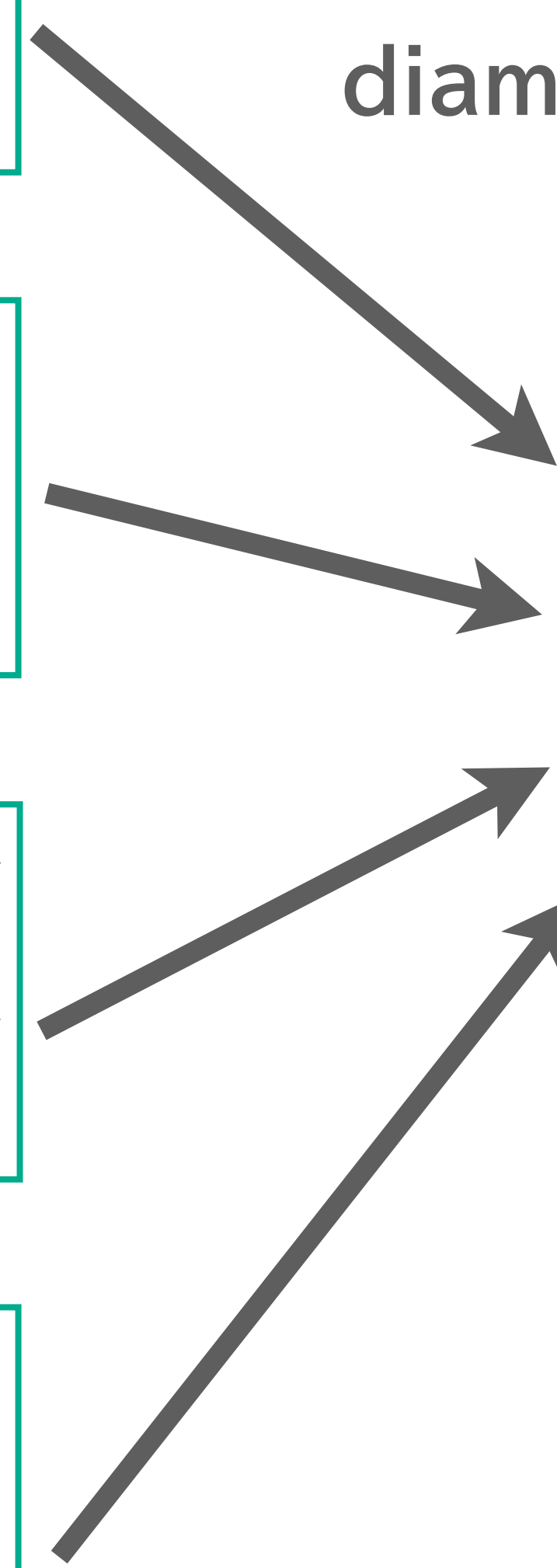
```
>spo:NP_001018187.2 dual specificity protein kinase Lkh1 [Schizosaccharomyces  
MHSLKRRRNHAPDWQDFYKNGVQEVIVIEDSASPRLTPNLPPFVSVHQLQSFVPPQPPSSSSPSTTGTVAVPINGANAV  
YPSTNSVSLPQSYDPWLDANGVVPLPHDVASHPSYMQSPTSYHACSNNQSPFPHSHHPPLHNPLPVSCQPVLRRPPVPQ  
VPSHWYPVSLPSPNLPHQPI SKPPVIVNLPKLQVHPNRLPHPIHNHPYSSPTSYPPLCPATYCPNPPQLAPATAIAPS  
SQSSQHKSVMYSVTPSSINNHTAVPLSPTLAVWLPMTQPTFPQPSANVYQPASNANQVITPVVISIDYRPPKRRKRAAWPP  
YKKVDRVNVVVDHDTAFDPSTFDDDDGHYKVVVNSKFNRYTVVRLGHGTFGKVIQCYDQSTGRHCAIKVTRAIKPKYR  
EASLIELRVLQTIASHDPTNENKCIQLRDYFDYRKHICIVTDLFGWSVDFLKNNNYIPFPLKHIQMLSQQLFKSVAFLH  
SLGLVHTDLKPENVLVSNASRTIRLPYRNYSQKVLNSCEIRLIDFGSATFEDEYHSSVVSTRHYRAPEIILGLGWSYPC  
DVWSIGCILVELFTGQALFQTHEDEHLCMMEKILGPFDRNMISRSSRTSQRFFKSDGKVRYPVLSNTPKKSINYLQSLQT  
LEQIFAVSSPEVALLDLLKVFVYDPKRRITAKEALWHPFFTQPISSNL
```

my.3.fa

```
>spo:NP_001018191.1 homologous-pairing protein 2 [Schizosaccharomyces pombe 972h  
MAKAKEVKAKPIKGEEAEKLVYEYLKRTNRPYSATDVSANLKNVSVKQVAQKALEQLRDTGLIHGKLYGKQSVFVCLQDD  
LAAATPEELAEMEKIQELKDEVSVVKTLYKEKCIELQALNNSLSPAIREKIQSIDKEIETSSKLES LRNGTVKQISK  
EAMQKTDKNYDFAKKGFNSRKKMFYDLWHLITDSLENPKQLWEKLGFEFTEGPIDLN  
>spo:NP_001018195.1 UBA domain protein Mud1 [Schizosaccharomyces pombe 972h-]  
MNNLTPENIRQITILATPFLNLRIRTEFPQLAAVLNDPNAFATTWQ SINASQLLQIPSSYSMGMPSFSEDDLFDVEVQRR  
IEEQIRQNAV TENMQSAIENHPEVFGQVYMLFVNVEINGHKVKAFVDSGAQATILSADCAEKCGLTRLLDTRFQGVAKGV  
GMAKILGCVHSAPLKI GDLYLPCRFTVIEGRDVMLLGLDMLRRYQACIDLENNVLR IHGKEIPFLGESEIPKLLANVEP  
SANAHGLGIEPASKASASSPNPQSGTRLGTKEVAPNNEGSSNPPSLVNPPTDPGLNSKIAQLVSMGFDPLEAAQALDAA  
NGDLLVAASFLL
```

my.4.fa

diamond



アレイジョブ

- qsubオプション **-J** 開始番号-終了番号
- ジョブファイル中で変数 **`\${PBS_ARRAY_INDEX}** に番号がセットされ、インクリメントしながら回数分実行される

```
#!/bin/sh
#PBS -J 1-100
#PBS -l ncpus=8
cd `${PBS_O_WORKDIR}`

diamond blastx --threads `${NCPUS}` --db /bio/db/diamond/nr/nr --outfmt 6 \
--query ./my.`${PBS_ARRAY_INDEX}`.fa --out ./out.`${PBS_ARRAY_INDEX}`.tab --sensitive
```

- my.1.fa ~ my.100.fa のファイルに対して、diamond検索を実行し、out.1.tab ~ out.100.tab というファイルに書き込む
- 連続した数字付きのファイルを用意する必要がある

アレイジョブのqstat見え方：*jobID[]*

- アレイジョブのみを確認する(オプション `-J`)

```
$ qstat -J
```

Job id	Name	User	Time Use	S	Queue
5162[].bias5-adm	diaarray	my_name	0	B	small

- サブジョブを含む全てのアレイジョブを確認する(オプション `-t`)

```
$ qstat -t
```

Job id	Name	User	Time Use	S	Queue
5162[].bias5-adm	diaarray	my_name	0	B	small
5162[1].bias5-adm	diaarray	my_name	00:00:00	R	small
5162[2].bias5-adm	diaarray	my_name	00:00:00	R	small
5162[3].bias5-adm	diaarray	my_name	00:00:00	R	small

qsub用スクリプト内でmodulesにあるプログラムを使う

- ジョブスクリプト内に動作条件を記述する

```
source /etc/profile.d/modules.sh  
module load program_name/version
```

```
#!/usr/bin/bash  
#PBS -q blast  
#PBS -l ncpus=40  
#PBS -l mem=188GB  
source /etc/profile.d/modules.sh  
module load Trinityrnaseq/2.14.0  
cd ${PBS_O_WORKDIR}  
Trinity --verbose --max_memory 188G --seqType fq \  
  --samples_file rnaseq_data.txt --CPU ${NCPUS} \  
  --output trinity_out --min_kmer_cov 2
```

様々な条件下でのqsub

- conda

```
#!/bin/bash
#PBS -q medium
source ${HOME}/miniconda3/etc/profile.d/conda.sh
conda activate base
cd ${PBS_O_WORKDIR}
python3 my_script.py
conda deactivate
```

←①conda.shをロード

←②conda環境を起動

←③conda環境を終了

- singularity

```
#!/bin/bash
#PBS -q small
cd ${PBS_O_WORKDIR}
source /etc/profile.d/modules.sh
module load singularity
singularity exec /bio/container/BUSCO/5.1.3/busco_5.1.3--pyhdfd78af_0 \
busco -i my.fa -m tran -o b_result -l eukaryota_odb10
```

ジョブが失敗する時は

PBS上でのエラー:

- ステータスEqwになってしまう
- なぜかステータスQのまま実行されない

`qstat -f` や `tracejob` で原因を探る

```
$ qstat -f jobID
$ tracejob jobID
```

プログラムでのエラー:

`.e` ファイルと `.o` ファイルを見て原因を探る

`.e` ファイルと `.o` ファイルができない

ジョブが実行されたノード上には残っていることが多いのでメールでご連絡ください

wikiのTips&FAQも参照ください

PBSコマンドのヘルプ

```
$ man command_name
```

- PBSのコマンド群には、man コマンドによるヘルプ表示機能がある
- オプションの働き等の確認

```
$ man qsub
```

```
$ man qstat
```

```
$ man tracejob
```

wiki

- 生物情報解析システムwiki

<https://biaswiki.nibb.ac.jp/menu/index.php>

- お問い合わせ先

support@nibb.ac.jp

その他 Advanced 追加情報

- MPI, mpiccコンパイラ有ります
- AlphaFold使えます