NIBB ゲノムインフォマティックス・トレーニングコース2024
RNA-seq入門
2025.3.5-2025.3.6

RNA-Seqパイプライン
ゲノムベースの解析法

基礎生物学研究所
超階層生物学センター
トランスオミクス解析室
山口勝司

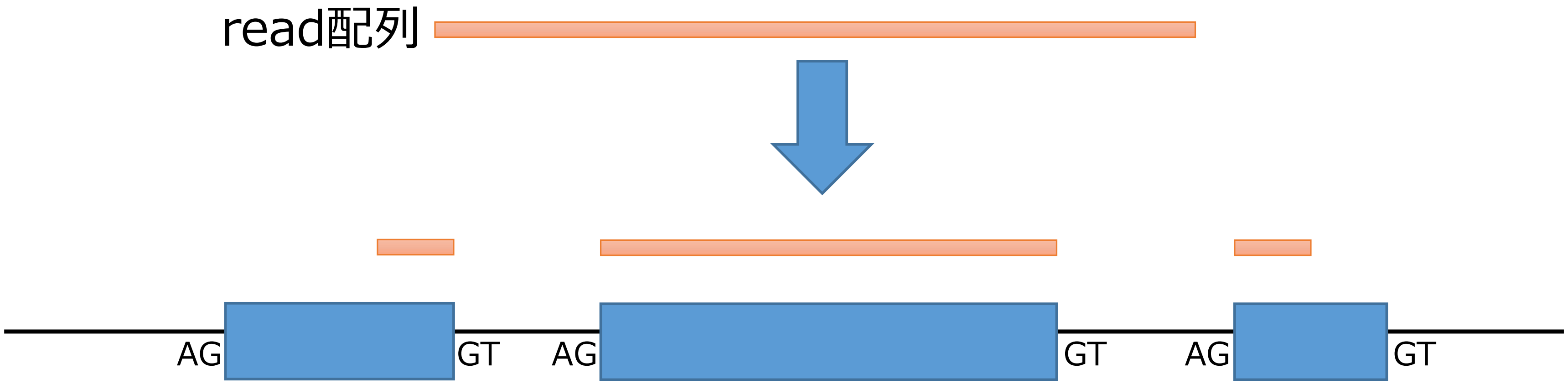
genomeをレファレンスとする場合

どのようなケースでgenomeをレファレンスとして解析するか

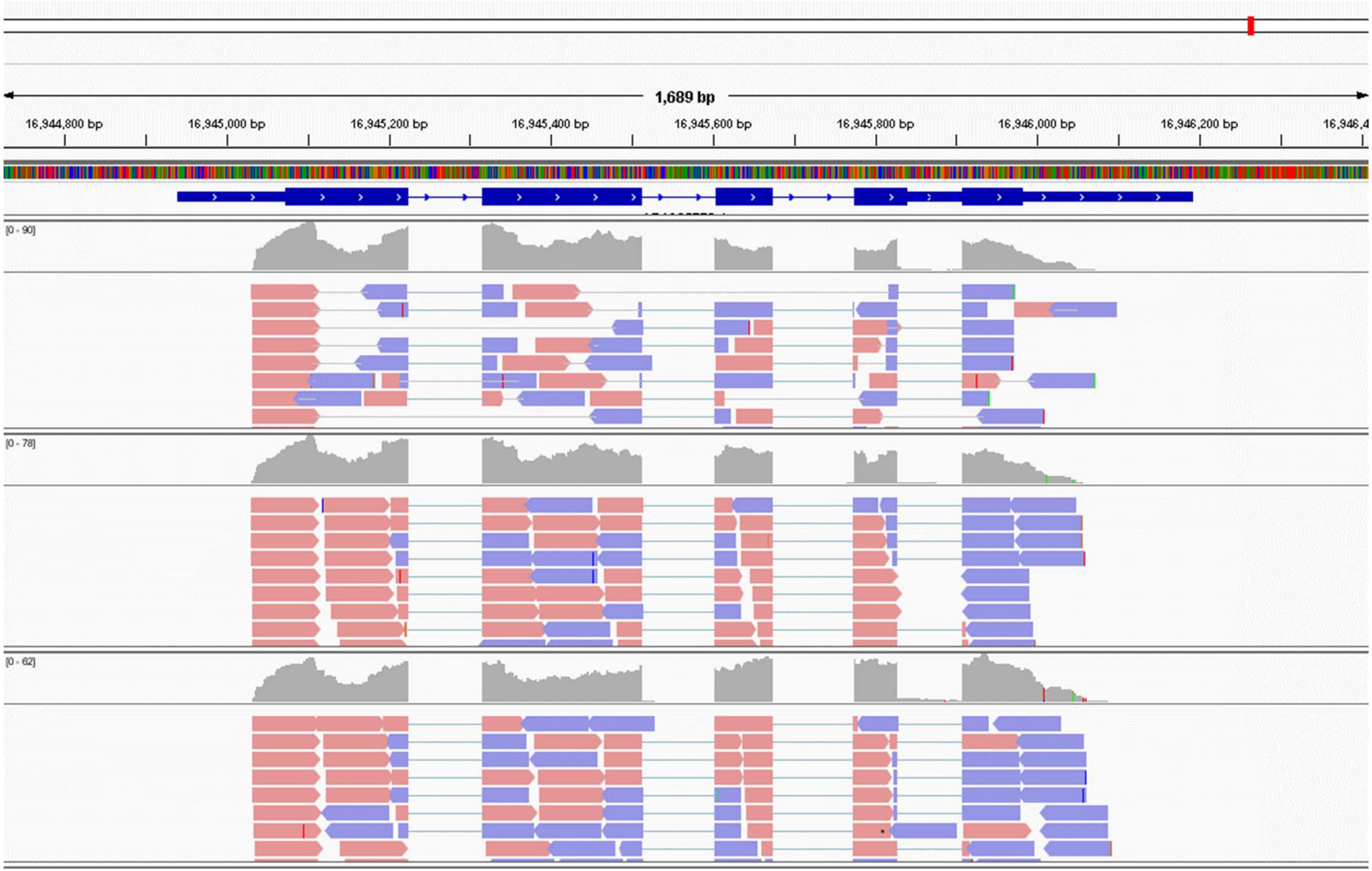
・ゲノム情報しかない、遺伝子情報のgftが不十分

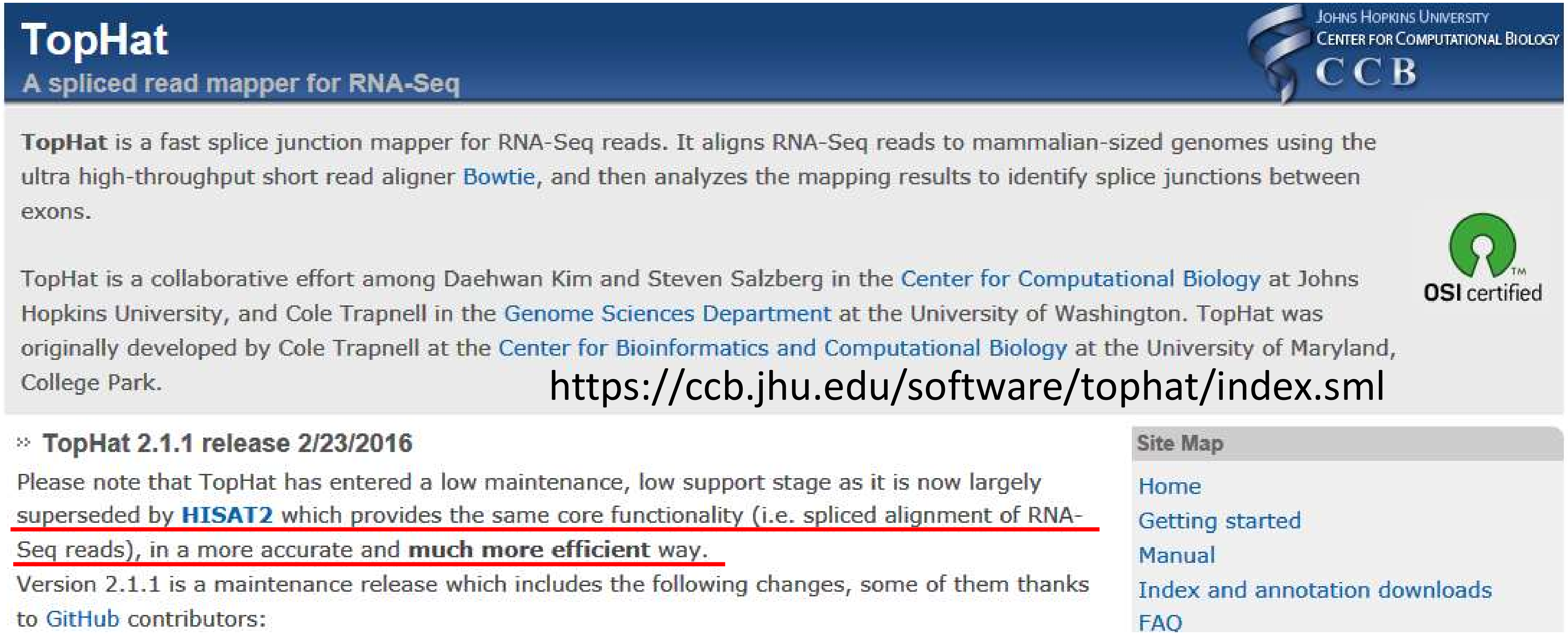
・新規な発現gene、発現エキソンを見出したい

Reference配列がゲノム配列の場合、
イントロン配列のスプライシングを考慮した
アライメントを行う必要がある。
今回はHISAT2を用いる
他　STAR,Tophat, Blat(Plat), MapSplice2



実際こんな感じにアラインされる





TopHat
A spliced read mapper for RNA-Seq

Johns Hopkins University
Center for Computational Biology
CCB

TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions between exons.

TopHat is a collaborative effort among Daehwan Kim and Steven Salzberg in the Center for Computational Biology at Johns Hopkins University, and Cole Trapnell in the Genome Sciences Department at the University of Washington. TopHat was originally developed by Cole Trapnell at the Center for Bioinformatics and Computational Biology at the University of Maryland, College Park.

OSI certified

https://ccb.jhu.edu/software/tophat/index.sml

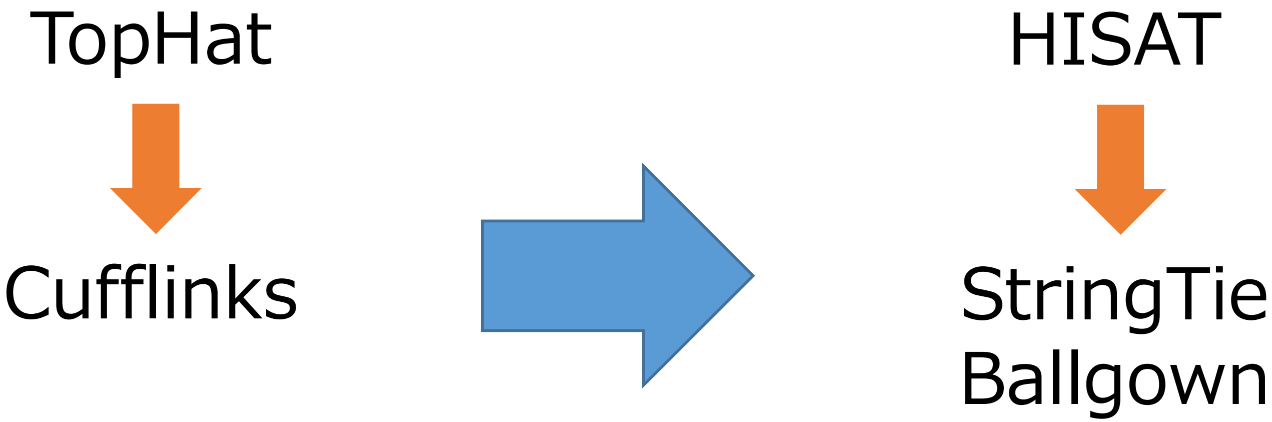
» **TopHat 2.1.1 release 2/23/2016**

Please note that TopHat has entered a low maintenance, low support stage as it is now largely superseded by **HISAT2** which provides the same core functionality (i.e. spliced alignment of RNA-Seq reads), in a more accurate and **much more efficient** way.

Version 2.1.1 is a maintenance release which includes the following changes, some of them thanks to GitHub contributors:

Traditional 'Tuxedo' package

New 'Tuxedo' package



TopHat

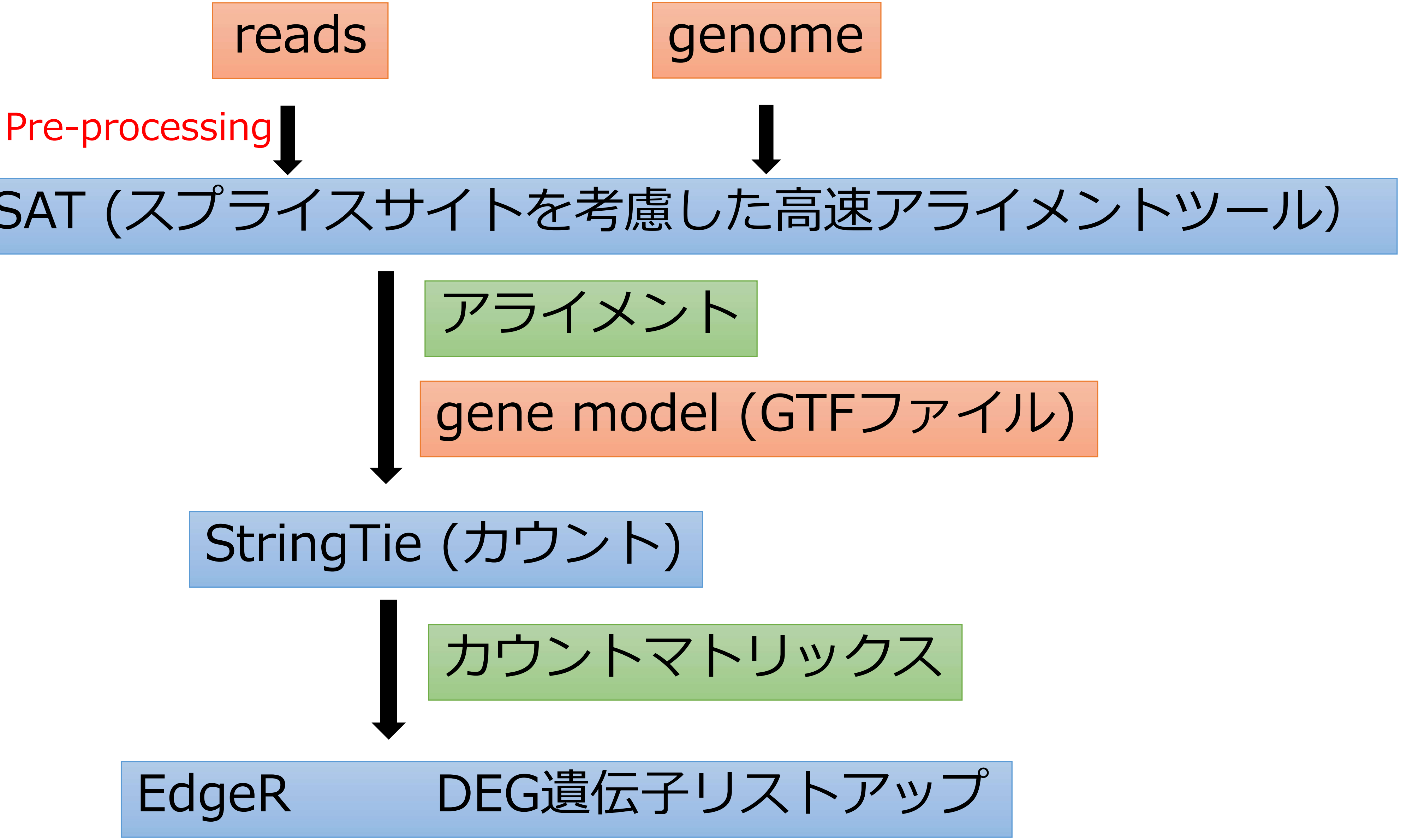
Cufflinks

HISAT

StringTie
Ballgown

劇的に解析速度が速くなった

本トレーニングコースでの流れ



reads

genome

Pre-processing

HISAT (スプライスサイトを考慮した高速アライメントツール)

アライメント

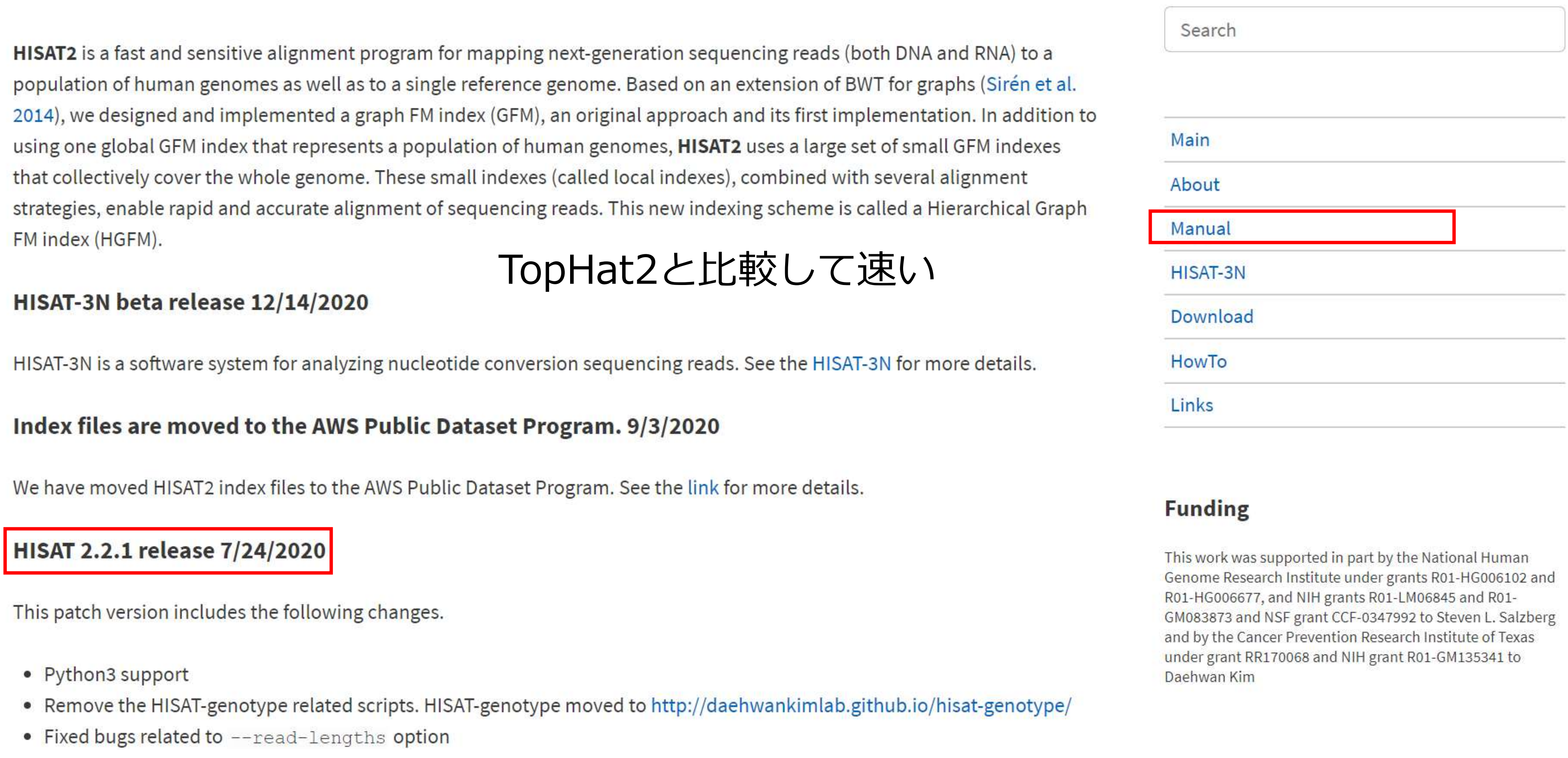
gene model (GTFファイル)

StringTie (カウント)

カウントマトリックス

EdgeR DEG遺伝子リストアップ

HISAT2



HISAT2 is a fast and sensitive alignment program for mapping next-generation sequencing reads (both DNA and RNA) to a population of human genomes as well as to a single reference genome. Based on an extension of BWT for graphs (Sirén et al. 2014), we designed and implemented a graph FM index (GFM), an original approach and its first implementation. In addition to using one global GFM index that represents a population of human genomes, **HISAT2** uses a large set of small GFM indexes that collectively cover the whole genome. These small indexes (called local indexes), combined with several alignment strategies, enable rapid and accurate alignment of sequencing reads. This new indexing scheme is called a Hierarchical Graph FM index (HGFM).

HISAT-3N beta release 12/14/2020

HISAT-3N is a software system for analyzing nucleotide conversion sequencing reads. See the HISAT-3N for more details.

Index files are moved to the AWS Public Dataset Program. 9/3/2020

We have moved HISAT2 index files to the AWS Public Dataset Program. See the link for more details.

HISAT 2.2.1 release 7/24/2020

This patch version includes the following changes.

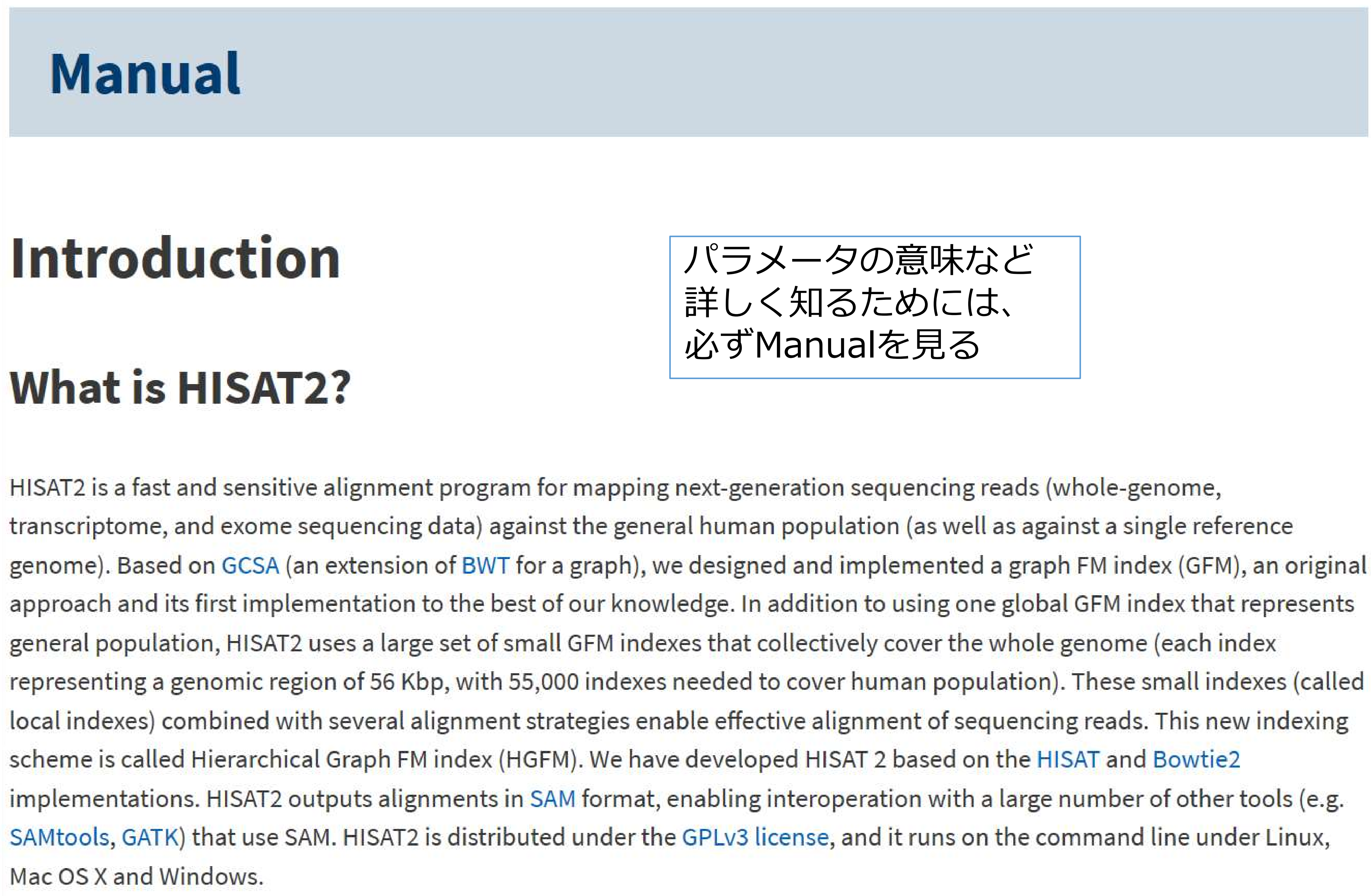
- Python3 support
- Remove the HISAT-genotype related scripts. HISAT-genotype moved to http://daehwankimlab.github.io/hisat-genotype/
- Fixed bugs related to `--read-lengths` option

Funding

This work was supported in part by the National Human Genome Research Institute under grants R01-HG006102 and R01-HG006677, and NIH grants R01-LM06845 and R01-GM083873 and NSF grant CCF-0347992 to Steven L. Salzberg and by the Cancer Prevention Research Institute of Texas under grant RR170068 and NIH grant R01-GM135341 to Daehwan Kim

TopHat2と比較して速い

http://daehwankimlab.github.io/hisat2



Manual

Introduction

What is HISAT2?

HISAT2 is a fast and sensitive alignment program for mapping next-generation sequencing reads (whole-genome, transcriptome, and exome sequencing data) against the general human population (as well as against a single reference genome). Based on GCSA (an extension of BWT for a graph), we designed and implemented a graph FM index (GFM), an original approach and its first implementation to the best of our knowledge. In addition to using one global GFM index that represents general population, HISAT2 uses a large set of small GFM indexes that collectively cover the whole genome (each index representing a genomic region of 56 Kbp, with 55,000 indexes needed to cover human population). These small indexes (called local indexes) combined with several alignment strategies enable effective alignment of sequencing reads. This new indexing scheme is called Hierarchical Graph FM index (HGFM). We have developed HISAT 2 based on the HISAT and Bowtie2 implementations. HISAT2 outputs alignments in SAM format, enabling interoperation with a large number of other tools (e.g. SAMtools, GATK) that use SAM. HISAT2 is distributed under the GPLv3 license, and it runs on the command line under Linux, Mac OS X and Windows.

パラメータの意味など
詳しく知るためには、
必ずManualを見る

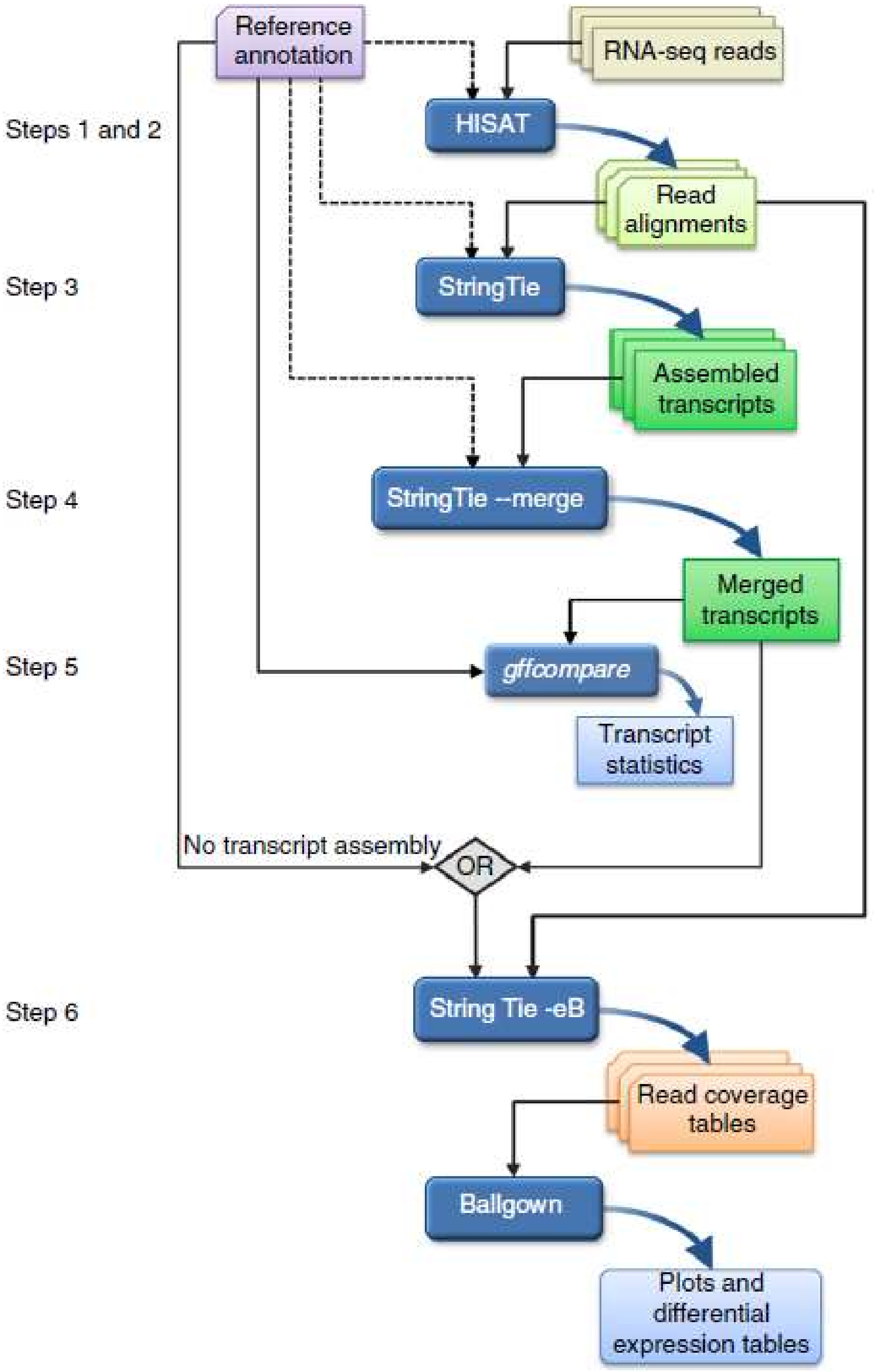
PROTOCOL

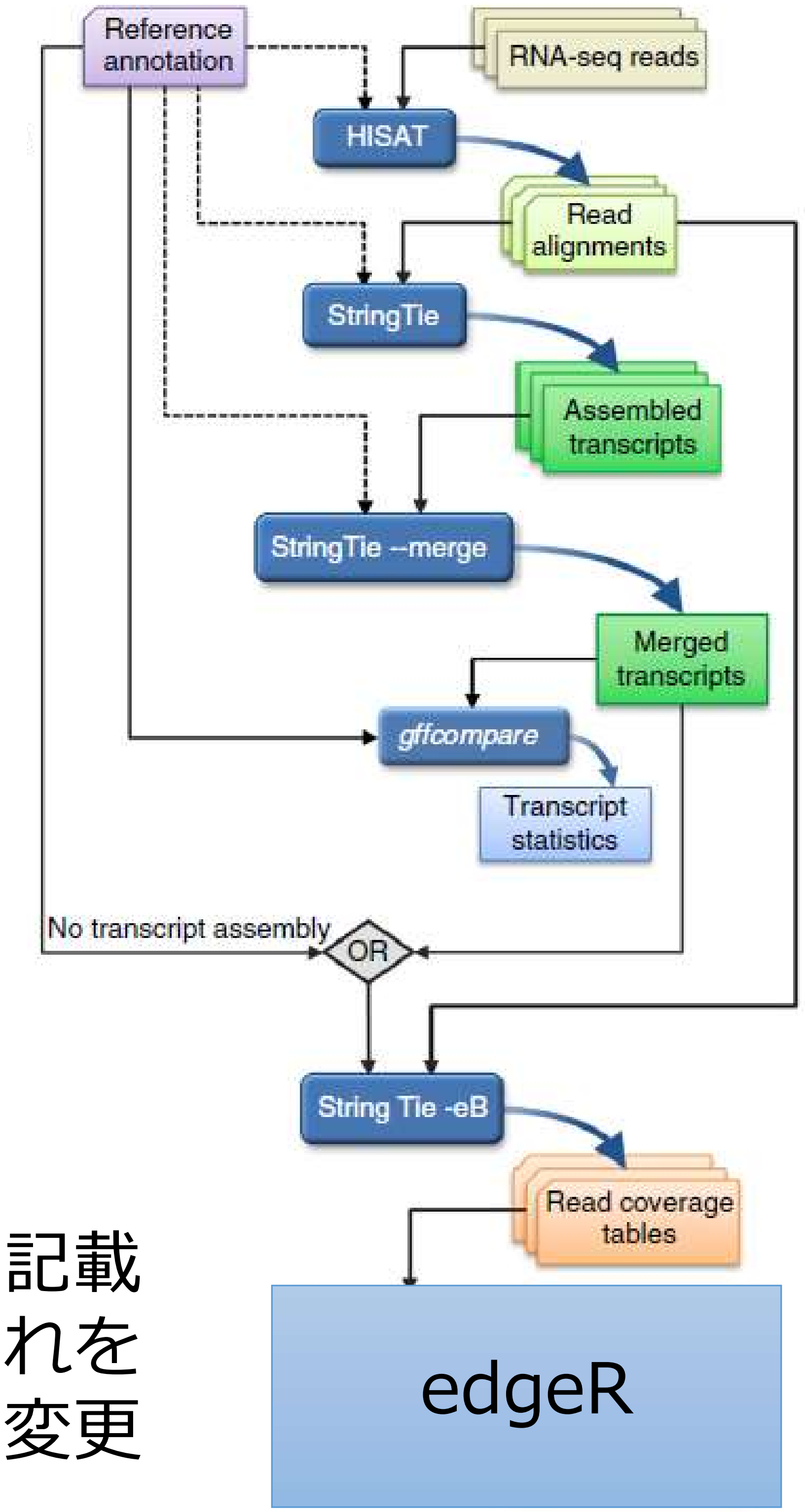
Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown

Mihaela Pertea[1,2], Daehwan Kim[1], Geo M Pertea[1], Jeffrey T Leek[3] & Steven L Salzberg[1–4]

[1]Center for Computational Biology, McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins School of Medicine, Baltimore, Maryland, USA. [2]Department of Computer Science, Whiting School of Engineering, Johns Hopkins University, Baltimore, Maryland, USA. [3]Department of Biostatistics, Bloomberg School of Public Health, Johns Hopkins University, Baltimore, Maryland, USA. [4]Department of Biomedical Engineering, Johns Hopkins University, Baltimore, Maryland, USA. Correspondence should be addressed to S.L.S. (salzberg@jhu.edu).

Published online 11 August 2016; doi:10.1038/nprot.2016.095





論文記載
の流れを
少し変更

hisat2-buildでリファレンスのインデックスを作る

```
$ hisat2-build -h
HISAT2 version 2.2.1 by Daehwan Kim (infphilo@gmail.com, http://www.ccb.jhu.edu/people/infphilo)
Usage: hisat2-build [options]* <reference_in> <ht2_index_base>
    reference_in            comma-separated list of files with ref sequences
    hisat2_index_base       write ht2 data to files with this dir/basename
Options:
    -c                      reference sequences given on cmd line (as
                            <reference_in>)
    --large-index           force generated index to be 'large', even if ref
                            has fewer than 4 billion nucleotides
    -a/--noauto             disable automatic -p/--bmax/--dcv memory-fitting
    -p <int>                number of threads
    --bmax <int>            max bucket sz for blockwise suffix-array builder
    --bmaxdivn <int>        max bucket sz as divisor of ref len (default: 4)
    --dcv <int>             diff-cover period for blockwise (default: 1024)
    --nodc                  disable diff-cover (algorithm becomes quadratic)
    -r/--noref              don't build .3/.4.ht2 (packed reference) portion
    -3/--justref            just build .3/.4.ht2 (packed reference) portion
    -o/--offrate <int>      SA is sampled every 2^offRate BWT chars (default: 5)
    -t/--ftabchars <int>    # of chars consumed in initial lookup (default: 10)
    --localoffrate <int>    SA (local) is sampled every 2^offRate BWT chars (default: 3)
    --localftabchars <int>  # of chars consumed in initial lookup in a local index (default: 6)
    --snp <path>            SNP file name
    --haplotype <path>      haplotype file name
    --ss <path>             Splice site file name
    --exon <path>           Exon file name
    --repeat-ref <path>     Repeat reference file name
    --repeat-info <path>    Repeat information file name
    --repeat-snp <path>     Repeat snp file name
    --repeat-haplotype <path>   Repeat haplotype file name
    --seed <int>            seed for random number generator
    -q/--quiet              disable verbose output (for debugging)
    -h/--help               print detailed description of tool and its options
    --usage                 print this usage message
    --version               print version information and quit
```

一部のモデル生物種以外は、リファレンス配列のインデックスを作る必要がある

実習1　hisat2-build

genome.faはArabidopsis thaliana (シロイヌナズナ）のレファレンスゲノム配列である。

中身を閲覧、query名およびreads数を確認せよ。

```
$ less genome.fa
$ grep '>' genome.fa
$ grep ‘>’ genome.fa|wc -l
```

indexを作製せよ。

```
$ hisat2-build genome.fa genome
```

新たに作製されたファイルを確認せよ。

```
$ ls –ltr      #新しいタイムスタンプのファイルが下になってリスト表示される
```

HISAT2基本コマンド

```
$ hisat2 -h
HISAT2 version 2.2.1 by Daehwan Kim (infphilo@gmail.com,
www.ccb.jhu.edu/people/infphilo)
Usage:
  hisat2 [options]* -x <ht2-idx> {-1 <m1> -2 <m2> | -U <r>} [-S <sam>]

  <ht2-idx>  Index filename prefix (minus trailing .X.ht2).
  <m1>       Files with #1 mates, paired with files in <m2>.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <m2>       Files with #2 mates, paired with files in <m1>.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <r>        Files with unpaired reads.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <sam>      File for SAM output (default: stdout)

  <m1>, <m2>, <r> can be comma-separated lists (no whitespace) and can be
  specified many times.  E.g. '-U file1.fq,file2.fq -U file3.fq'.
```

結果はsamファイルで出力される

実習2　hisat2

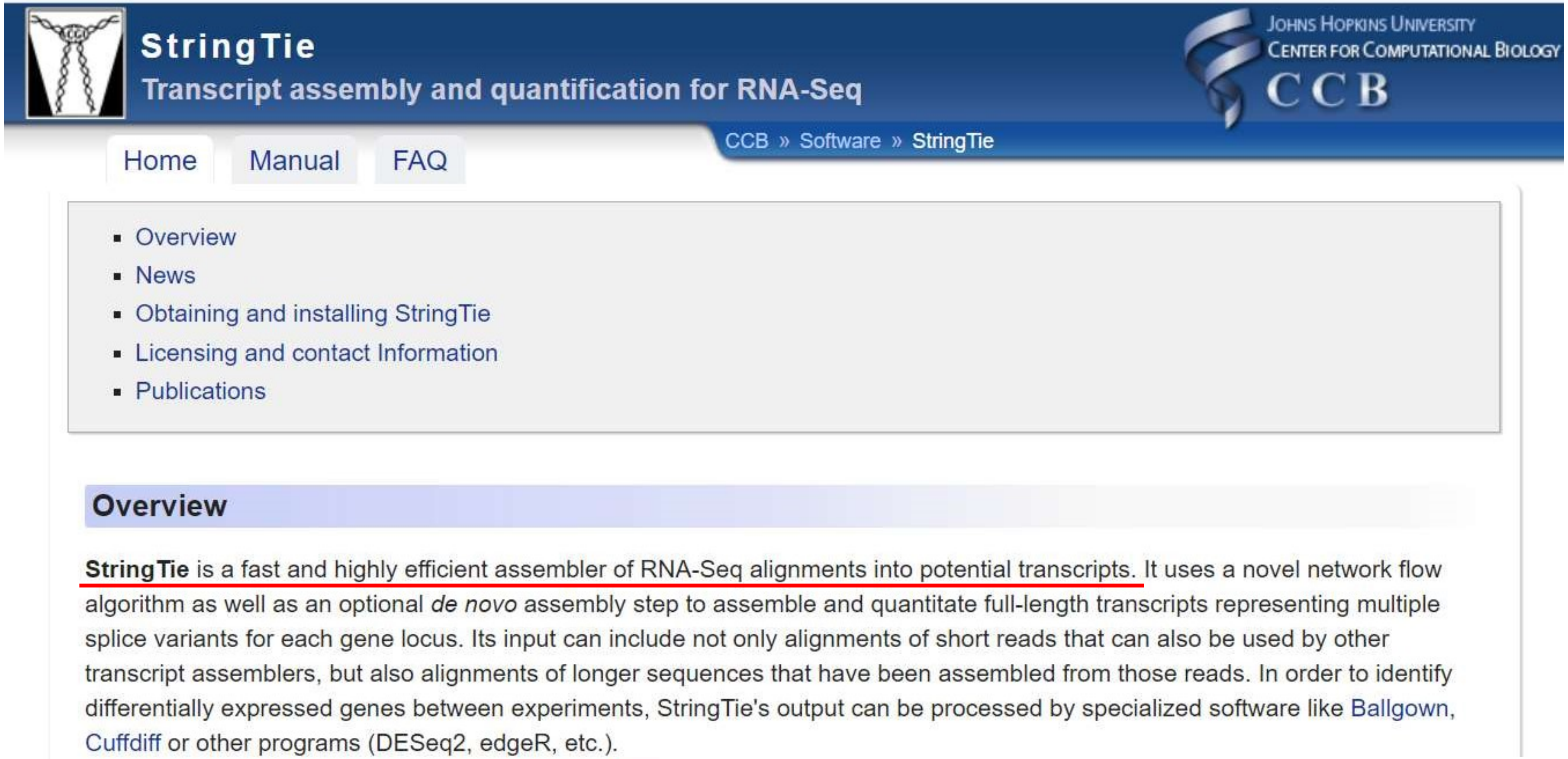
read結果
2D2L_rep1_R1.fastq
2D2L_rep1_R2.fastq
を先にindexを作製したリファレンスにmapさせよ。

```
$ hisat2 -p 4 --dta ¥
-x genome ¥
-1 2D2L_rep1_R1.fastq ¥
-2 2D2L_rep1_R2.fastq ¥
-S 2D2L_rep1.sam
```

¥は表記上、分かりやすくするための
改行を入れているが、入力の際は
改行せずに入力するという意味

samファイルの内容を確認しよう

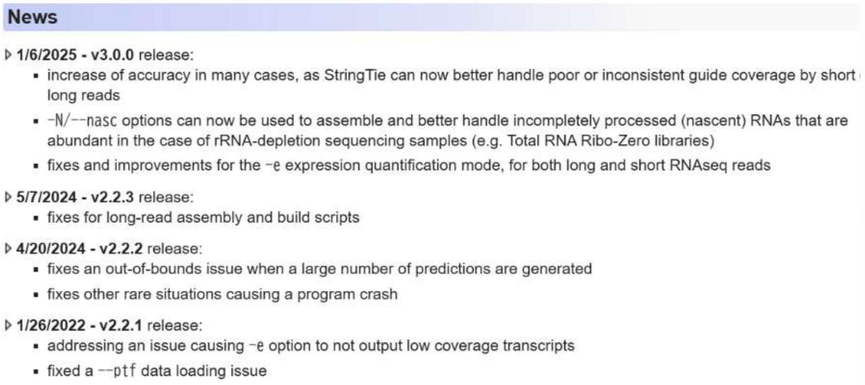
```
$ less 2D2L_rep1.sam
```


StringTie
Transcript assembly and quantification for RNA-Seq
Johns Hopkins University
Center for Computational Biology
CCB
Home
Manual
FAQ
CCB » Software » StringTie

- Overview
- News
- Obtaining and installing StringTie
- Licensing and contact Information
- Publications

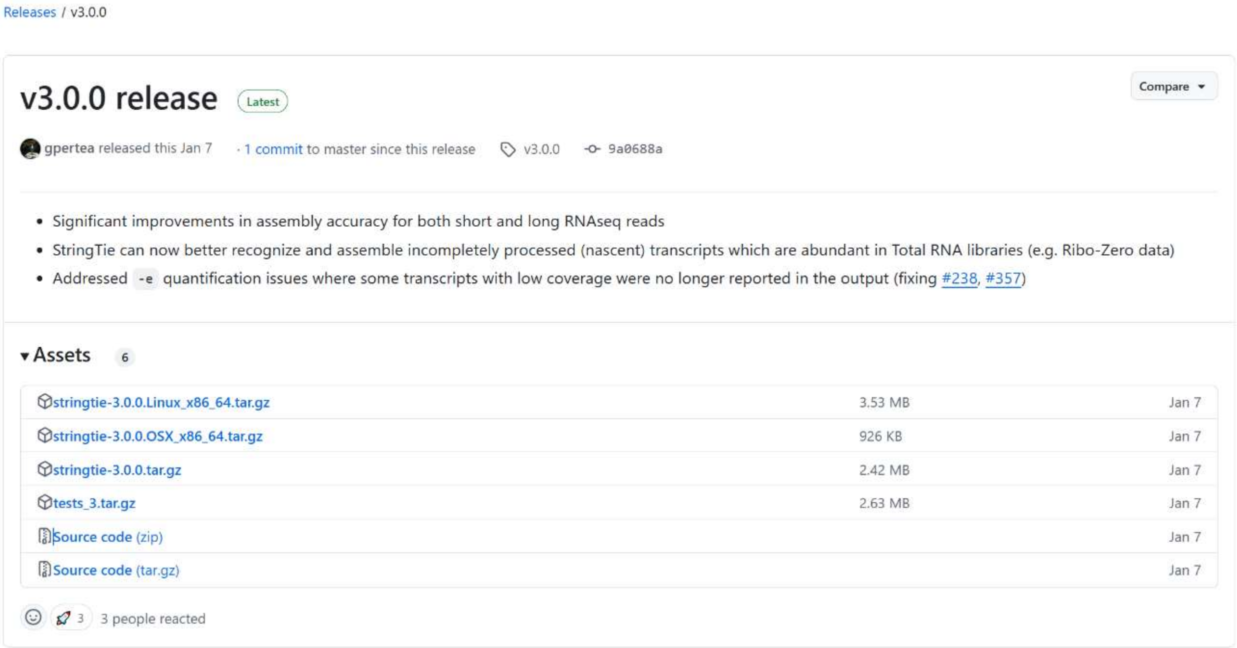
Overview

StringTie is a fast and highly efficient assembler of RNA-Seq alignments into potential transcripts. It uses a novel network flow algorithm as well as an optional *de novo* assembly step to assemble and quantitate full-length transcripts representing multiple splice variants for each gene locus. Its input can include not only alignments of short reads that can also be used by other transcript assemblers, but also alignments of longer sequences that have been assembled from those reads. In order to identify differentially expressed genes between experiments, StringTie's output can be processed by specialized software like Ballgown, Cuffdiff or other programs (DESeq2, edgeR, etc.).



News

- **1/6/2025 - v3.0.0** release:
  - increase of accuracy in many cases, as StringTie can now better handle poor or inconsistent guide coverage by short long reads
  - `-N/--nasc` options can now be used to assemble and better handle incompletely processed (nascent) RNAs that are abundant in the case of rRNA-depletion sequencing samples (e.g. Total RNA Ribo-Zero libraries)
  - fixes and improvements for the `-e` expression quantification mode, for both long and short RNAseq reads
- **5/7/2024 - v2.2.3** release:
  - fixes for long-read assembly and build scripts
- **4/20/2024 - v2.2.2** release:
  - fixes an out-of-bounds issue when a large number of predictions are generated
  - fixes other rare situations causing a program crash
- **1/26/2022 - v2.2.1** release:
  - addressing an issue causing `-e` option to not output low coverage transcripts
  - fixed a `--ptf` data loading issue



Releases / v3.0.0

v3.0.0 release

gpertea released this Jan 7 · 1 commit to master since this release v3.0.0 9a0688a

- Significant improvements in assembly accuracy for both short and long RNAseq reads
- StringTie can now better recognize and assemble incompletely processed (nascent) transcripts which are abundant in Total RNA libraries (e.g. Ribo-Zero data)
- Addressed `-e` quantification issues where some transcripts with low coverage were no longer reported in the output (fixing #238, #357)

Assets 6

stringtie-3.0.0.Linux_x86_64.tar.gz	3.53 MB	Jan 7
stringtie-3.0.0.OSX_x86_64.tar.gz	926 KB	Jan 7
stringtie-3.0.0.tar.gz	2.42 MB	Jan 7
tests_3.tar.gz	2.63 MB	Jan 7
Source code (zip)		Jan 7
Source code (tar.gz)		Jan 7

3 3 people reacted

https://github.com/gpertea/stringtie/releases/tag/v3.0.0

StringTieを用いてアラインされたreadを数える

StringTieの解析の方向性として大きく2つある

- GTFファイルに記載された遺伝子モデルのみを数える

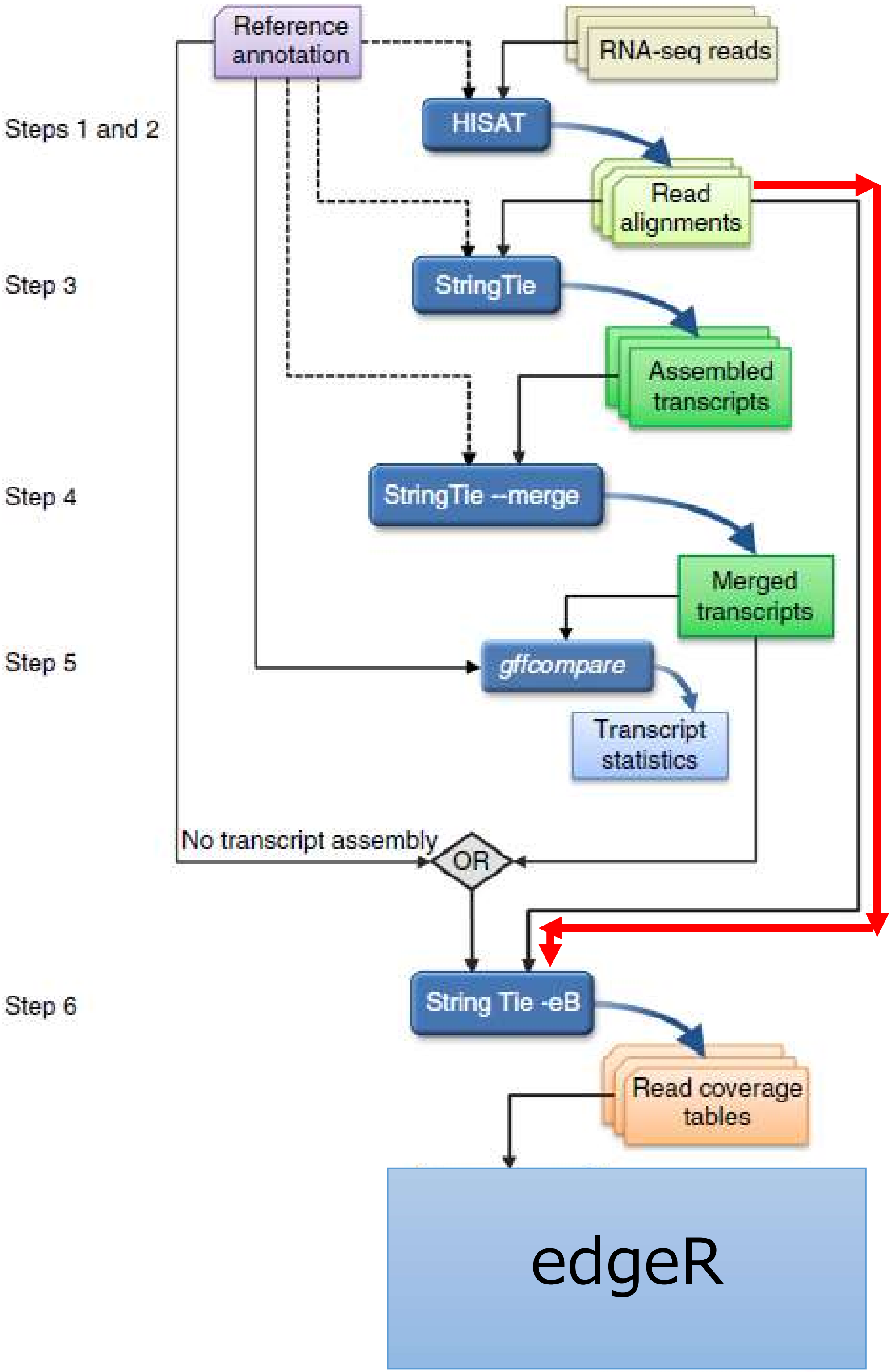
- トランスクリプトのアセンブルをおこない、新規な遺伝子モデルを見出し、それも含めて数える
  新規な遺伝子モデルはサンプルによって異なりうるので、得られた個々のモデルをStringTieのmerge modeでmergeすることで、全サンプル由来の新規遺伝子モデルを構築する

```
1   protein_coding   exon          3631   3913    .    +    .                 exon_number "1"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; seqedit "false"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   CDS            3760   3913    .    +    0                 exon_number "1"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; protein_id "AT1G01010.1"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   start_codon   3760   3762    .    +    0                 exon_number "1"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   CDS            3996   4276    .    +    2                 exon_number "2"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; protein_id "AT1G01010.1"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   exon           3996   4276    .    +    .                 exon_number "2"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; seqedit "false"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   CDS            4486   4605    .    +    0                 exon_number "3"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; protein_id "AT1G01010.1"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   exon           4486   4605    .    +    .                 exon_number "3"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; seqedit "false"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   CDS            4706   5095    .    +    0                 exon_number "4"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; protein_id "AT1G01010.1"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   exon           4706   5095    .    +    .                 exon_number "4"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; seqedit "false"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   CDS            5174   5326    .    +    0                 exon_number "5"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; protein_id "AT1G01010.1"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   exon           5174   5326    .    +    .                 exon_number "5"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; seqedit "false"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   CDS            5439   5627    .    +    0                 exon_number "6"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; protein_id "AT1G01010.1"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   exon          5439   5899    .    +    .                 exon_number "6"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; seqedit "false"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
1   protein_coding   stop_codon                         5628   5630    .    +    0 exon_number "6"; gene_id "AT1G01010"; gene_name "ANAC001"; p_id "P20332"; transcript_id "AT1G01010.1"; transcript_name "AT1G01010.1"; tss_id "TSS22545";
```

StringTie基本コマンド

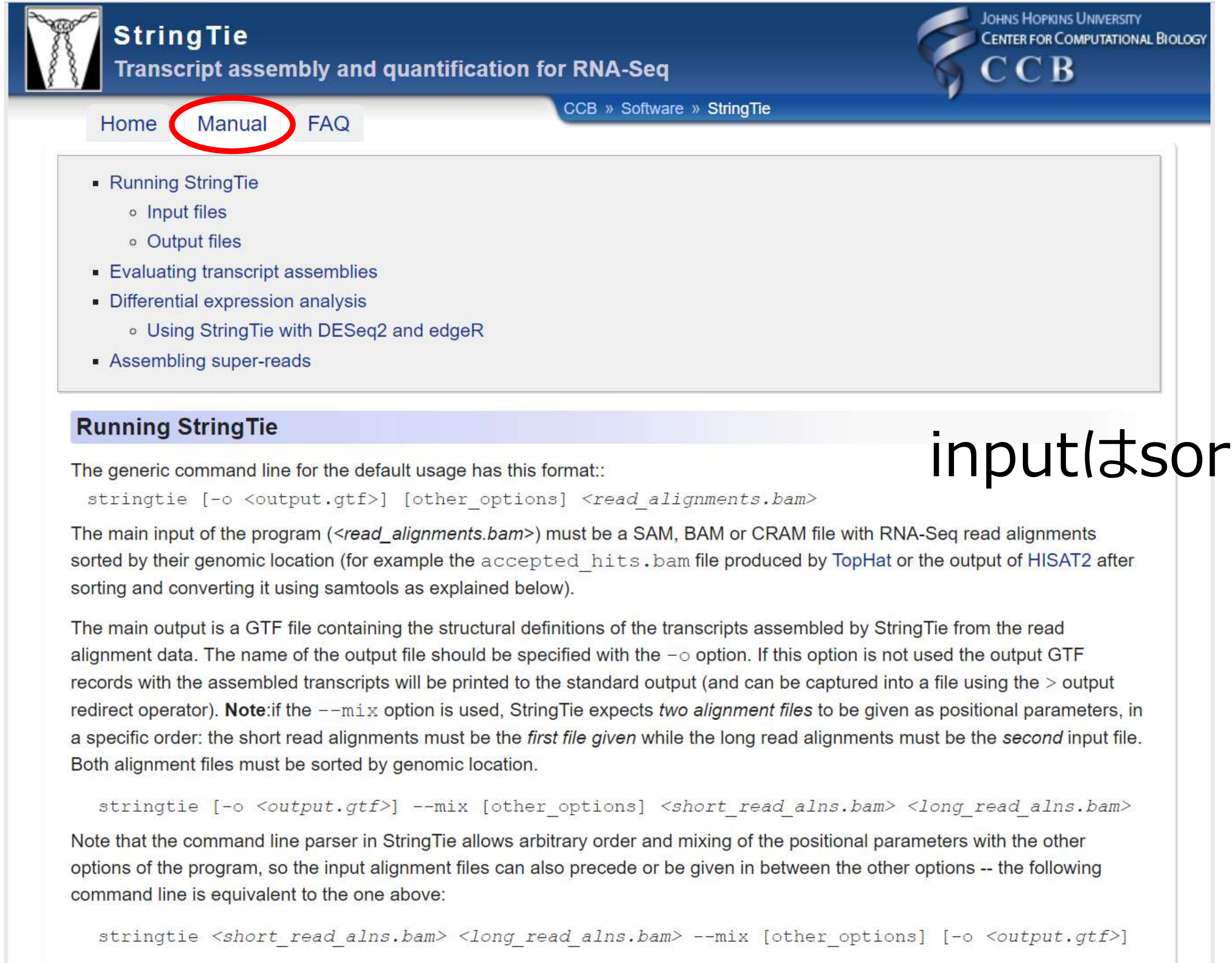
```
$ stringtie
StringTie v3.0.0 usage:

stringtie <in.bam ..> [-G <guide_gff>] [-l <prefix>] [-o <out.gtf>] [-p <cpus>]
 [-v] [-a <min_anchor_len>] [-m <min_len>] [-j <min_anchor_cov>] [-f <min_iso>]
 [-c <min_bundle_cov>] [-g <bdist>] [-u] [-L] [-e] [--viral] [-E <err_margin>]
 [--ptf <f_tab>] [-x <seqid,..>] [-A <gene_abund.out>] [-h] {-B|-b <dir_path>}
 [--mix] [--conservative] [--rf] [--fr]
Assemble RNA-Seq alignments into potential transcripts.
Options:
 --version : print just the version at stdout and exit
 --conservative : conservative transcript assembly, same as -t -c 1.5 -f 0.05
 --mix : both short and long read data alignments are provided
        (long read alignments must be the 2nd BAM/CRAM input file)
 --rf : assume stranded library fr-firststrand
 --fr : assume stranded library fr-secondstrand
 -G reference annotation to use for guiding the assembly process (GTF/GFF)
 --ptf : load point-features from a given 4 column feature file <f_tab>
 -o output path/file name for the assembled transcripts GTF (default: stdout)
         :
         :
         :
```



Case A

GTFファイルに記載された
遺伝子モデルのみを数える場合



StringTie
Transcript assembly and quantification for RNA-Seq
Johns Hopkins University
Center for Computational Biology
CCB
CCB » Software » StringTie
Home
Manual
FAQ

- Running StringTie
  - Input files
  - Output files
- Evaluating transcript assemblies
- Differential expression analysis
  - Using StringTie with DESeq2 and edgeR
- Assembling super-reads

Running StringTie

inputはsortされたBAM

The generic command line for the default usage has this format::

```
stringtie [-o <output.gtf>] [other_options] <read_alignments.bam>
```

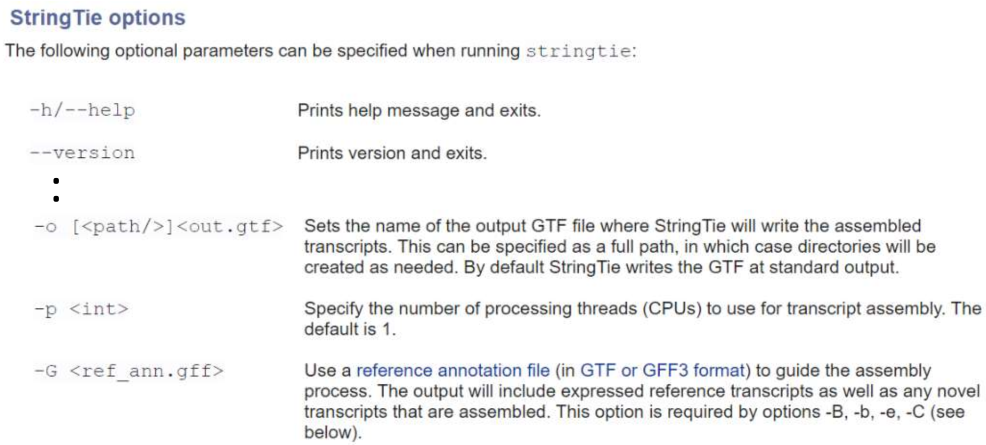
The main input of the program (*<read_alignments.bam>*) must be a SAM, BAM or CRAM file with RNA-Seq read alignments sorted by their genomic location (for example the `accepted_hits.bam` file produced by TopHat or the output of HISAT2 after sorting and converting it using samtools as explained below).

The main output is a GTF file containing the structural definitions of the transcripts assembled by StringTie from the read alignment data. The name of the output file should be specified with the `-o` option. If this option is not used the output GTF records with the assembled transcripts will be printed to the standard output (and can be captured into a file using the `>` output redirect operator). **Note**:if the `--mix` option is used, StringTie expects *two alignment files* to be given as positional parameters, in a specific order: the short read alignments must be the *first file given* while the long read alignments must be the *second* input file. Both alignment files must be sorted by genomic location.

```
stringtie [-o <output.gtf>] --mix [other_options] <short_read_alns.bam> <long_read_alns.bam>
```

Note that the command line parser in StringTie allows arbitrary order and mixing of the positional parameters with the other options of the program, so the input alignment files can also precede or be given in between the other options -- the following command line is equivalent to the one above:

```
stringtie <short_read_alns.bam> <long_read_alns.bam> --mix [other_options] [-o <output.gtf>]
```



StringTie options

The following optional parameters can be specified when running `stringtie`:

`-h/--help` Prints help message and exits.

`--version` Prints version and exits.

:

`-o [<path/>]<out.gtf>` Sets the name of the output GTF file where StringTie will write the assembled transcripts. This can be specified as a full path, in which case directories will be created as needed. By default StringTie writes the GTF at standard output.

`-p <int>` Specify the number of processing threads (CPUs) to use for transcript assembly. The default is 1.

`-G <ref_ann.gff>` Use a reference annotation file (in GTF or GFF3 format) to guide the assembly process. The output will include expressed reference transcripts as well as any novel transcripts that are assembled. This option is required by options -B, -b, -e, -C (see below).

```
$ stringtie ¥
-e ¥
-p 4 ¥
-G genes.gtf ¥
-o count_genes.gtf ¥
hoge.sort.bam
```

-G reference annotation to use for guiding the assembly process (GTF/GFF3)
-e only estimate the abundance of given reference transcripts (requires -G)
-p number of threads (CPUs) to use (default: 1)
-o output path/file name for the assembled transcripts GTF (default: stdout)

個々のサンプルごとに行う

実習3　stringtie

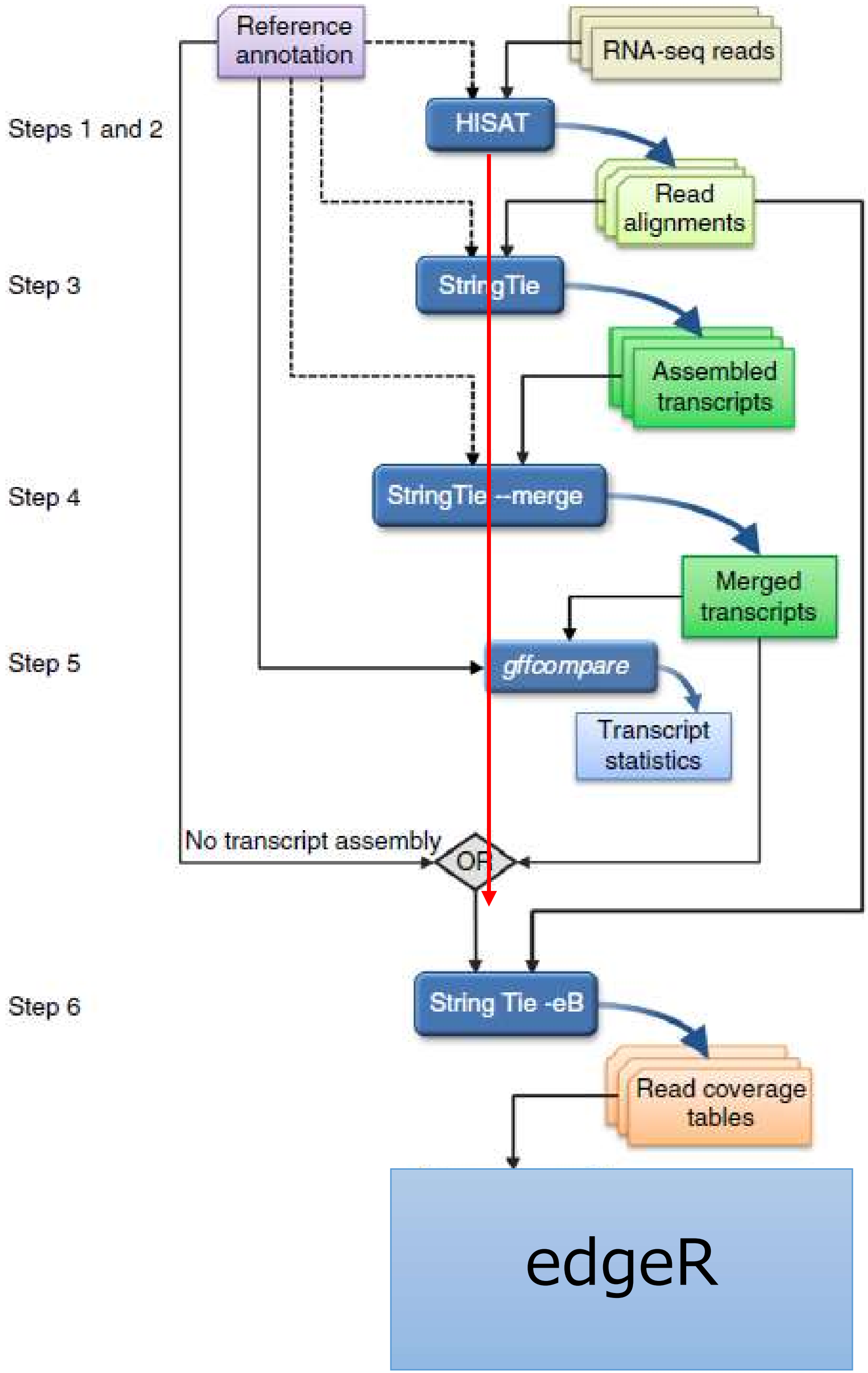
GTFファイルに記載された遺伝子モデルのみを対象とするケース

HISATで作製したsamをsort.bamにし、StringTieにかける
hisat結果　2D2L_rep1.sam

```
$ samtools sort ¥
-@ 4 ¥
-o 2D2L_rep1.sort.bam ¥
2D2L_rep1.sam

$ stringtie -e -p 4 ¥
-G genes.gtf ¥
-o count_2D2L_rep1.gtf ¥
2D2L_rep1.sort.bam
```

samtools v1.3以降、
samファイルのsortとbam化
は同時にできる



Case B

新規な遺伝子モデルを見出し、
それも数える場合

```
$ stringtie ¥
-p 4 ¥
-G genes.gtf ¥
-o count_genes.gtf ¥
hoge.sort.bam
```

-G reference annotation to use for guiding the assembly process (GTF/GFF3)
-p number of threads (CPUs) to use (default: 1)
-o output path/file name for the assembled transcripts GTF (default: stdout)

-e の指定はしない

個々のサンプルごとに
全サンプルで行う。

StringTieのmerge modeでmerged_gtfファイルを作製する

```
$ stringtie ¥
--merge ¥
-p 4 ¥
-G genes.gtf ¥
-o stringtie_merged.gtf ¥
sample.list   #個々のgtfファイルのスペース区切りでの羅列も可
```

Sample.list　　個々サンプルのgtfファイルを指定したファイルを用意

例)
count_2D_rep1.gtf
count_2D_rep2.gtf
count_2D_rep3.gtf
count_2D2L_rep1.gtf
count_2D2L_rep2.gtf
count_2D2L_rep3.gtf

新規に見出した遺伝子モデルを含む、
`stringtie_merged.gtf`が作成される。

次にこのmergeしたgtfファイルを-Gで指定して、先と同様-eを指定し、個々のbamからカウントデータを得る

```
$ stringtie ¥
-e ¥
-p 4 ¥
-G stringtie_merged.gtf ¥
-o count_genes.gtf ¥
hoge.sort.bam
```

-G reference annotation to use for guiding the assembly process (GTF/GFF3)
-e only estimate the abundance of given reference transcripts (requires -G)
-p number of threads (CPUs) to use (default: 1)
-o output path/file name for the assembled transcripts GTF (default: stdout)

個々のサンプルごとに
全サンプルで行う。

実習4　stringtie

新規な遺伝子モデルを見出し、それも対象とするケース

ここではアラビドプシス5本の染色体のうち、
chr5にあるすべてのgtf情報を削除したgtfファイル
（genes_except_chr5.gtf)をrefとする。

HISATで作製したsamをsort.bamにし、StringTieにかける
hisat結果　2D2L_rep1.sam

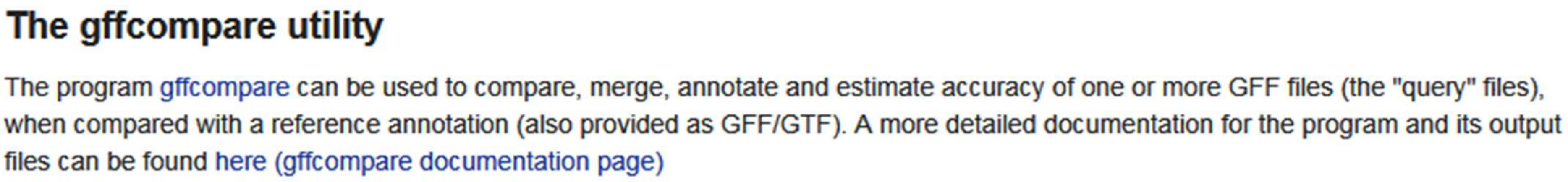
```
$ stringtie -p 4 ¥
-G genes_except_chr5.gtf ¥
-o count_2D2L_rep1.gtf ¥
2D2L_rep1.sort.bam
```

-eを抜く

得られたcount_2D2L_rep1.gtfを確認する

```
$ less gemes_exvept_chr5.gtf
```

GTFファイルを比較するツール



The gffcompare utility

The program gffcompare can be used to compare, merge, annotate and estimate accuracy of one or more GFF files (the "query" files), when compared with a reference annotation (also provided as GFF/GTF). A more detailed documentation for the program and its output files can be found here (gffcompare documentation page)

https://ccb.jhu.edu/software/stringtie/gff.shtml#gffcompare

```
gffcompare ¥
-r gene.gtf ¥
-o merged ¥
stringtie_merged.gtf
```

```
# gffcompare v0.12.9 | Command line was:
#gffcompare -r genes.gtf -o merged stringtie_merged.gtf
#

#= Summary for dataset: stringtie_merged.gtf
#     Query mRNAs :   42241 in   33367 loci  (30667 multi-exon
transcripts)
#            (6233 multi-transcript loci, ~1.3 transcripts per
locus)
# Reference mRNAs :   41607 in   33350 loci  (30127 multi-exon)
# Super-loci w/ reference transcripts:    33240
#-----------------| Sensitivity | Precision  |
        Base level:   100.0     |    99.8    |
        Exon level:   100.0     |    99.4    |
      Intron level:   100.0     |    99.8    |
Intron chain level:   100.0     |    98.2    |
  Transcript level:   100.0     |    98.5    |
       Locus level:   100.0     |    99.9    |

     Matching intron chains:   30127
       Matching transcripts:   41607
              Matching loci:   33350

          Missed exons:       0/169264  (  0.0%)
           Novel exons:     102/170581  (  0.1%)
        Missed introns:       0/127896  (  0.0%)
         Novel introns:      55/128111  (  0.0%)
           Missed loci:       0/33350   (  0.0%)
            Novel loci:      37/33367   (  0.1%)
```

gene.gtf ← 既知model
stringtie_merged.gtf ← 含新規 model

この両者を比較できる

Hisat2→StringTieでの新規gene探索テスト

TAIR10 gene annotationのgenes.gtfをrefにしたものと、
合わせてそこからchr5の情報のみ除去したものの2種をrefとして

アラビドプシス　leaf, flowerのRNA-Seqデータ (PE101)
cutadapt後　218.6M PE reads, 42.00GbaseのRNA-Seqデータ

Hisat2
Stringtieは
genes.gtfおよびgenes_except_chr5.gtfをrefとして、
-eを付けず新規モデルも探索するオプションで実行。
Stringtie mergeにかけ、merged.gtfを得る。

gffcompareで比較

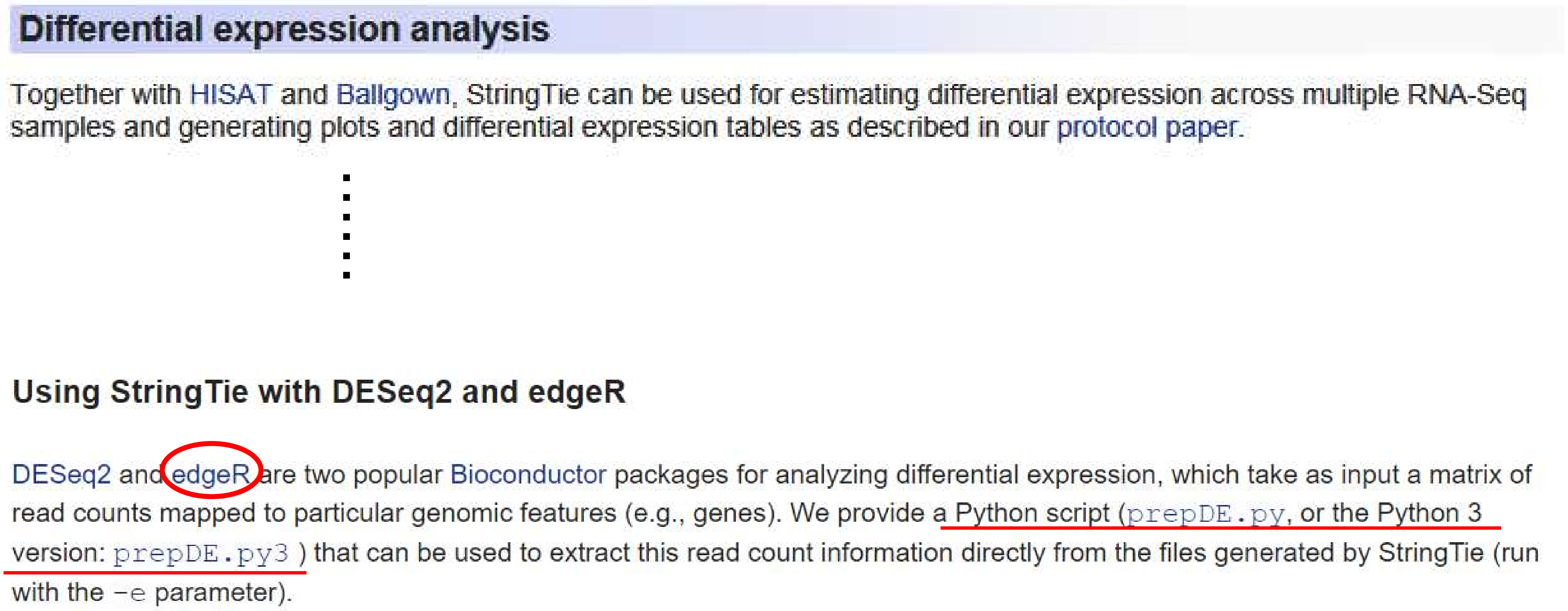
genes.gtf	41,611 mRNA (33,350 loci)
	↓
merged.gtf	52,535 mRNA (32,997 loci)

genes_except_chr5	32,322 mRNA (25,885 loci)
	↓
merged_except_chr5	47,874 mRNA (30,275 loci)

元のchr5にあるmodel	9,268 mRNA (7,453 loci)
新規にfinding出来たもの	7,127 mRNA (4,664 loci)

mRNA 76.9%　loci 62.6%
が見つけ出せた

Differential expression analysisへ



Differential expression analysis

Together with HISAT and Ballgown, StringTie can be used for estimating differential expression across multiple RNA-Seq samples and generating plots and differential expression tables as described in our protocol paper.

⋮

Using StringTie with DESeq2 and edgeR

DESeq2 and edgeR are two popular Bioconductor packages for analyzing differential expression, which take as input a matrix of read counts mapped to particular genomic features (e.g., genes). We provide a Python script (`prepDE.py`, or the Python 3 version: `prepDE.py3`) that can be used to extract this read count information directly from the files generated by StringTie (run with the `-e` parameter).

カウントマトリックス作製

```
$ python prepDE.py -h
Usage: prepDE.py [options]

Generates two CSV files containing the count matrices for genes and
transcripts, using the coverage values found in the output of `stringtie -e`

Options:
  -h, --help            show this help message and exit
  -i INPUT, --input=INPUT, --in=INPUT
                        the parent directory of the sample sub-directories or
                        a textfile listing the paths to GTF files [default:
                        ballgown]
  -g G                  where to output the gene count matrix [default:
                        gene_count_matrix.csv
  -t T                  where to output the transcript count matrix [default:
                        transcript_count_matrix.csv]
  -l LENGTH, --length=LENGTH
                        the average read length [default: 75]
  -p PATTERN, --pattern=PATTERN
                        a regular expression that selects the sample
                        subdirectories
  -c, --cluster         whether to cluster genes that overlap with different
                        gene IDs, ignoring ones with geneID pattern (see
                        below)
  -s STRING, --string=STRING
                        if a different prefix is used for geneIDs assigned by
                        StringTie [default: MSTRG]
  -k KEY, --key=KEY     if clustering, what prefix to use for geneIDs assigned
                        by this script [default: prepG]
  --legend=LEGEND       if clustering, where to output the legend file mapping
                        transcripts to assigned geneIDs [default: legend.csv]
```

```
$ python prepDE.py
```

gene_count_matrix.csv
transcript_count_matrix.csv

Case study 2: Genome-based RNA-Seq pipeline
を進め、確認してみよう。

```
gene_id,2D2L_rep1,2D2L_rep2,2D2L_rep3,2D2L_rep4,2D_rep1,2D_rep2,2D_rep3,4D_rep1,4
D_rep2,4D_rep3,4D_rep4
AT4G22890,295,204,203,154,20,22,17,35,26,17,22
AT1G38440,0,0,0,0,0,0,0,0,0,0,0
AT3G27910,0,0,0,0,0,0,0,0,0,0,0
AT1G06620,3,0,6,0,0,3,4,9,0,3,0
AT5G54067,0,0,0,0,0,0,0,0,0,0,0
AT2G34630,52,13,10,18,9,0,3,11,7,12,11
AT2G46660,0,0,0,3,4,0,0,16,23,3,6
AT2G25590,13,7,7,12,3,4,7,21,15,13,15
AT1G43171,0,0,0,0,0,0,0,0,0,0,0
AT5G25130,3,5,3,5,0,0,0,0,0,0,0
AT2G32280,6,0,7,0,5,0,15,0,5,6,0
AT3G15020,5,0,4,7,40,9,23,9,18,10,0
AT5G61100,0,0,0,0,0,0,0,0,0,0,0
AT5G01650,42,15,27,13,35,19,33,0,23,10,18
AT5G05570,6,8,4,4,3,5,3,0,11,9,3
AT3G09770,47,30,25,10,3,14,14,38,46,13,26
AT3G10210,9,0,5,12,0,7,12,20,9,9,3
AT5G06000,0,0,0,5,7,0,5,0,0,0,0
AT5G64620,40,31,20,31,64,35,41,21,37,41,36
AT1G75280,36,45,36,44,8,11,14,16,10,4,11
```

このカウントマトリックスファイルをedgeRへのinputとして、transcript base解析で扱った同一の方法で解析を進める。

edgeRでの解析

このケースでは　,が区切りのテキストとして得られているので、read.csvを用いる。

```
$ R
> library(edgeR)
> dat<-read.csv("gene_count_matrix.csv",row.names=1)
> group <- c(rep("2D",3),rep("2D2L",3))
> D<-DGEList(dat,group=group)
> D<-calcNormFactors(D)
> D<-estimateCommonDisp(D)
> D<-estimateTagwiseDisp(D)
```

```
2D vs 2D2Lの比較
> de_2D_2D2L <- exactTest(D,pair=c("2D","2D2L"))
> tmp <- topTags(de_2D_2D2L, n=nrow(de_2D_2D2L$table))
> write.table(tmp$table, "de.tagwise2.txt", sep="¥t", quote=F)
```

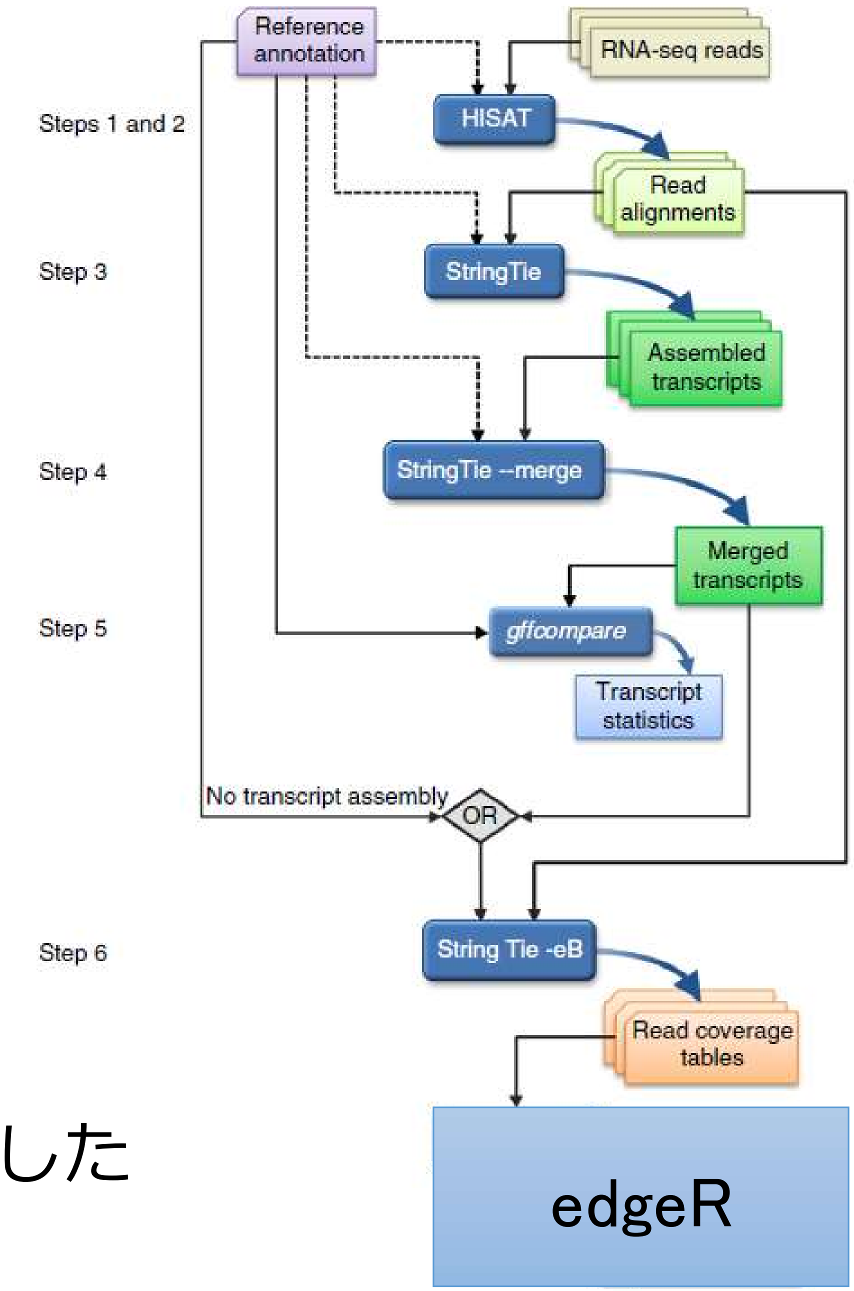
まとめ

HISAT

StringTie

edgeR

上記の流れを基盤にした、
genome baseのDEG解析を紹介した


Reference annotation
RNA-seq reads
Steps 1 and 2
HISAT
Read alignments
Step 3
StringTie
Assembled transcripts
Step 4
StringTie --merge
Merged transcripts
Step 5
gffcompare
Transcript statistics
No transcript assembly
OR
Step 6
String Tie -eB
Read coverage tables
edgeR