SCLS計算機システム講習会

GHOST-MP 実習

独立行政法人理化学研究所
HPCI計算生命科学推進プログラム

SCLS計算機システムの GHOST-MP

GHOST-MP

BLAST のように遠縁のホモログを検出可能なホモロジー検索ツールである GHOSTX を「京」で高速化したもの。

GHOST-MP ver.201311

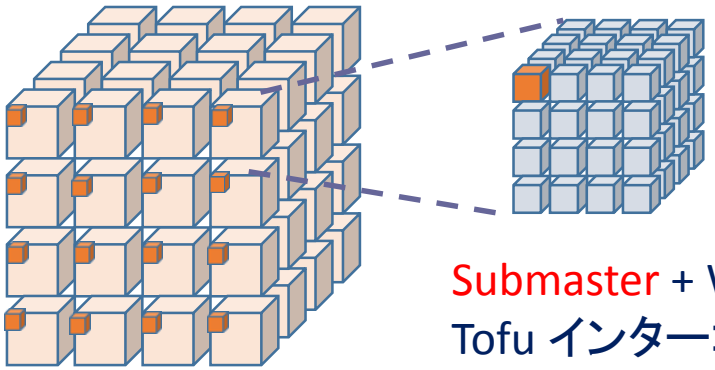
- OpenMP　node内のスレッド並列
- MPI
  - master – worker 方式による多数のクエリファイルの並列分散処理
  - Tofu 高機能バリア通信によるデータベースの高速 Broadcast 通信
- ファイル I/O

  master - submaster – worker 方式（3階層）による
  ファイル I/O の submaster での処理

  ➡ 全 worker によるファイル I/O の競合を回避

GHOST-MP

Tofu 高機能バリア通信によるデータベースの高速 Broadcast 通信



Master は rank=0 の 1 node だけ
赤が Submaster（数十～数百 node）
それ以外はすべて Worker（数万 node）

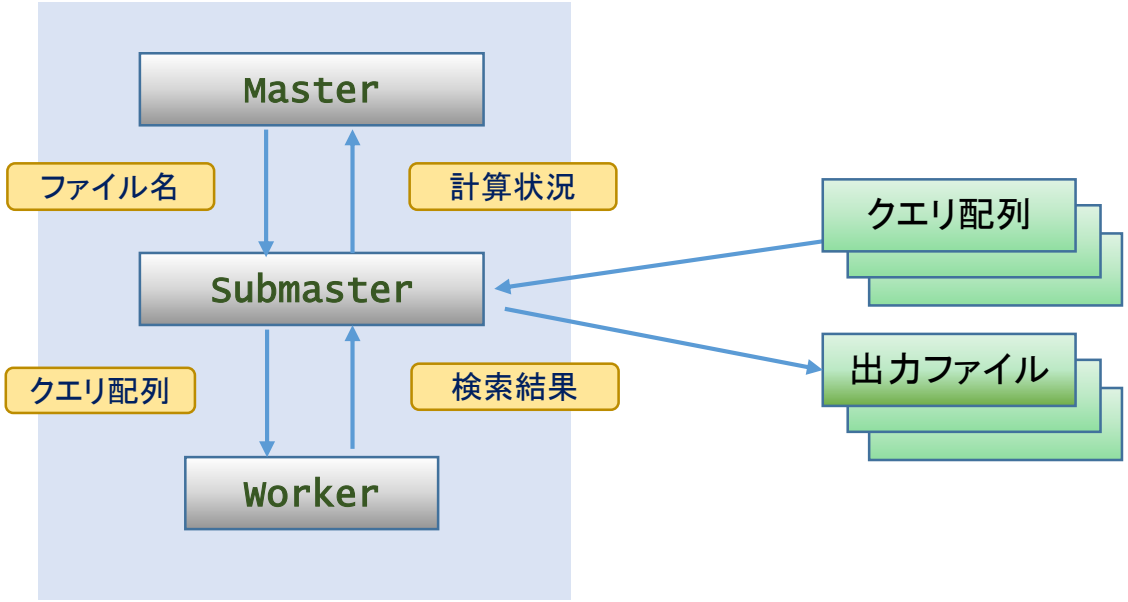
Submaster + Workers
Tofu インターコネクトを考慮した直方体内で
データベースの高速 Broadcast 通信

赤は Submaster
Tofu 座標空間で全系を直方体分割

- Submaster だけでデータベース（一般には数GB～数十GBのサイズ）を read することによりファイル入力の負荷削減
- データベースはチャンクという単位に分割。各直方体で担当するチャンク番号を限定して、さらにファイル入力の負荷削減

GHOST-MP

Master – Submaster – Worker 方式（3階層）による
ファイル入出力の Submaster での処理

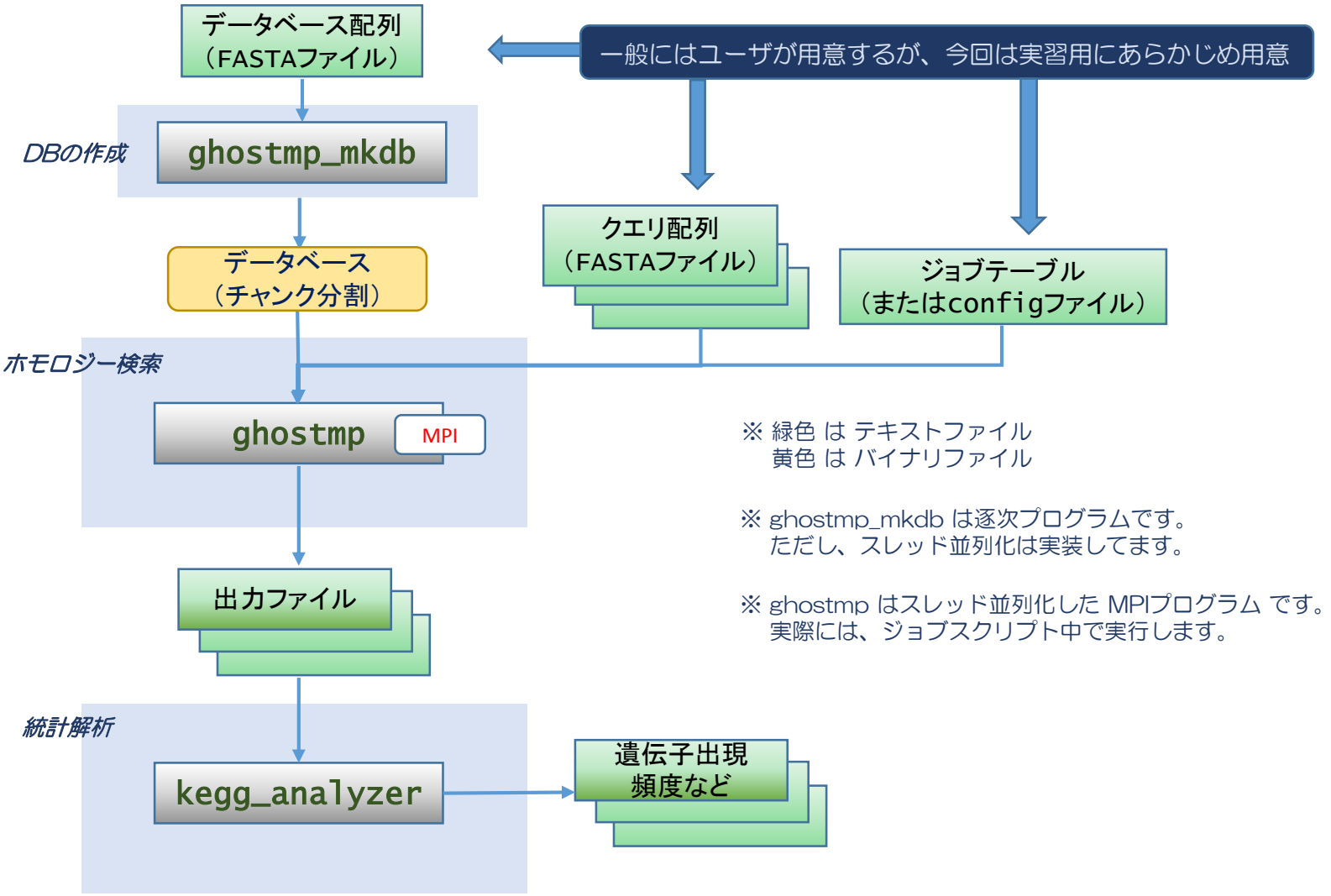


➡ 全 Worker によるファイル入出力の競合を回避

実習のながれ

実習用にあらかじめ用意した クエリ配列 と データベース配列 を使って、
ホモロジー検索の並列分散処理を実行し、検索結果の統計解析を行います。
見つかった遺伝子の phylum（門）の出現頻度などを計算します。

GHOST-MP コマンドの実行のながれ



※ 緑色 は テキストファイル
黄色 は バイナリファイル

※ ghostmp_mkdb は逐次プログラムです。
ただし、スレッド並列化は実装してます。

※ ghostmp はスレッド並列化した MPIプログラム です。
実際には、ジョブスクリプト中で実行します。

実習の準備 (1)

実習環境の準備

```
[user1@scls ~]$ cp ~kakuta/lec/ghostmp-k-201311.tar.gz .
[user1@scls ~]$ tar zxvf ghostmp-k-201311.tar.gz
[user1@scls ~]$ cd ghostmp-k-201311
[user1@scls ghostmp-k-201311]$ ls -lF
total 40
drwxr-xr-x 10 user1 group1 4096 Oct 24 10:37 boost_1_46_1/
drwxr-xr-x  3 user1 group1 4096 Nov 21 11:41 database/
drwxr-xr-x  3 user1 group1 4096 Nov 21 11:12 db_kegg/
drwxr-xr-x  2 user1 group1 8192 Nov 26 17:13 ghostmp/
drwxr-xr-x  3 user1 group1 4096 Sep 13 16:53 matrix/
drwxr-xr-x  2 user1 group1 4096 Nov 18 17:49 query/
drwxr-xr-x  3 user1 group1 4096 Nov 26 17:12 run/
drwxr-xr-x  4 user1 group1 4096 Nov 21 15:32 summarize_search_result/
```

- 実習用ファイルをカレントディレクトリにコピー
- 実習用ファイルを解凍展開します
- カレントディレクトリを、ghostmp-k-201311 ディレクトリに変更
- カレントディレクトリの内容を表示
- Boost ライブラリ関係のディレクトリ
- データベースを格納しているディレクトリ
- GHOST-MP プログラムを格納しているディレクトリ
- スコアマトリックスを格納しているディレクトリ
- クエリファイルを格納しているディレクトリ
- ジョブスクリプトを格納しているディレクトリ
- 統計解析プログラムを格納しているディレクトリ

実習の準備 (2)

実習で使用するジョブタイプ

処理	ジョブタイプ
データベースの作成 (ghostmp_mkdb)	バッチジョブ
ホモロジー検索 (ghostmp)	バッチジョブ
統計解析 (kegg_analyzer)	python スクリプトのコマンド実行 （login ノード）

コンパイル

Boost C++ライブラリ

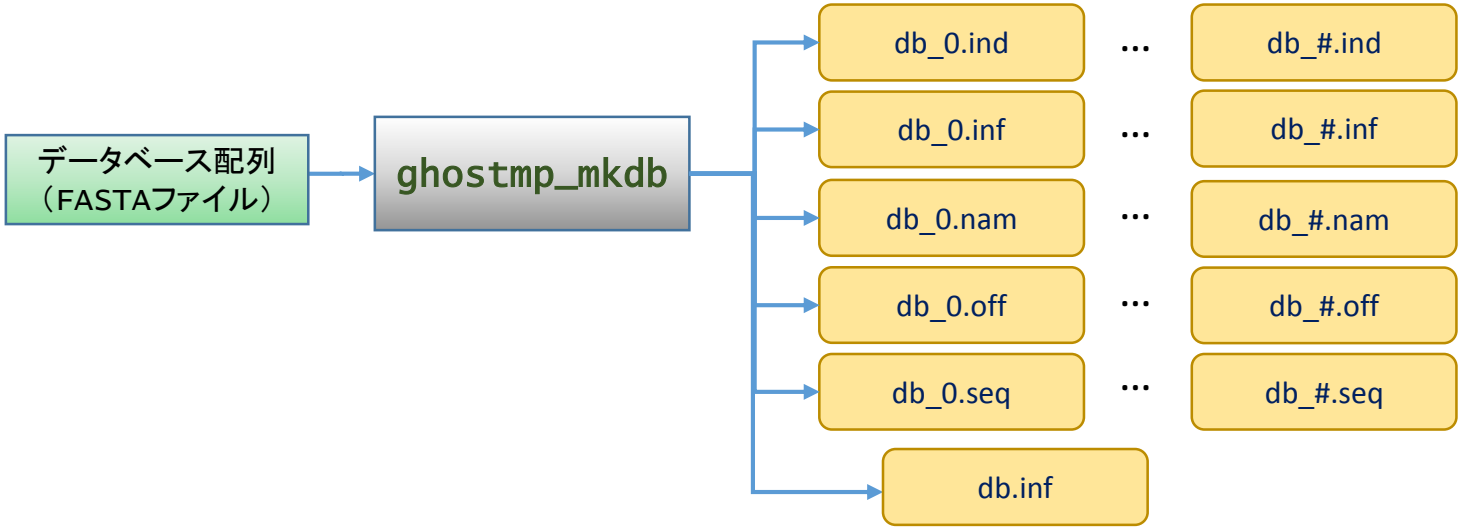
本来は、「京」用の patch を当ててコンパイルが必要だが、時間がかかるので省略。
すでにコンパイルされている添付の boost_1_46_1 を使用する。
GHOST-MP のコンパイル時にリンクされる。

ghostmp_mkdb　　ghostmp

```
$ cd ghostmp
$ make
    ⋮

$ ls -la ghostmp_mkdb
-rwxr-xr-x 1 user1 group1  8041692 Nov 26 17:00 ghostmp_mkdb
$ ls -la ghostmp
-rwxr-xr-x 1 user1 group1 13496215 Nov 26 17:02 ghostmp
```

コンパイルに成功すると
ghostmp_mkdb　と
ghostmp が作成される

データベースの作成 [ghostmp_mkdb]



ホモロジー検索の参照元となる FASTA フォーマットの DNA または アミノ酸配列 の
データベースファイルを、GHOST-MP のデータベースファイルに変換することが必要

FASTA フォーマットとは

```
>gi|66816243|ref|XP_642131.1| hypothetical protein
MASTQNIVEEVQKMLDTYDTNKDGEITKAEAVEYFKGKKAFNPERSAIYLFQVYDKDNDGKITIKELAGDIDFDKALKEY
KEKQAKSKQQEAEVEEDIEAFILRHNKDDNTDITKDELIQGFKETGAKDPEKSANFILTEMDTNKDGTITVKELRVYYQK
VQKLLNPDQ
```

“>”で始まるヘッダ行　配列の説明　1行で書く

配列データ　アミノ酸を一文字表記で表している　改行 OK

データベースファイルはサイズが大きい（数GB～数十GB)ので #個 のチャンクに分割して扱う。
それぞれのチャンクについて 5つのファイル (.ind .inf .nam .off .seq) を生成します。
チャンク分割サイズは、生成時に引数で指定できます。

データベースの作成 [ghostmp_mkdb]

ジョブスクリプト （逐次ジョブ）

```
[user1@scls ghostmp-k-201311]$ less database/run.sh
#!/bin/sh
#------ pjsub options ------#
#PJM -L "rscgrp=small"
#PJM -L "node=1"
#PJM -L "elapse=00:10:00"
#PJM -j
#------ Program Execution --------#

export OMP_NUM_THREADS=16

GHOSTX="../ghostmp/ghostmp_mkdb"
INPUT="./genes.pep.201106.107283"
DBDIR="chunkdb"
DB="${DBDIR}/db"
ARGS="-i ${INPUT} -o ${DB} -l 33554432"

$GHOSTX $ARGS
```

リソースグループ「small」
使用ノード数「1」
最大経過時間「10分間」
標準エラー出力を標準出力に向ける

スレッド並列数の設定 （「京」の場合は8）

ghostmp_mkdb プログラムのパス

データベース配列（KEGG GENES アミノ酸配列の一部）

オプション指定： チャンク分割サイズ 32MB に設定

ghostmp_mkdb の実行

※ これは、実習用の小さなデータベースです。
一般的には、実際の実行には、数十分～数時間かかります。

データベースの作成 [ghostmp_mkdb]

ジョブの投入 と 状態確認

```
[user1@scls ghostmp-k-201311]$ cd database
[user1@scls database]$ pjsub run.sh
[INFO] PJM 0000 pjsub Job 10569 submitted.
[user1@scls database]$ pjstat

  ACCEPT QUEUED  STGIN  READY RUNING RUNOUT STGOUT   HOLD  ERROR   TOTAL
       0      0      0      0      1      0      0      0      0       1
s      0      0      0      0      1      0      0      0      0       1

JOB_ID     JOB_NAME   MD ST  USER     START_DATE      ELAPSE_LIM NODE_REQUIRE
10569      run.sh     NM RUN user1    10/24 14:49:45  0000:10:00 1
```

ジョブ実行結果

```
[user1@scls database]$ less run.sh.oXXXXX

[GHOSTX] start ghostx.
The number of chunks :2
Max length of a chunk : 33553615
Total database length : 50369581
Total number of sequences : 107283
```

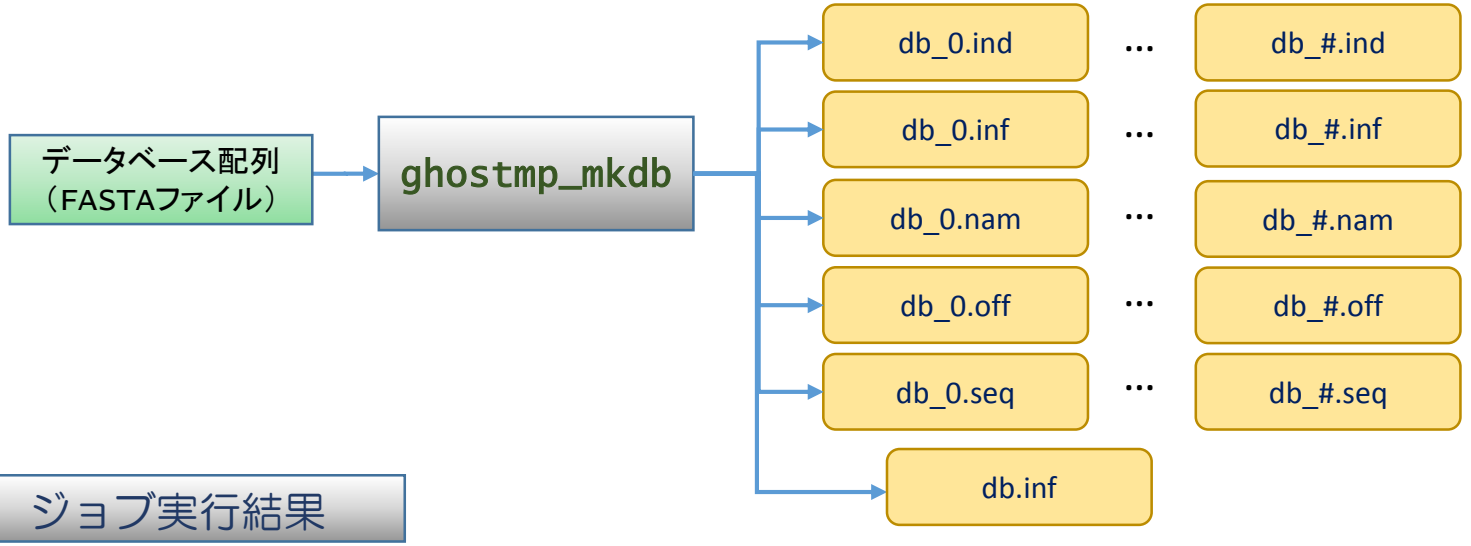
分割チャンク数２

チャンクあたりの最大長

全データベース長

全配列数

データベースの作成 [ghostmp_mkdb]



ジョブ実行結果

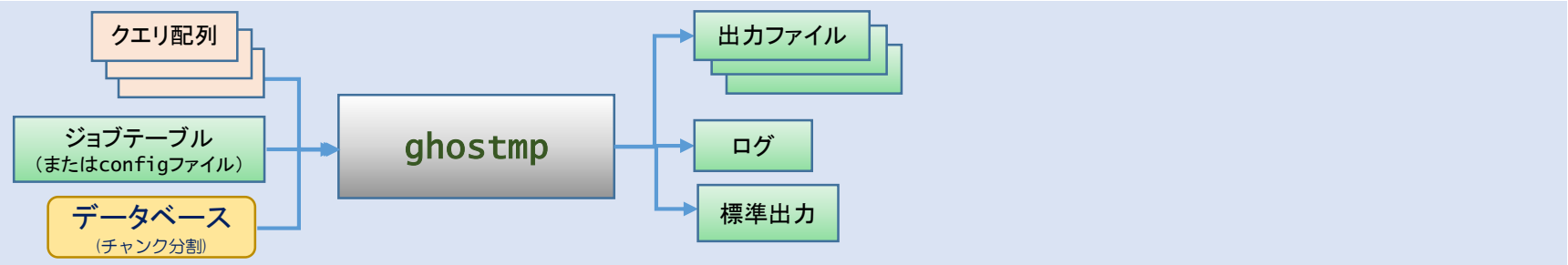
```
[user1@scls database]$ ls -lF chunkdb
total 845888
-rw-r--r-- 1 user1 group1 444862785 Oct 24 14:50 db_0.ind
-rw-r--r-- 1 user1 group1         8 Oct 24 14:49 db_0.inf
-rw-r--r-- 1 user1 group1   5507157 Oct 24 14:49 db_0.nam
-rw-r--r-- 1 user1 group1    293568 Oct 24 14:49 db_0.off
-rw-r--r-- 1 user1 group1  33553615 Oct 24 14:49 db_0.seq
-rw-r--r-- 1 user1 group1 361710965 Oct 24 14:50 db_1.ind
-rw-r--r-- 1 user1 group1         8 Oct 24 14:50 db_1.inf
-rw-r--r-- 1 user1 group1   2960877 Oct 24 14:50 db_1.nam
-rw-r--r-- 1 user1 group1    135564 Oct 24 14:50 db_1.off
-rw-r--r-- 1 user1 group1  16923251 Oct 24 14:50 db_1.seq
-rw-r--r-- 1 user1 group1        24 Oct 24 14:50 db.inf
```

チャンク0　データベース

チャンク1　データベース

それぞれのチャンクについて 5つのファイル (.ind .inf .nam .off .seq) を生成

ホモロジー検索 [ghostmp]



入力ファイル確認 ― クエリ配列

```
[user1@scls ghostmp-k-201311]$ less ./query/q.1
```

クエリファイルの一個

```
>61JGYAAXX100510:7:100:10000:7489/1
```

ヘッダ行

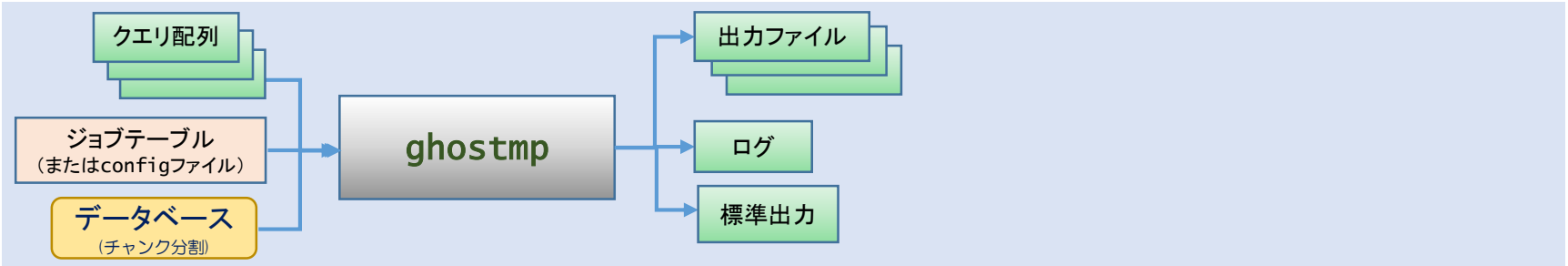
```
AACCCAACATGGCATCCTTGGTCCCTAGAGCAATCTCCTTGCCCTCTTTTTTAGCATAACTGATTAACTGGCGCAAATGTAACTTGG
AAATTGGGCTCGTG
```

配列データ　DNA 塩基配列

```
>61JGYAAXX100510:7:100:17664:14710/1
ACTTGCAATCTGCGATTCTGTTTGCCCAGACTCCGACAAAATTTTCACCTGAGTAAAAGTCCGAAATTGTCCTAGCATGACTGCAAT
CAACTTGATTTCAT
>61JGYAAXX100510:7:100:8182:5401/1
CAACTCTTGCTTTTTCACTTTCAGAATAAGGAACCGGATAAATTTTTCGGGCTCATTGAGGACAATCTAAAGCAGGTTCATCCTCTT
TTTCAGACTGTCTT
>61JGYAAXX100510:7:101:15839:13397/1
TAATCTGTAGAAAGCAAAATTTCAAAAACATAGATATGATACTTTAAGCACCACACCTTATTTCCATCCTGCCTACATATTAATGAT
AAAAATACGGTGTA
   ⋮
```

※ クエリファイルの配列は、次世代シーケンサーで得られた　HMPの頬粘膜のDNA塩基配列データです。

ホモロジー検索 [ghostmp]



入力ファイル確認 ― ジョブテーブル

```
[user1@scls ghostmp-k-201311]$ less run/table

TITLE=ghostmp
PARAM=-i $1 -d $2 -o $3
../query/q.1     ../db_kegg/chunkdb/db    ./out/o.1
../query/q.2     ../db_kegg/chunkdb/db    ./out/o.2
../query/q.3     ../db_kegg/chunkdb/db    ./out/o.3

    ⋮                  ⋮                      ⋮

../query/q.30    ../db_kegg/chunkdb/db    ./out/o.30
```

ジョブのタイトル（任意）

オプション指定

30個の ジョブ を並列分散計算する：

30個の クエリファイル から
30個の 出力ファイル が生成される

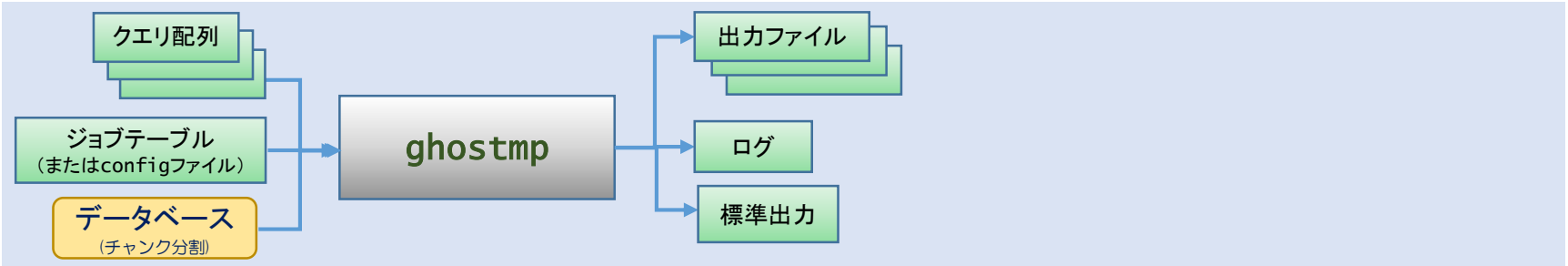
クエリ配列
入力ファイル

データベース

ホモロジー検索結果
出力ファイル

※ データベースは、あらかじめ作成しておいた、KEGG GENES アミノ酸配列の一部です。

ホモロジー検索［ghostmp］



ジョブスクリプト（MPI並列ジョブ）

```
[user1@scls ghostmp-k-201311]$ less run/run.sh
#!/bin/sh
#------ pjsub options ------#
#PJM -L "rscgrp=small"
#PJM -L "node=2"
#PJM --mpi "proc=8"
#PJM -L "elapse=00:10:00"
#PJM -j
#------ Program Execution --------#

export OMP_NUM_THREADS=4

GHOST_MP="../ghostmp/ghostmp"
TABLE="./table"
SCORE_MATRIX="../matrix/blast-2.2.24/data/PAM30"
LOG="./log"

ARGS="-tb $TABLE -lg $LOG -rc 1 -po 1
 -a $OMP_NUM_THREADS -b 5 -T 32 -r -1 -q d -t p -M $SCORE_MATRIX -G 9 -E 1"

mpiexec $GHOST_MP $ARGS
```

リソースグループ「small」
使用ノード数「2」
プロセス数「8」
最大経過時間「10分」
標準エラー出力を標準出力に向ける

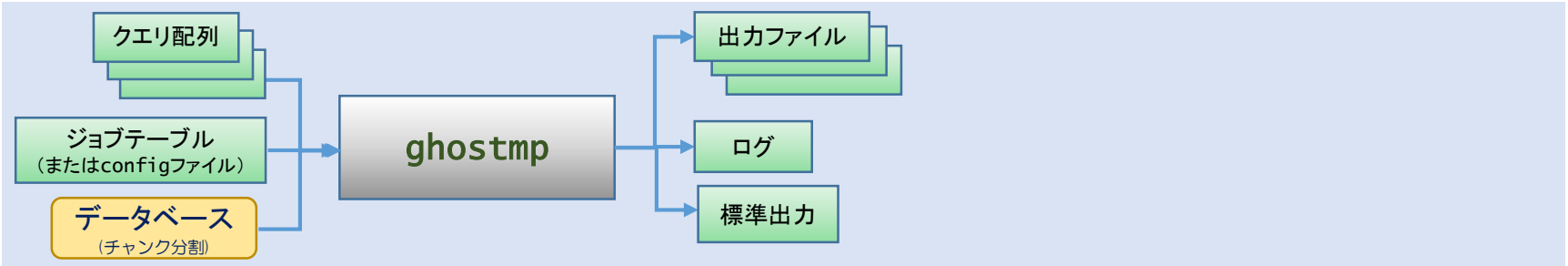
スレッド並列数の設定

ghostmp プログラムのパス

ghostmp
のオプション

ghostmp の実行

ホモロジー検索［ghostmp］



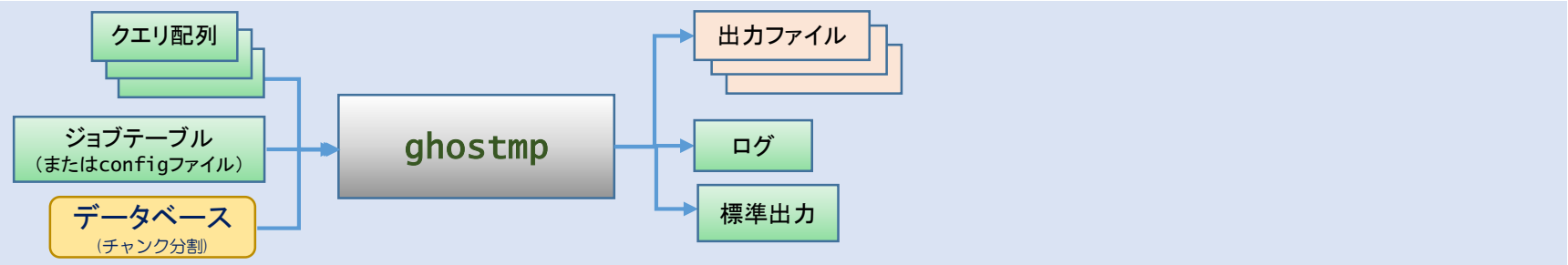
ジョブの投入と状態確認

```
[user1@scls ghostmp-k-201311]$ cd run
[user1@scls run]$ pjsub run.sh

[INFO] PJM 0000 pjsub Job 10583 submitted.
[user1@scls run]$ pjstat

  ACCEPT QUEUED  STGIN  READY RUNING RUNOUT STGOUT   HOLD  ERROR   TOTAL
       0      0      0      0      1      0      0      0      0       1
s      0      0      0      0      1      0      0      0      0       1

JOB_ID     JOB_NAME   MD ST  USER     START_DATE      ELAPSE_LIM NODE_REQUIRE
10583      run.sh     NM RUN user1    10/24 16:23:48  0000:10:00 2
```

ジョブ終了確認

```
[user1@scls run]$ pjstat

  ACCEPT QUEUED  STGIN  READY RUNING RUNOUT STGOUT   HOLD  ERROR   TOTAL
       0      0      0      0      0      0      0      0      0       0
s      0      0      0      0      0      0      0      0      0       0
```

ホモロジー検索［ghostmp］

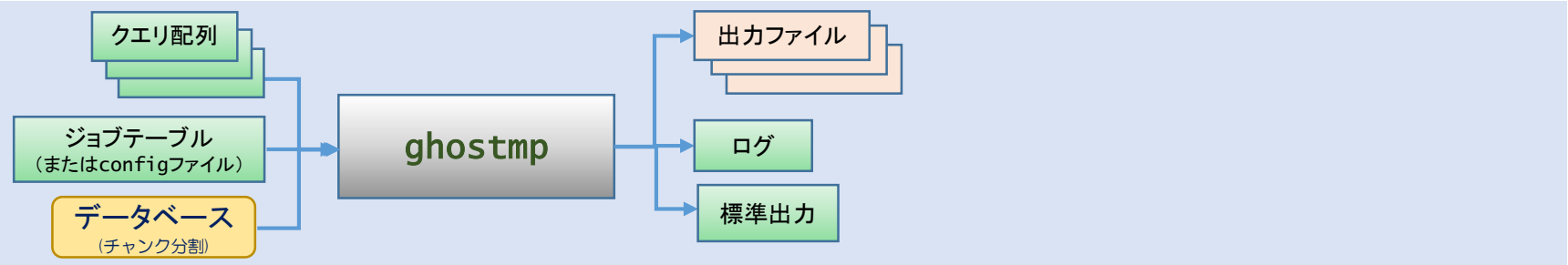


結果確認1 ― 出力ファイル

```
[user1@scls run]$ ls -lF ./out

total 2880
-rw-r--r-- 1 user1 group1 75335 Nov 21 11:49 o.1
-rw-r--r-- 1 user1 group1 77403 Nov 21 11:49 o.10
-rw-r--r-- 1 user1 group1 77107 Nov 21 11:49 o.11

      :


-rw-r--r-- 1 user1 group1 77727 Nov 21 11:49 o.9
```

30個 のジョブの出力ファイル

ホモロジー検索［ghostmp］



結果確認1 ― 出力ファイル

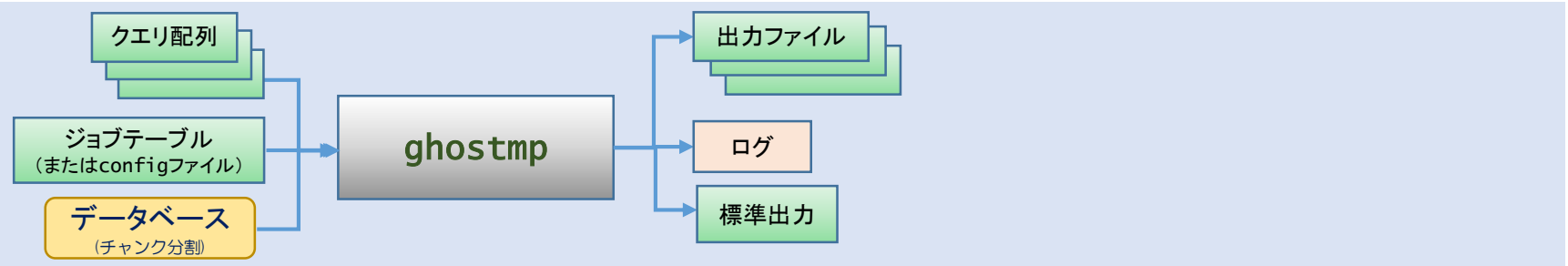
```
[user1@scls run]$ less ./out/o.1

61JGYAAXX100510:7:100:10000:7489/1      sti:Sthe_1623  binding-protein-dependent
transport systems inner membrane component
     0.714286        14      2       1       32      73      105     116     2.88961
29.4819
   ⋮
```

30個 の出力ファイルのうちの１個

※一行毎に、以下の12種の内容が タブ区切りで記述されます。

クエリ配列名、ヒット配列名、一致率、アラインメント長、不一致率、ギャップ数、
クエリ配列におけるアラインメント開始位置、その終了位置、
ヒット配列におけるアラインメント開始位置、その終了位置、
E-value、正規化スコア

ホモロジー検索［ghostmp］



結果確認２ ー ログ

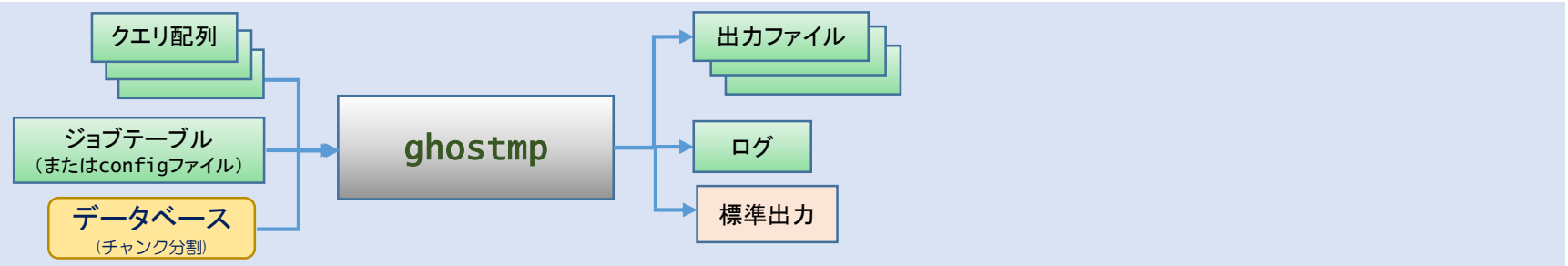
```
[user1@scls run]$ less log
   ⋮


5       0
6       8       00004           1       00030           1       00054           1
00017           1       00039
        1       00045           1       00051           1       00057           1
7       9       00006           1       00026           1       00050           1
00015           1       00039
        1       00037           1       00043           1       00049           1
00055           1

Elapsed time GHOST-MP = 112.911 sec.

Elapsed time POST = 0.479745 sec.

Elapsed time GHOST-MP + POST = 113.391 sec.
```

GHOST-MP の ホモロジー検索の 計算時間

GHOST-MP の ポスト処理の 計算時間

GHOST-MP の 全 計算時間

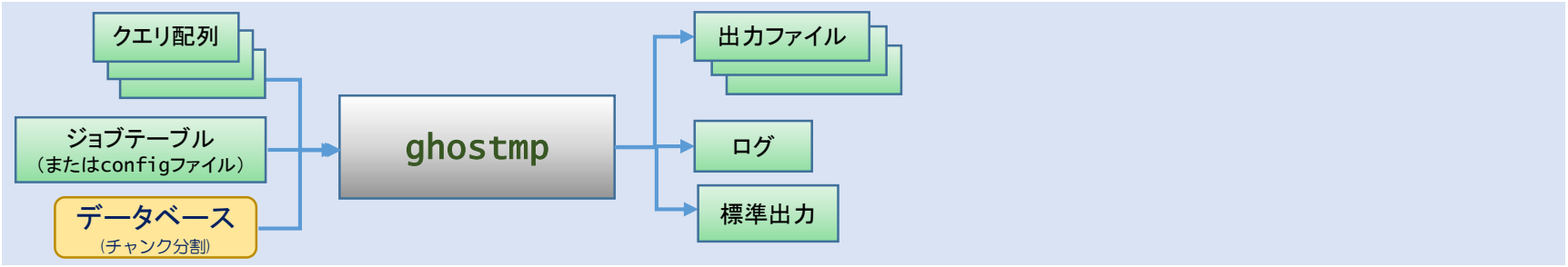
ホモロジー検索 [ghostmp]



結果確認３ ー 標準出力

```
[user1@scls run]$ less run.sh.oXXXXX

Bcasttime= 5.41858 sec myid= 4 localmaster= 1

Bcasttime= 15.6326 sec myid= 5 localmaster= 0
[GHOSTX] start ghostx.
[GHOSTX] ALIGN mode
Presearching ...

Bcasttime= 15.7927 sec myid= 7 localmaster= 0

Bcasttime= 15.7928 sec myid= 6 localmaster= 0
[GHOSTX] start ghostx.
[GHOSTX] ALIGN mode
[GHOSTX] start ghostx.
[GHOSTX] ALIGN mode
Presearching ...
Presearching ...
   ⋮
```

ホモロジー検索［ghostmp］



「京」での実行

SCLS計算システム（今回の実習）	「京」（実用的な計算）
• 2ノード 8プロセス 4スレッド • データベース 739,884本のアミノ酸配列 • クエリ配列 100本×30ファイル =3000本 の DNA配列	• 1296ノード 1296プロセス 8スレッド • データベース 8,578,853本のアミノ酸配列 （約12倍） • クエリ配列 1,000本×2,566ファイル =2,566,000本 の DNA配列（約855倍）
• 計算時間 約2分	• 計算時間 約10分

※「京」では、ステージングが必要で、実行スクリプトに ステージング処理 を書く必要があります。
※「京」では、1296=6×12×18 のように、３次元形状 で MPIジョブ を投入します。

ホモロジー検索［ghostmp］

「京」での実行スクリプト例

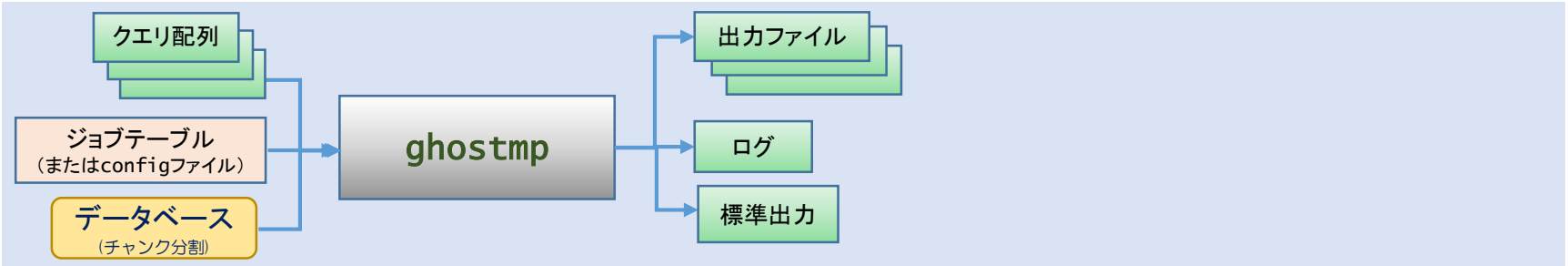
```
#!/bin/sh
#PJM --rsc-list "rscgrp=large"
#PJM --rsc-list "elapse=0:30:00"
#PJM --rsc-list "node=6x12x18:strict"
#PJM --mpi "shape=6x12x18"
#PJM --mpi "proc=1296"
#PJM -S
#PJM --stg-transfiles all
#PJM --mpi "use-rankdir"
#PJM --stgin "rank=* ../ghostmp/ghostmp %r:./a.out"
#PJM --stgin "rank=* table %r:./table"
#PJM --stgin "rank=0 ../query/q.* 0:../"
#PJM --stgin "rank=* ../database/chunkdb/db* %r:./db/"
#PJM --stgin "rank=* ../matrix/blast-2.2.24/data/PAM30 %r:./PAM30"
#PJM --stgout "rank=0 0:./log ./log"
#PJM --stgout "rank=* %r:../o.* ./out/"

. /work/system/Env_base
export OMP_NUM_THREADS=8

NODES=1296
GHOST_MP="./a.out"
TABLE="./table"
SCORE_MATRIX="./PAM30"
LOG="./log"

ARGS="-tb $TABLE -lg $LOG -rc 0 -dx 6 -dy 4 -dz 6 -po 1
 -a $OMP_NUM_THREADS -b 5 -T 32 -r -1 -q d -t p -M $SCORE_MATRIX -G 9 -E 1"

mpiexec -n $NODES -mca coll_tuned_bcast_same_count 1 ¥
 lpgparm -s 4MB -d 4MB -h 4MB -t 4MB -p 4MB $GHOST_MP $ARGS
```

全系は 6×12×18

ステージイン
ステージアウト処理

全系 6×12×18 を
6×4×6 の 9個 の直方体に分割

ホモロジー検索［ghostmp］



「京」での実行 ― ジョブテーブル例

```
TITLE=ghostmp
PARAM=-i $1 -d $2 -o $3
../q.1     ./db/db    ../o.1
../q.2     ./db/db    ../o.2
../q.3     ./db/db    ../o.3
  ⋮          ⋮          ⋮
```

ジョブのタイトル（任意）

オプション指定

ジョブ を並列分散計算する

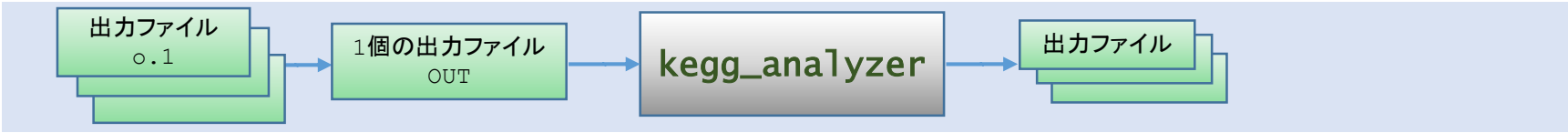
クエリ配列
入力ファイル

データベース

ホモロジー検索結果
出力ファイル

※「京」でのジョブテーブルは、入出力ファイルがステージインされた状態を考慮して記述します。
※「京」では、“．．”は、グローバルファイルシステム、“．”はローカルファイルシステムを表します。
GHOST-MPの一般的な使用法では、クエリ配列入力ファイルと出力ファイルをグローバルファイルシステムに、データベースをローカルファイルシステムに置きます。

結果の統計解析 [kegg_analyzer]



GHOST-MP の出力ファイルを一個にまとめる

```
[user1@scls ghostmp-k-201311]$ cd summarize_search_result

[user1@scls summarize_search_result]$ cat ../run/out/o.* > OUT



[user1@scls summarize_search_result]$ ls -la OUT
-rw-r--r-- 1 user1 group1 2316600 Nov 21 13:26 OUT
```

カレントディレクトリを summarize_search_result にする

../run/out ディレクトリにある 30個 の 出力ファイル o.1, o.2, o.3 ・・・ o.30 を1個の OUT にまとめる

約 2.3 Mバイトの OUT ファイルを確認

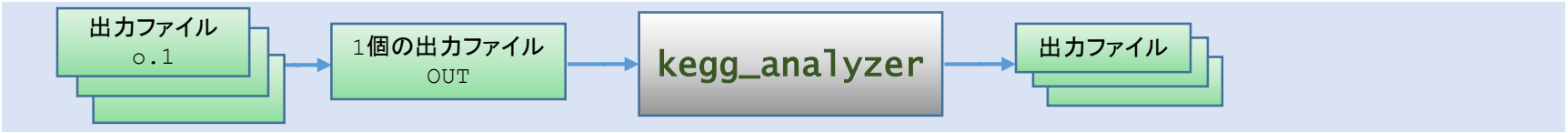
統計解析を実行する

```
[user1@scls summarize_search_result]$ python summarize_search_result.py OUT
```

pythonスクリプトの実行 OUT ファイル を引数にする

※ 実習では、KEGG GENES アミノ酸配列の 約1/10 のサイズを使ってホモロジー検索しています。
（本来は、KEGG GENES アミノ酸配列の フルサイズ を使って計算します。）

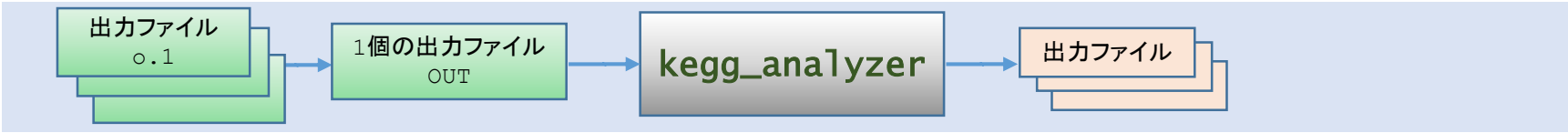
結果の統計解析 [kegg_analyzer]



統計解析を実行する　標準出力

```
[user1@scls summarize_search_result]$ python summarize_search_result.py OUT

2013/11/21 14:44:21     START   Start KEGG Analyzer
2013/11/21 14:44:21     START   Loading Gi-Taxid map file...
2013/11/21 14:45:27     END     54380376 genes loaded.
2013/11/21 14:45:27     START   Loading Taxonomy root map file...
2013/11/21 14:45:33     END     919194 species loaded.
2013/11/21 14:45:33     START   Loading KO-Enzyme map file...
2013/11/21 14:45:33     END     4808 KOs loaded.
2013/11/21 14:45:33     START   Loading USCG map file...
2013/11/21 14:45:33     END     36 USCGs loaded.
2013/11/21 14:45:33     START   Loading KEGG genes file...
2013/11/21 14:46:22     END     8782317 genes loaded.
2013/11/21 14:46:24     START   Loading Blast result...
2013/11/21 14:46:25     END     15000 blast results loaded.
2013/11/21 14:46:25     START   Normalizing...
2013/11/21 14:46:25     START   1. count genes...
2013/11/21 14:46:26     END     done.
2013/11/21 14:46:26     START   2. count USCGs...
2013/11/21 14:46:26     END     done.
2013/11/21 14:46:26     START   3. normalizing...
2013/11/21 14:46:28     END     done.
2013/11/21 14:46:28     END     Normalize.
2013/11/21 14:46:28     END     End KEGG Analyzer
```

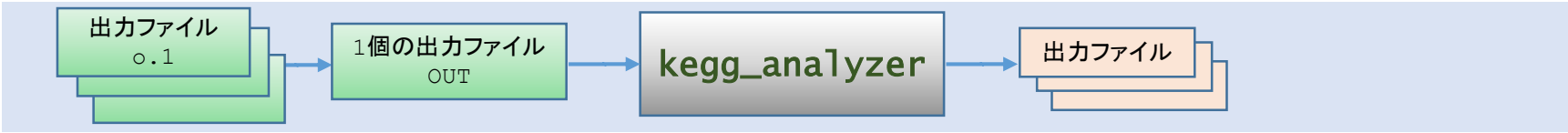
結果の統計解析 [kegg_analyzer]



出力ファイル（主なもの）

ファイル名	内容
genes_freq	遺伝子 の出現頻度
otu_freq	OTU の出現頻度
ec_ratio	各 EC number を有する 遺伝子 の割合
ko_ratio	各 KO（KEGG Orthology）を有する 遺伝子の割合
phylum_ratio	phylum の割合
genus_ratio	genus の割合

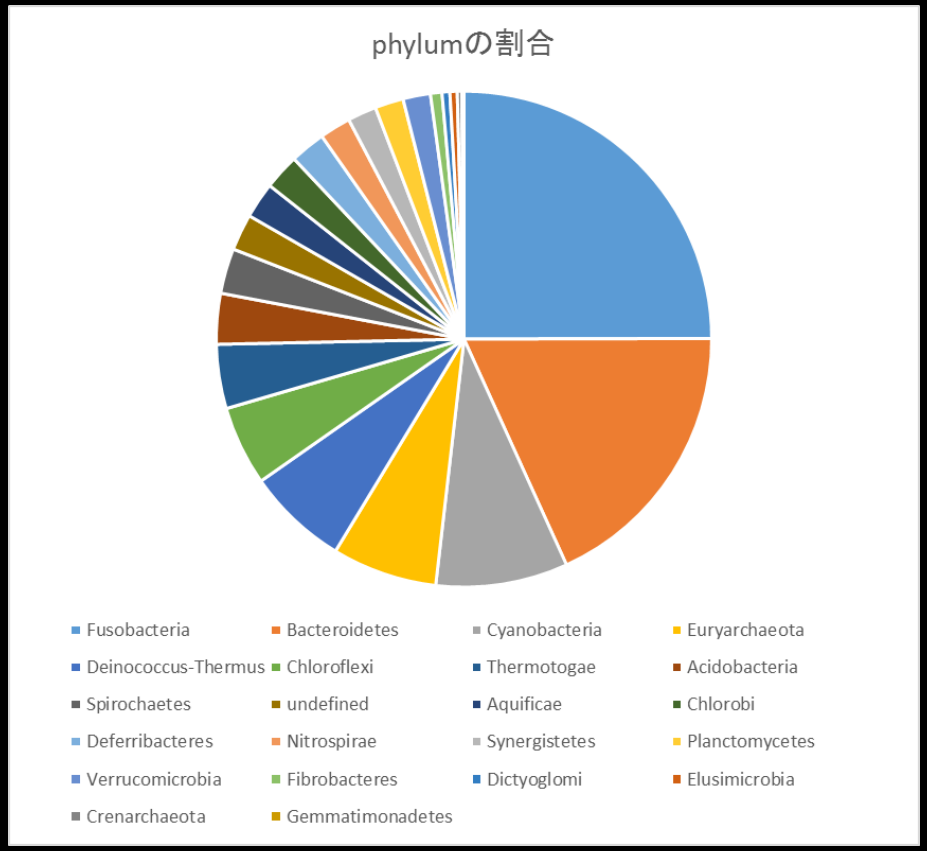
結果の統計解析 [kegg_analyzer]



phylum_ratio ファイルの確認

```
[user1@scls ghostmp-k-201311]$ less phylum_ratio

Verrucomicrobia 0.017921
Elusimicrobia   0.004741
Chlorobi        0.023247
Deferribacteres 0.023058
Dictyoglomi     0.005112
Aquificae       0.023399
Deinococcus-Thermus     0.066145
Gemmatimonadetes        0.001370
Acidobacteria   0.033031
Spirochaetes    0.029487
Nitrospirae     0.020319
Chloroflexi     0.051829
Planctomycetes  0.018602
Fusobacteria    0.249573
Synergistetes   0.018786
Crenarchaeota   0.002963
undefined       0.023746
Bacteroidetes   0.182508
Cyanobacteria   0.086210
Thermotogae     0.041861
Fibrobacteres   0.007519
Euryarchaeota   0.068572
```

エクセル で 円グラフ を書くと以下のとおりです



GHOST-MPについて

監修・製作：東京工業大学 秋山研究室 GHOST-MPチーム
秋山 泰、石田 貴士、角田 将典、鈴木 脩司

資料製作　：（株）情報数理バイオ

ghost-mp@bi.cs.titech.ac.jp

2013年11月

独立行政法人理化学研究所
HPCI計算生命科学推進プログラム