Genomon-fusion for K

ユーザマニュアル ver.1.0

東京大学医科学研究所　ヒトゲノム解析センター　DNA 情報解析分野

目次

更新履歴

2015/07/28　　ユーザマニュアル公開

1. はじめに

Genomon-fusion for K（以下，GFK）は Genomon-fusion[1][2]の京コンピュータ向け移植・拡張版である. MPI/OpenMP によるプロセス並列/スレッド並列を実装することにより，多数検体の同時解析を効率よく実施できるようになっている. 当然のことながら，京コンピュータ以外の一般的なスーパーコンピュータにおいても動作するよう設計されている．一方，京/スーパーコンピュータとの親和性が低い部分については，オリジナルの Genomon-fusion から変更が加えられている．利用にあたっては，以下に列挙する主な変更点を考慮されたい.

主な変更点：

- グリッドエンジン（SGE/UGE）の代わりに MPI/OpenMP による並列処理
- ソートおよび PCR duplicate 除去に Picard を使わず samtools を使用
- SoftClipping リードからの juncContig 生成に，CAP3 ではなく SOAPdenovo-Trans[3]を使用

また，samtools に関しては京上での実行ができるよう，一部ソースコードの修正を行っている．blat[4]**エラー! 参照元が見つかりません。**に関しては非常に計算時間が掛かるため，OpenMP によるスレッド並列化を行い，そのパッチを公開している[5]．詳細に関してはインストールの章で説明する.

GFK および blat の OpenMP 化パッチに関しては，HPCI 戦略プログラム分野 1「予測する生命科学・医療および創薬基盤」課題 4「大規模生命データ解析」の研究成果の一部である.

2015 年 7 月 28 日
東京大学　医科学研究所 ヒトゲノム解析センター
DNA 情報解析分野
伊東　聰
sito@hgc.jp

2. インストール

2.1 必要条件

下記のソフトウェア/ライブラリが稼働中/インストールできる並列計算機.

- Message Passing Interface（MPI）
- perl
- blat ver.35
- Bowtie ver.0.12.7
- samtools ver.0.1.20
- fasta36 ver.36.3.5c
- SOAP denovo-Trans ver.1.0.4
- bedtools ver.2.14.3

また，データセットのダウンロードを Makefile で自動実行する場合は，インターネットへの接続が必要である.

2.2 インストール方法

tar.gz アーカイブを展開し，生成されたディレクトリに移動する．以降，このディレクトリを$TOP と表記する.

2.2.1 blat およびデータセットのライセンス準備

GFK ではアラインメントおよび fusion gene 検出処理中で blat を使用する．**blat は再配布および商用利用に許諾が必要**である．よって，ユーザ自身で blat を準備する必要がある．**非商用または学術利用に関してはライセンスフリーで利用可能**である．詳細に関しては開発者のウェブサイト[4]を参照されたい.

blat はソースコードからビルドするか，上記サイトから利用環境用実行ファイルをダウンロードする．得られた実行ファイル群を，$TOP/bin ディレクトリにコピーする.

bowtie および blat の使用するリファレンスファイルは UCSC が公開しているデータセットから生成する．UCSC のデータセットの利用には別途ライセンスが必要である．**商用利用には UCSC の許諾が必要**である．**非商用または学術利用に関してはライセンスフリーで利用可能**である．詳細については UCSC のウェブサイト[6]を参照されたい.

2.2.2 環境設定

コンパイル等に必要な設定はすべて Makefile.in の中で行う．ユーザが設定すべき項目は以下の 5 点である.

Tab. 1 Compile parameters of GFK

CC	MPI C コンパイラの指定.
COPTIONS	OpenMP 化 blat を利用する場合は-DBLAT_OPENMP を指定する.
OPTFLAGS	コンパイラの最適化レベルおよび OpenMP 有効化の指定.
INSTDIR	GFK のインストールディレクトリ.
SHAREDDIR	共有ディレクトリの設定（京コンピュータ用）.

SHAREDDIR は京コンピュータの採用している特殊なファイルシステムに対応するための設定である．bowtie や blat のリファレンスなど，複数プロセスで参照するファイルを置くディレクトリが別途用意されている場合にそのパスを指定する．京コンピュータの場合はカレントディレクトリの一つ上（親ディレクトリ）を指定する．一般のシステムでは，カレントディレクトリを設定しておけばよい.

2.2.3 ビルド

GFK のビルドはいくつかのステップに分かれている．最も単純な方法は$TOP ディレクトリで make するだけである.

```
> make
```

ツールのダウンロード，ビルド，データセットのダウンロード，リファレンスファイルの生成，GFK ソースコードのビルドが自動的に実行され，$TOP/bin ディレクトリにパッケージに必要なすべてのプログラムおよびファイルが格納される．INSTDIR にインストールする場合は，この後に make install する.

ビルドの各ステップを個別に実行する場合は，対応する引数を make に与えて実行すればよい．主な引数は以下の通りである.

Tab. 2 Options for make

tool	samtools, fasta36 等のソフトウェアをビルドする.
ref	データセットのダウンロードおよびリファレンスファイルの生成を行う.

src	GFK のソースをビルドする
download	ビルドの各ステップで発生するインターネットからのダウンロードのみを実施する．このステップをあらかじめネット接続されたマシン上で実施しておくことで，ネット接続が確保できない環境でのビルドを行うことが可能になる．

3. 解析実行

GFK はアラインメント（GFKalign），PCR 重複除去（GFKdedup），融合遺伝子検出（GFKdetect）の 3 つのパートで構成されている．まず初めにテストデータを用いて GFK による融合遺伝子検出までの簡単な流れを説明する．各パートについての詳細は，あとの節で説明する．

3.1 テストデータ準備

ビルドが成功したことを確認するため，テストデータを用いてテストランを行う．テストデータは Cancer Cell Line Encyclopedia (CCLE) [7] の RNA-seq データ（CCLE_COLO_783_RNA_08）1 サンプル分である．テストデータのアーカイブを以下からダウンロードする．

http://www.hgc.jp/~sito/GFK/test-1.0.tar.gz

ダウンロードしたアーカイブを$TOP/に置き，展開する．

```
> tar xvzf test-1.0.tar.gz
```

3.2 テスト実行

展開したディレクトリに移動し，テスト実行を行う．実行方法は以下の通りである：

```
> cd test
> cd Input; ../bin/GFKpre.sh ../input.txt
> mpirun -np 624 ../bin/GFKalign Output
> cd Output; ../../bin/GFKpost.sh ../Input/GFKPostProcess.txt
> cd ..; mpirun -np 1 ../bin/GFKdedup Output
> mpirun -np 44 ../bin/GFKdetect Output
```

各コマンドの解説は以下の通りである．

1. テスト用ディレクトリ$TOP/test に移動．
2. 並列アラインメントのために，Fastq ファイルをスプリットする．スプリットしたファイルは Input ディレクトリに格納する．スプリットの調整は input.dat ファイルで行っている．デフォルトのセッティングでは 624 分割（Read1 と Read2 の合計で 1048 ファイル）になる．
3. GFKalign による並列スプリットを行う．2 での分割数を並列数に指定すること．
4. アラインメント結果の SAM ファイルを検体ごとにマージする．

5. マージされた SAM ファイルに，ソート，PCR 重複除去，インデックス作成を行い，BAM 圧縮を行う．
6. PCR 重複除去された BAM ファイルから融合遺伝子の検出を行う．この際の並列数は，サンプル数*44 である．

GFKdetect まで正常に終了すると，fusion.0.txt ファイルが生成されるはずである．fusion.0.ref と同じ内容になっていれば正しく動作したことが確認できる．

3.3 GFKalign

アラインメントパートのフローを Fig. 1 に示す．本章では GFKpre.sh による Fastq ファイルのスプリット，および GFKalign による並列アラインメントの方法について説明する．

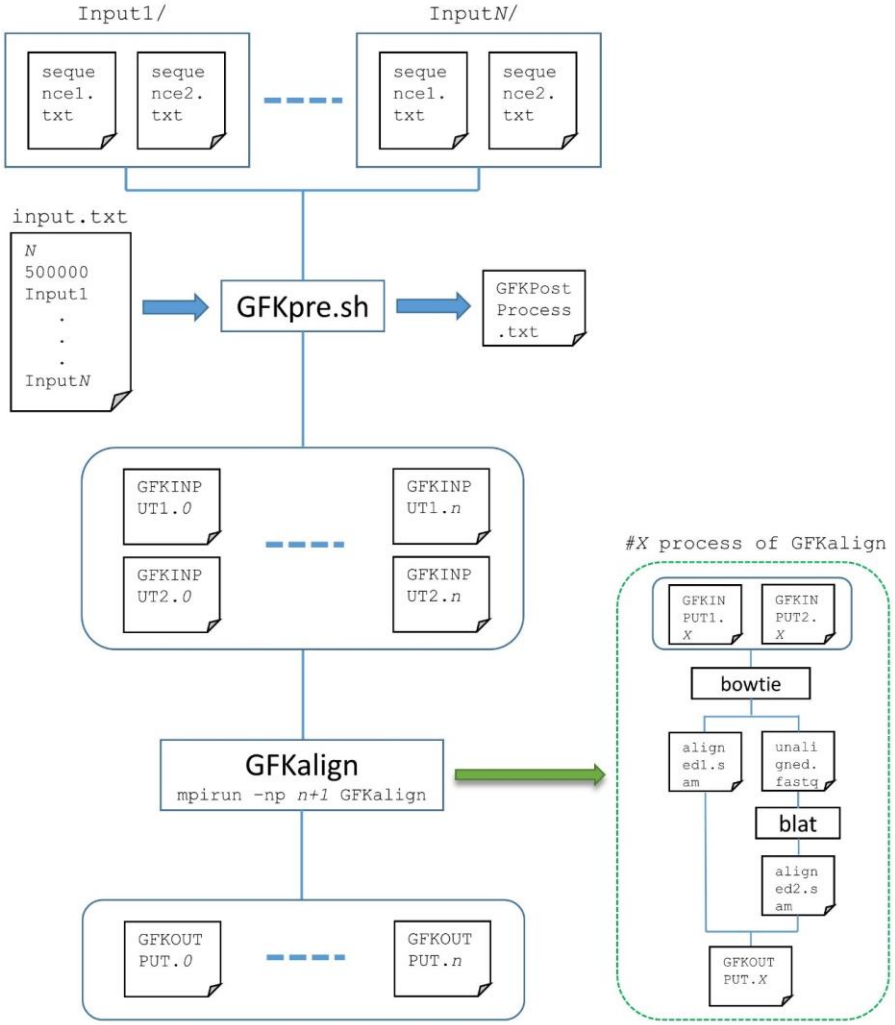

Fig. 1 Flowchart of GFKalign

3.3.1 GFKpre.sh

インプットデータは paired-end の Fastq ファイルである．Read1 および Read2 のファイルをそれぞれ sequence1.txt，sequence2.txt（固定）にする．Fastq ファイル名は固定だが，ディレクトリ名は自由につけてよい．複数サンプルを同時に処理する場合，1 サンプル（正確には，1Fastq ペア）ごとに個別のディレクトリを用意すること．次に，サンプルをスプリットするためのパラメータファイル（Fig. 1 中の input.dat）を準備する．パラメータファイルの書式は以下の通りである．

```
Number_of_samples
Number_of_lines_for_split
Input_dir_path_1
Input_dir_path_2
   .
   .
   .
```

Fig. 2 Parameter file format for GFKpre.sh

1 行目はサンプル数，2 行目は Fastq ファイルを分割する行数である．Fastq ファイルは 4 行で 1 リードになっているため，4 の倍数で指定すること．3 行目以降は Fastq ファイルを格納したディレクトリのパスを列挙する．検体数が多い場合のために，パラメータファイル作成を補助する付属ツール GFKmakelist.sh が準備されている．

Fastq を格納したディレクトリは何処かのディレクトリの下に纏めておき，GFKmakelist.sh にそのディレクトリのパスを引数として渡せば, そのディレクトリ直下にあるディレクトリの絶対パスのリストを返してくれる．例えば，$HOME/Fastq ディレクトリ以下に Input1，Input2，…とサンプルが格納されていた場合，以下のように実行する．

```
> $TOP/bin/GFKmakelist.sh $HOME/Fastq
```

```
$HOME/Fastq/Input1
$HOME/Fastq/Input2
   .
   .
   .
```

Fig. 3 Usage example of GFKmakelist.sh (upper) and its output

結果は標準出力に出力されるので，リダイレクトで適当なファイルに出力するなどしてパラメータファイル作成に利用する．パラメータファイルが準備出来たら，分割したファイルを格納するディレクトリ（Input とする）に移動し，GFKpre.sh で Fastq ファイルを分割する．

```
> cd Input
> $TOP/bin/GFKpre.sh ../input.txt 0 N-1
> $TOP/bin/GFKpre.sh -f ../input.txt
```

GFKpre.sh の使い方は以下の通りである．

Usage	`GFKpre.sh [-f] dir_list_file [SID] [EID]`
dir_list_file	パラメータファイル名
SID / EID	開始/終了　サンプル番号．番号はゼロ始まりの連番でカウントする． 指定がない場合，SID はゼロ，EID は(サンプル数-1)になる．
-f	終了処理．全サンプルを通した連番のファイル名に修正する．

3.3.2 GFKalign

Bowtie および blat を用いたアラインメントを行う．インプットファイルは前項で作成された，分割された Fastq ファイル群（GFKINPUT1.*および GFKINPUT2.*）である．MPI による並列計算ですべてのアラインメントを一度に実施する．並列数はインプットファイルの数の半分（GFKINPUT1.*の個数）である．ファイル数は GFKpre.sh の終了処理時に生成された GFKPostProcess.txt を見ることでも確認できる（最終行の数値）．GFKalign の使い方は以下の通りである．

Usage	`mpirun -np` *n* `GFKalign [-i input_dir] [-o output_dir]`
n	並列数．`GFKPostProcess.txt` の最終行の数値．
-i	GFKINPUT1/2.*ファイルの格納されているディレクトリ．デフォルトはカレントディレクトリ．
-o	アウトプットファイルの出力先ディレクトリ．デフォルトはカレントディレクトリ．

mpirun は MPI（Message Passing Interface）のプログラム並列実行用コマンドである．Usage は MPI におけるジョブ投入方法の一般的な書式であり，ジョブスケジューラの有無，

種類, 運用方法等によってマシンごとに異なる可能性がある. 利用する環境に合わせた MPI プログラムのジョブ投入方法について，管理人に確認してもらいたい.

3.3.3 結果

アラインメントが正常に終了した場合, アウトプットディレクトリ（指定なしの場合はカレントディレクトリ）に n 個の sam ファイル（GFKOUTPUT.*）が生成されているはずである．また，標準エラーに各プロセスの計算時間が出力（GFK_message）される．計算時間が投入プロセス数分出力され，その時間に極端な差がないことを確認しておく.

結果のチェックは次のステップでファイルマージ後に再度確認する.

3.4 GFKdedup

本節では，GFKalign がアラインメントした分割 sam ファイルのマージ，ソート，および PCR 重複除去を実施し，次の融合遺伝子検出パイプラインのインプットとなる BAM ファイルを生成する．GFK のアラインメント部分だけの利用を考える場合，本ステップ終了後の BAM ファイルを用いると良い． Fig. 4 にフローチャートを示す．

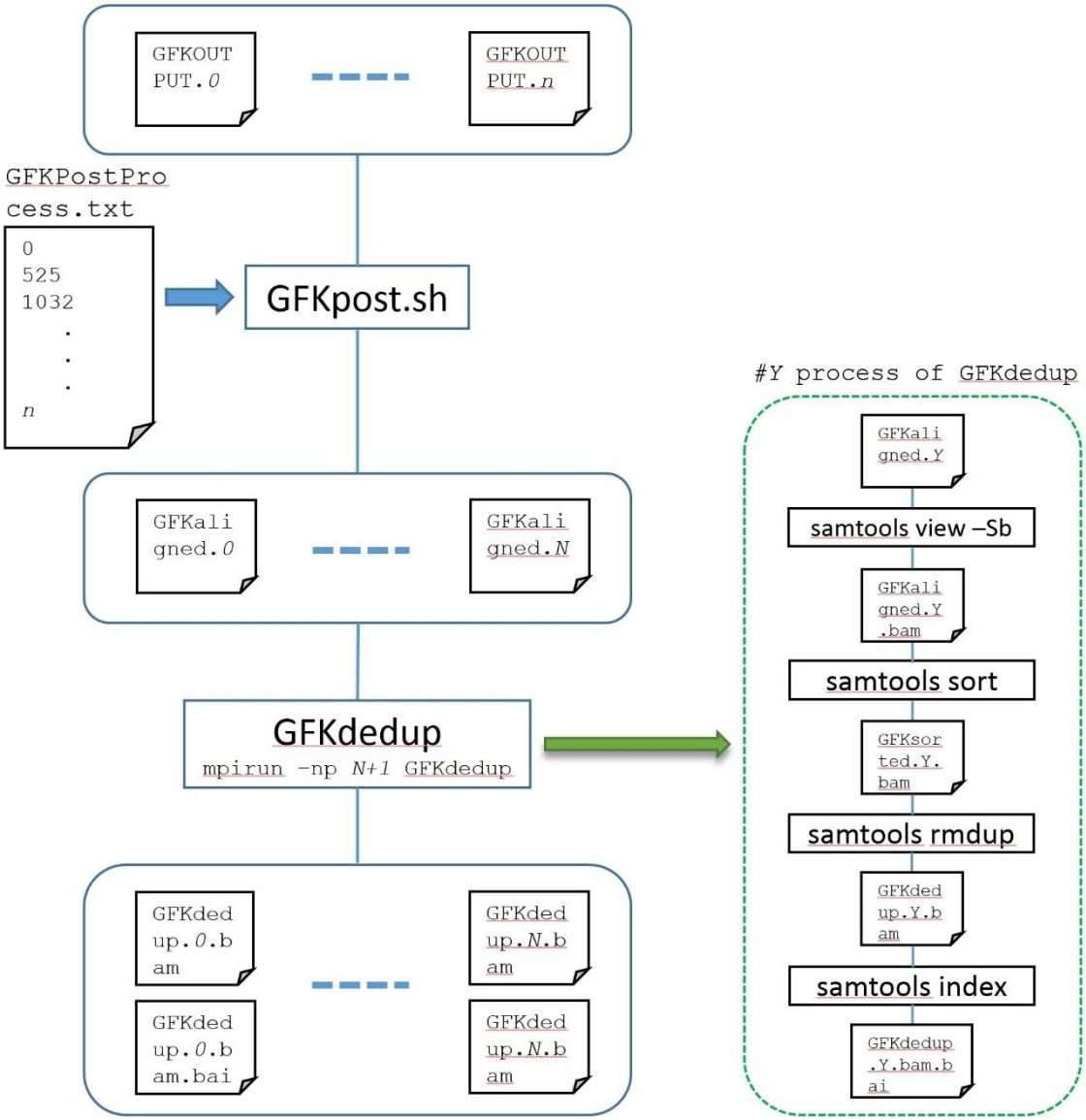

Fig. 4 Flowchart of GFKdedup

3.4.1 GFKpost.sh

複数サンプルの分割されたアラインメント結果である GFKOUTPUT.*を検体ごとにマージする．各サンプルの番号の範囲は，GFKpre.sh でのスプリットの際に自動的に生成される GFKPostProcess.txt に記録されている．GFKPostProcess.txt には検体数 *N+1* の番号（GPP_i）が記録されている．*Y*番目の検体の番号の範囲は GPP_Y から GPP_{Y+1}-1 である．

このファイルを用いて，GFKpost.sh は結果ファイルを検体ごとにマージする．書式は以下の通りである．実行は結果ファイルの格納されているディレクトリで行う．

```
> $TOP/bin/GFKpost.sh GFKPostProcess.txt
```

マージされた SAM ファイルは GFKaligned.*として出力される．マージされた SAM ファイルに関して，対応する Fastq ファイルとの行数による同一性検証を行うことが可能である．検証をする場合は GFKcheck.sh を同じディレクトリでイカのように行う．

```
> $TOP/bin/GFKcheck.sh ../input.txt
```

input.txt は GFKpre.sh の実行時に用いたパラメータファイルである．検証結果は同じディレクトリに check.txt ファイルとして出力される．”Faild sample”の項目がゼロであれば検証をクリアしたことになる．それ以外の場合，行数の一致しないサンプルの一覧が表示される．

3.4.2 GFKdedup

GFKaligned.*に対し，ソート・PCR 重複除去を行った BAM ファイルおよびインデックスファイルの作成を行う．GFKdedup の使い方は以下の通りである．

Usage	`mpirun -np N GFKdedup [-i input_dir] [-o output_dir]`
N	並列数．サンプル数を指定する．
-i	GFKaligned.*ファイルの格納されているディレクトリ．デフォルトはカレントディレクトリ．
-o	アウトプットファイルの出力先ディレクトリ．デフォルトはカレントディレクトリ．

3.3.3 結果

正常に終了した場合，GFKdedup.*.bam と GFKdedup.*.bam.bai ファイルの 2 種類のファイルが *N*個ずつ生成される．

3.5 GFKdetect

本節では，PCR 重複除去の終わった BAM ファイルおよびそのインデックスファイルを用いて，融合遺伝子の検出を行う．Fig. 5 に解析のフローチャートを示す．

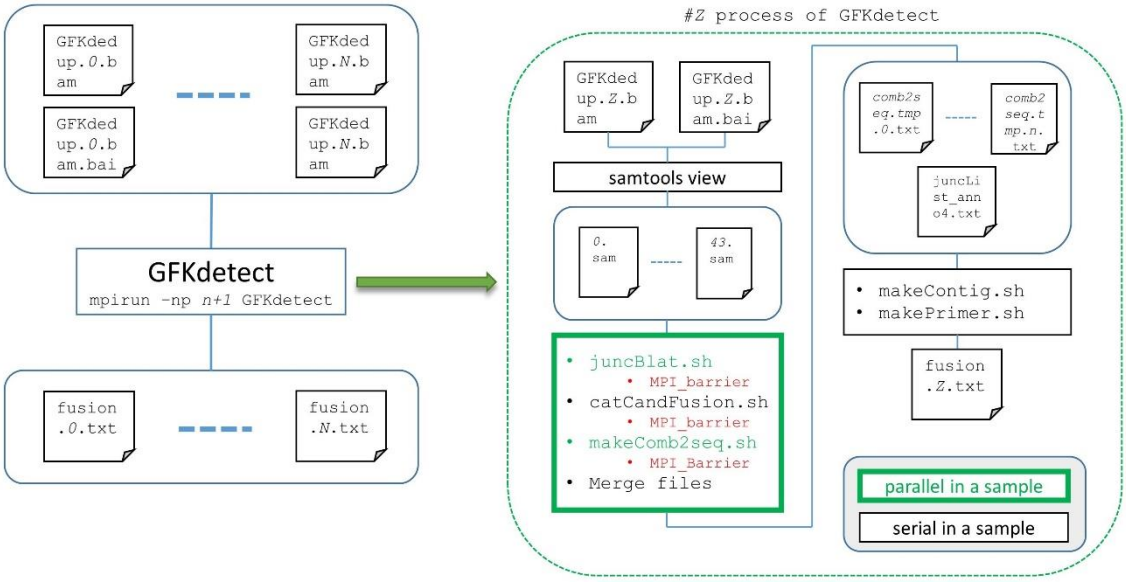

Fig. 5 Flowchart of GFKdetect

インプットファイルは 3.3 節で出力された GFKdedup.*.bam および GFKdedup.*.bam.bai のペアである．Fig. 5 右側の緑のボックスの中が GFKdetect の実施している処理の概要である．GFKdetect では処理の高速化のため，1 サンプルあたり 44 プロセスを使った並列処理を行っている．具体的には Fig. 5 中の”parallel in a sample”で示される juncBlat.sh 等の処理である．

よって，GFKdetect の実行にあたっては，投入する並列数はサンプル数の 44 倍の値を指定する必要があることに注意する．また，本処理中でも blat が一部用いられており，OpenMP によるスレッド並列の併用が可能である．MPI/OpenMP 併用のハイブリッド並列計算を行う際には，アラインメント時と同様に 1 プロセスあたりの並列数およびメモリ使用量に関して留意すること．

計算が終了すると，fusion.*.txt ファイルが生成される．ファイル内容の説明に関しては Genomon-fusion の WEB[1]を参照されたい．

謝辞

本ソフトウェアの一部は、文部科学省「特定先端大型研究施設運営費等補助金（次世代超高速電子計算機システムの開発・整備等 ）」で実施された内容に基づくものである。
テストデータは the Cancer Cell Line Encyclopedia (CCLE) [7] の公開データを利用している.

参考文献

[1] http://genomon.hgc.jp/rna/
[2] Y. Sato et. al., Integrated molecular analysis of clear-cell renal cell carcinoma, Nature Genetics 45, pp.860–867 (2013), doi:10.1038/ng.2699
[3] http://soap.genomics.org.cn/SOAPdenovo-Trans.html
[4] http://www.cse.ucsc.edu/~kent/
[5] http://www.scls.riken.jp/scruise/software/blat_open_MPpatch.html
[6] http://genome.ucsc.edu/conditions.html
[7] http://www.broadinstitute.org/ccle/home