Mapping NGS Data
Sequence Analysis Pipeline

Transcript fragments

1. PREPROCESSING
   - Removal of contaminants, vector, adaptors, etc

2. ASSEMBLY (today)
   - Put overlapping sequence together and calculate bigger sequences

3. Analysis/Annotation
   - Discover biological significance

Rebuild transcriptome

Sol Genomics Network
Assembly
De novo Vs Reference guided

Reference-guided - map to previously-assembled genome or transcriptome

* pros: computationally easier
* cons: must have good reference assembly to a related species, may miss some transcripts

De novo - cluster reads into transcripts

* pros: no prior assemblies/annotations necessary
* cons: more difficult, more memory needed
Assembly

De novo Vs Reference guided

Reference-guided
today

De novo
Exercise 1

- Two RNA-seq datasets used in the tomato genome project were downloaded from the SRA in .sra format and extracted using the SRA toolkit (NCBI). [http://www.ncbi.nlm.nih.gov/sra](http://www.ncbi.nlm.nih.gov/sra)

- They were already cleaned using fastq-mcf. All data is from S. pimpinellifolium.

Datasets:
- breaker fruit (two files)
- immature fruit (two files)

In this exercise, we will map the reads to tomato chromosome 4 using reference-guided assembly
Reference-guided assembly

**HISAT** - Hierarchical Indexing for Spliced Alignment of Transcripts

**PROTOCOL**

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

Mihaela Pertea¹,², Daehwan Kim¹, Geo M Pertea¹, Jeffrey T Leek³ & Steven L Salzberg¹⁴

¹Center for Computational Biology, McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins School of Medicine, Baltimore, Maryland, USA. ²Department of Computer Science, Whiting School of Engineering, Johns Hopkins University, Baltimore, Maryland, USA. ³Department of Biostatistics, Bloomberg School of Public Health, Johns Hopkins University, Baltimore, Maryland, USA. ⁴Department 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

High-throughput sequencing of mRNA (RNA-seq) has become the standard method for measuring and comparing the levels of
Exercise 1

Make sure your machine has enough memory
machine -> settings -> System -> Base memory

- Run shell script from ~/Scripts/

```
$ cd ~/Scripts
$ wget ftp://solgenomics.net/bioinfo_class/BTIBIC/2018/map_rna_seq.sh
$ chmod 755 map_rna_seq.sh
$ sh map_rna_seq.sh
```

This will run for a few minutes
We’ll discuss what it is doing while it runs.
Exercise 1

• Run shell script `map_rna_seq.sh` (to be explained)

```bash
# copy data dir to desktop and extract
$ cd ~/Desktop
$ cp ~/Data/Slch04_demo.tar.gz .
$ tar -xvf Slch04_demo.tar.gz
$ rm Slch04_demo.tar.gz
$ cd Slch04_demo
```
map_rna_seq.sh

shell script that runs multiple linux commands - each can be typed in a terminal!

STEP 1: process reference genome sequence file for building index for hisat2

$ gffread ITAG3.10_gene_models.gff -o ITAG3.10_gene_models.gtf -T
map_rna_seq.sh

shell script that runs multiple linux commands - each can be typed in a terminal!

**STEP 1: process reference genome sequence file for building index for hisat2**

```
$ gffread ITAG3.10_gene_models.gff -o ITAG3.10_gene_models.gtf -T

$ python ~/Programs/hisat2-2.1.0/extract_splice_sites.py ./ITAG3.10_gene_models.gtf > splicesites.txt
```
map_rna_seq.sh

shell script that runs multiple linux commands - each can be typed in a terminal!

**STEP 1: process reference genome sequence file for building index for hisat2**

```bash
$ gffread ITAG3.10_gene_models.gff -o ITAG3.10_gene_models.gtf -T

$ python ~/Programs/hisat2-2.1.0/extract_splice_sites.py ./ITAG3.10_gene_models.gtf > splicesites.txt

$ hisat2-build --ss splicesites.txt S_lycopersicum_chromosomes.3.00_ch04.fa S_lycopersicum_chromosomes.3.00_ch04
```
map_rna_seq.sh

STEP 2: map each .fastq file to the reference with *hisat2*

```bash
#map reads with hisat2
for file in `dir -d *_ch4.fastq` ; do

    #create output file name
    samfile=`echo "$file" | sed 's/.fastq/.sam/'`
    metrics=`echo "$file" | sed 's/.fastq/.metrics/'

    #run mapping with hisat2
    hisat2 --max-intronlen 20000 --dta --met-file $metrics -p $CPU -x S_lycopersicum_chromosomes.3.00_ch04 -U $file -S $samfile

done
```
* Loop through each .fastq file in your working directory

```bash
#map reads with hisat2
for file in `dir -d *_ch4.fastq` ; do

    #create output file name
    samfile=`echo "$file" | sed 's/.fastq/.sam/'`
    metrics=`echo "$file" | sed 's/.fastq/.metrics/'`

    #run mapping with hisat2
    hisat2 --max-intronlen 20000 --dta --met-file $metrics -p $CPU -x S_lycopersicum_chromosomes.3.00_ch04 -U $file -S $samfile

done
```

bioinfo@biodebian:~/Desktop/Slch04_demo$ dir -dl *_ch4.fastq
-rw-r--r-- 1 bioinfo bioinfo 99770485 Mar 15 2013 SRR404331_ch4.fastq
-rw-r--r-- 1 bioinfo bioinfo 97357674 Mar 15 2013 SRR404333_ch4.fastq
-rw-r--r-- 1 bioinfo bioinfo 77462997 Mar 15 2013 SRR404334_ch4.fastq
-rw-r--r-- 1 bioinfo bioinfo 117216335 Mar 15 2013 SRR404336_ch4.fastq
* create `.sam` file names

```bash
# map reads with hisat2
for file in `dir -d *_ch4.fastq` ; do

    # create output file name
    samfile=`echo "$file" | sed 's/.fastq/.sam/''
    metrics=`echo "$file" | sed 's/.fastq/.metrics/''

    # run mapping with hisat2
    hisat2 --max-intronlen 20000 --dta --met-file $metrics -p $CPU \
    -x S_lycopersicum_chromosomes.3.00_ch04 -U $file -S $samfile

done
```

```
biinfo@biodebian:~/Desktop/Slch04_demo$ dir -dl *_ch4.fastq
-rw-r--r-- 1 bioinfo bioinfo  99770485 Mar 15 2013 SRR404331_ch4.fastq
-rw-r--r-- 1 bioinfo bioinfo  97357674 Mar 15 2013 SRR404333_ch4.fastq
-rw-r--r-- 1 bioinfo bioinfo  77462997 Mar 15 2013 SRR404334_ch4.fastq
-rw-r--r-- 1 bioinfo bioinfo 117216335 Mar 15 2013 SRR404336_ch4.fastq

SRR404331_ch4.sam

biinfo@biodebian:~/Desktop/Slch04_demo$ echo SRR404331_ch4.fastq | sed 's/.fastq/.sam/'
```

```
SRR404331_ch4.sam

biinfo@biodebian:~/Desktop/Slch04_demo$
```
* create `.metrics` file names

```bash
# map reads with hisat2
for file in `dir -d *_ch4.fastq`; do
    # create output file name
    samfile=`echo "$file" | sed 's/.fastq/.sam/'`
    metrics=`echo "$file" | sed 's/.fastq/.metrics/'`
    # run mapping with hisat2
    hisat2 --max-intronlen 20000 --dta --met-file $metrics -p $CPU --x S_lycopersicum_chromosomes.3.00_ch04 -U $file -S $samfile
done
```

```bash
bioinfo@biodebian:~/Desktop/Slch04_demo$ echo SRR404331_ch4.fastq | sed 's/.fastq/.metrics/'
SRR404331_ch4.metrics
bioinfo@biodebian:~/Desktop/Slch04_demo$
```
* Run hisat2 on each .fastq file

#map reads with hisat2
for file in `dir -d *_ch4.fastq` ; do

    #create output file name
    samfile=`echo "$file" | sed 's/.fastq/.sam/'`
    metrics=`echo "$file" | sed 's/.fastq/.metrics/'`

    #run mapping with hisat2
    hisat2 --max-intronlen 20000 --dta --met-file $metrics -p $CPU -x S_lycopersicum_chromosomes.3.00_ch04 -U $file -S $samfile

done

$  hisat2 --max-intronlen 20000 --dta --met-file SRR404331_ch4.metrics -p 1 -x S_lycopersicum_chromosomes.3.00_ch04 -U SRR404331_ch4.fastq -S SRR404331_ch4.sam
* Run samtools

```
#convert sam files to bam files to save space and sort
ls *.sam |parallel -j $CPU samtools view -Sb -o {.}.bam {}
r
rm *.sam
ls *.bam |parallel -j $CPU samtools sort -o {.}.sort.bam {}
r
rm *4.bam
ls *.sort.bam |parallel -j $CPU samtools flagstat {} ">" {.}.flagstat
```
List all .sam files

* `samtools view` - convert .sam to .bam

# convert sam files to bam files to save space and sort
ls *.sam | parallel -j $CPU samtools view -Sb -o {.}.bam {}
rm *.sam
ls *.bam | parallel -j $CPU samtools sort -o {.}.sort.bam {}
rm *4 bam
ls *.sort.bam | parallel -j $CPU samtools flagstat {}
> {.}.flagstat

Pipe output, use `parallel` for looping through all .sam files

Send each .sam file to `samtools view`

Output .bam files Base-name remains the same
* **samtools sort** - sort .bam output files

```bash
# convert sam files to bam files to save space and sort
ls *.sam | parallel -j $CPU samtools view -Sb -o {.}.bam {}
rm *.sam
ls *.bam | parallel -j $CPU samtools sort -o {.}.sort.bam {}
rm *4.bam
ls *.sort.bam | parallel -j $CPU samtools flagstat {} "">" { }.flagstat
```

- Remove original .sam files
- use *parallel* for looping through all .bam files and sorting by position
* **samtools flagstat - mapping efficiency %**

```bash
# convert sam files to bam files to save space and sort
ls *.sam | parallel -j $CPU samtools view -Sb -o {.}.bam {}
rm *.sam
ls *.bam | parallel -j $CPU samtools sort -o {.}.sort.bam {}
rm *4.bam
ls *.sort.bam | parallel -j $CPU samtools flagstat {} ">>" {.}.flagstat
```

- Remove original `.bam` files
- Use `parallel` for looping through all `.sort.bam` files and generate `.flagstat` files
Reference-guided assembly

```bash
#run stringtie and produce counts table for DE analysis with edgeR or DESeq
for file in `dir -d *.sort.bam` ; do
    outfile=`echo "$file" | sed 's/.bam/.gtf/'`
    outdir=`echo "$file" |sed 's/.bam//'
    stringtie -e -B -p $CPU -G ITAG3.10_gene_models.gtf -o ballgown/$outdir/$outfile $file
done
```

- Loop through all the `.sort.bam` files
- Create `.gtf` file names and output directory names
- Run `stringtie` - assembles RNA-Seq alignments into potential transcripts
Reference-guided assembly

**stringtie output**

- Gene Transfer Format (GTF) file that contains details of the transcripts that StringTie assembles from RNA-Seq data

- Ballgown Input Table Files

```
$ ls -ltrh
total 64M
-rw-r--r-- 1 bioinfo bioinfo 38M Mar 7 10:02 SRR404331_ch4.sort.gtf
-rw-r--r-- 1 bioinfo bioinfo 4.1M Mar 7 10:02 t_data.ctab
-rw-r--r-- 1 bioinfo bioinfo 13M Mar 7 10:02 e_data.ctab
-rw-r--r-- 1 bioinfo bioinfo 2.1M Mar 7 10:02 e2t.ctab
-rw-r--r-- 1 bioinfo bioinfo 6.1M Mar 7 10:02 i_data.ctab
-rw-r--r-- 1 bioinfo bioinfo 1.7M Mar 7 10:02 i2t.ctab
```

* We will use EdgeR for differential expression *
Assembly - viewing output

Tablet  http://bioinf.scri.ac.uk/tablet/

On your machine:
$ cd ~/Programs/Tablet
$ ./tablet
Assembly - viewing output

Tablet  http://bioinf.scri.ac.uk/tablet/

Index your .fasta and .bam files

1. reference sequence

```
samtools faidx S_lycopersicum_chromosomes.3.00_ch04.fa
```

2. bam file

```
samtools index SRR404331_ch4.sort.bam
```
Load into Tablet:
1. reference .fa file
2. mapped reads .bam file
3. annotations .gff file
Load into Tablet:
1. reference .fa file
2. mapped reads .bam file
3. annotations .gff file
Reference-guided assembly

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

Mihaela Pertea1, 2, Daehwan Kim1, Geo M Pertea1, Jeffrey T Leek3, Steven L Salzberg1, 2, 3, 4,

Affiliations | Contributions | Corresponding author

Published online 11 August 2016

edgeR
:a Bioconductor package for differential expression analysis of digital gene expression data

Mark D. Robinson; Davis J. McCarthy; Gordon K. Smyth

DOI: https://doi.org/10.1093/bioinformatics/btp616