Genome Assembly Software for Different Technology Platforms

PacBio	10x	Illumina
Canu	SuperNova	Soap Denovo
Falcon		Discovar
		Platinus
		MaSuRCA

Experimental design using Illumina Platform

Estimate genome size:
500 mb

Platform: Illumina Hiseq

Paired-end library: 150bp x 2 ; 2 lanes; ~100x coverage;
Mate pair library: three libraries (5kb, 10kb, 15kb), 1 lane

Software:
Soap denovo
Discovar
Platinus (for heterozygous genome)
MaSuRCA (hybrid, computationally demanding)

Large memory computer
BioHPC lab large memory server: 512mb to 1TB RAM

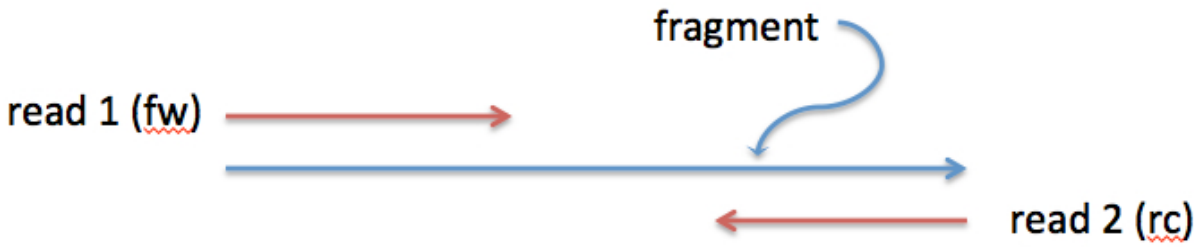
Workflow of de novo assembly

- Experimental Design
- Clean sequencing data
  (trim adapter and low quality sequences)
- Run assembly software for contiging and scaffolding
- Evaluation of assembly

Several iterations: adjust setting and software, or add more reads to improve assemblies

DISCOVAR approach

- **Library fragment size: ~450 bp**
- **Read size: >= 250 bp x2**
- **Depth: about 60x**



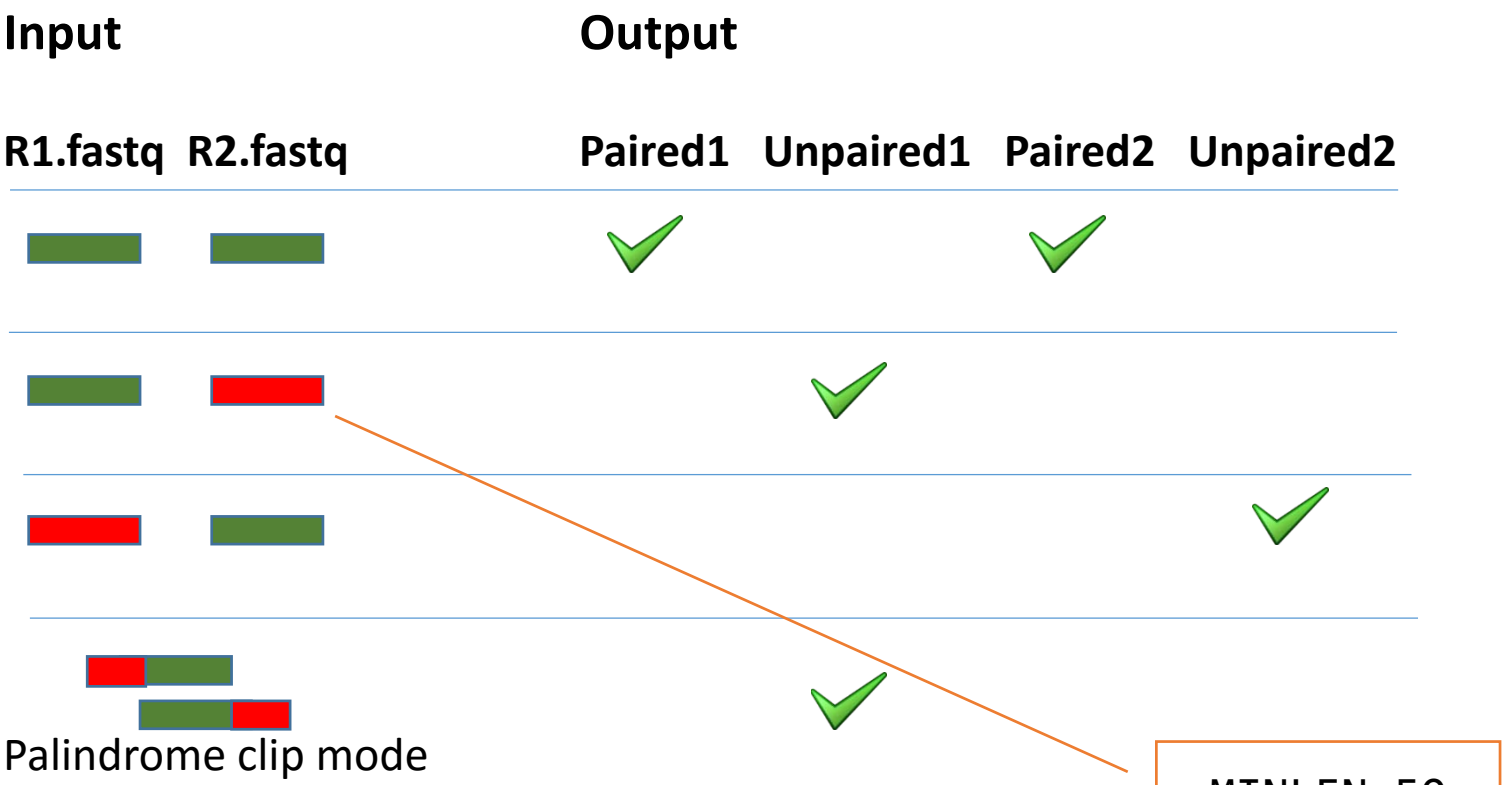
Data cleaning with Trimmomatic

- Trim low quality data (quality score based trimming)
- Clip sequencing adapters (alignment to adapter sequence)

```
java -jar /programs/trimmomatic/trimmomatic-0.32.jar PE -phred33 \
SRR1554178_1.fastq SRR1554178_2.fastq \
r1.fastq u1.fastq r2.fastq u2.fastq \
ILLUMINACLIP:/programs/trimmomatic/adapters/TruSeq3-PE-2.fa:2:30:10
LEADING:10 \
TRAILING:10 \
SLIDINGWINDOW:4:15 \
MINLEN:50
```

Minimum read length to keep

Trimmomatic



r1.fastq	417503786
r2.fastq	411903328
u1.fastq	62712666
u2.fastq	2776034

Running assembly software

Testing different size kmers and assembly software.

Not always possible, as assembly of large genomes takes very long time on a large memory computer.

Test different kmer sizes

Two binary codes with max kmer size 63 or 127

SOAP denovo2 on BioHPC Lab

/programs/SOAPdenovo2/SOAPdenovo-63mer [options]
/programs/SOAPdenovo2/SOAPdenovo-127mer [options]

```
SOAPdenovo-127mer all -s config.txt -K 101 -R -o assembly

SOAPdenovo-127mer all -s config.txt -K 127 -R -o assembly
```

SOADdenovo config file

```
#maximal read length
max_rd_len=101
[LIB]
#average insert size
avg_ins=300
#if sequence needs to be reversed
reverse_seq=0
#in which part(s) the reads are used
asm_flags=3
#in which order the reads are used while scaffolding
rank=1
# cutoff of pair number for a reliable connection (at least 3 for short insert size)
pair_num_cutoff=3
#minimum aligned length to contigs for a reliable read location (at least 32 for short insert s
map_len=32
#a pair of fastq file, read 1 file should always be followed by read 2 file
q1=r1.fastq
q2=r2.fastq
```

```
/programs/SOAPdenovo2/SOAPdenovo-127mer all -s config.txt -K 127 -R -o assembly
```

avg_ins=2000 reverse_seq=1

Multiple libraries can be mixed in one assembly

```
[LIB]
avg_ins=450
reverse_seq=0
asm_flags=3
q1=r1.fastq
q2=r2.fastq
[LIB]
asm_flags=1
q=u1.fastq
[LIB]
avg_ins=2000
reverse_seq=1
asm_flags=3
q1=r1.fastq
q2=r2.fastq
```

Multi-Kmer Approach in Soap Denovo2

- **Start building graph with small kmer;**
- **Iteratively rebuild kmer by mapping larger kmers to previous graph**

//Start

k <- k_{min} (k_{min} is set at graph construction 'pregraph' step);

Construct initial *de Bruijn* graph with k_{min};

Remove low depth k-mers and cut tips;

Merge bubbles of the de Bruijn graph;

Repeat {

    k <- k + 1;

Get contig graph H_k from previous loop or construct from *de Bruijn* graph;

    Map reads to H_k and remove the reads already represented in the graph;

Construct H_{k+1} graph base on H_k graph and the remaining reads with k;

Remove low depth edges and weak edges in H_k;

} Stop if k >= k_{max}(k_{max} = k set in contig step(-m));

Cut tips and merge bubbles;

Output all contigs;

//End

Multi-Kmer Approach in Soap Denovo2

```
SOAPdenovo-127mer all -s config.txt -K 95 -m 127 -R -o assembly
```

-K: starting Kmer

-m: end kmer

Run DISCOVAR on BioHPC Lab

Use PICARD to convert fastq.gz files to bam file

```
export JAVA_HOME=/usr/local/jdk1.8.0_45
export PATH=$JAVA_HOME/bin:$PATH
java -jar /programs/picard-tools-2.1.1/picard.jar FastqToSam \
    FASTQ=file1.fastq.gz   FASTQ2=file2.fastq.gz \
    O=reads.bam \
```

Run Discovar

```
/programs/discovar/bin/Discovar \
    READS=reads.bam \
    OUT_HEAD=assembly \
    REGIONS=all
```

Evaluation of Genome assembly 1
Metrics for contig length

N50 and L50 *

N50 50% (base pairs) of the assemblies are contigs above this size.

L50 Number of contigs greater than the N50 length.

NG50 and LG50

N50 is calculated based on assembly size. NG50 is calculated based on estimated genome size.

Standalone tools for generating metrics

(Most assembly software provides N50/L50 metrics in the report)

Quast (http://bioinf.spbau.ru/quast)

- Computing evaluation metrics

- Comparing with a reference genome (or between assemblies)
  - Structure variation;
  - Genome fraction: % represented reference genome
  - Duplication ration: copy number ratio between assembly and reference in aligned region.
  - Reference gene representation .

2. REAPR: Scoring each base of the assembly based on alignment of paired-end reads

Input:
BAM file from alignment of reads to the assembly
(independent alignment of paired ends)

Metrics reported by REAPR:

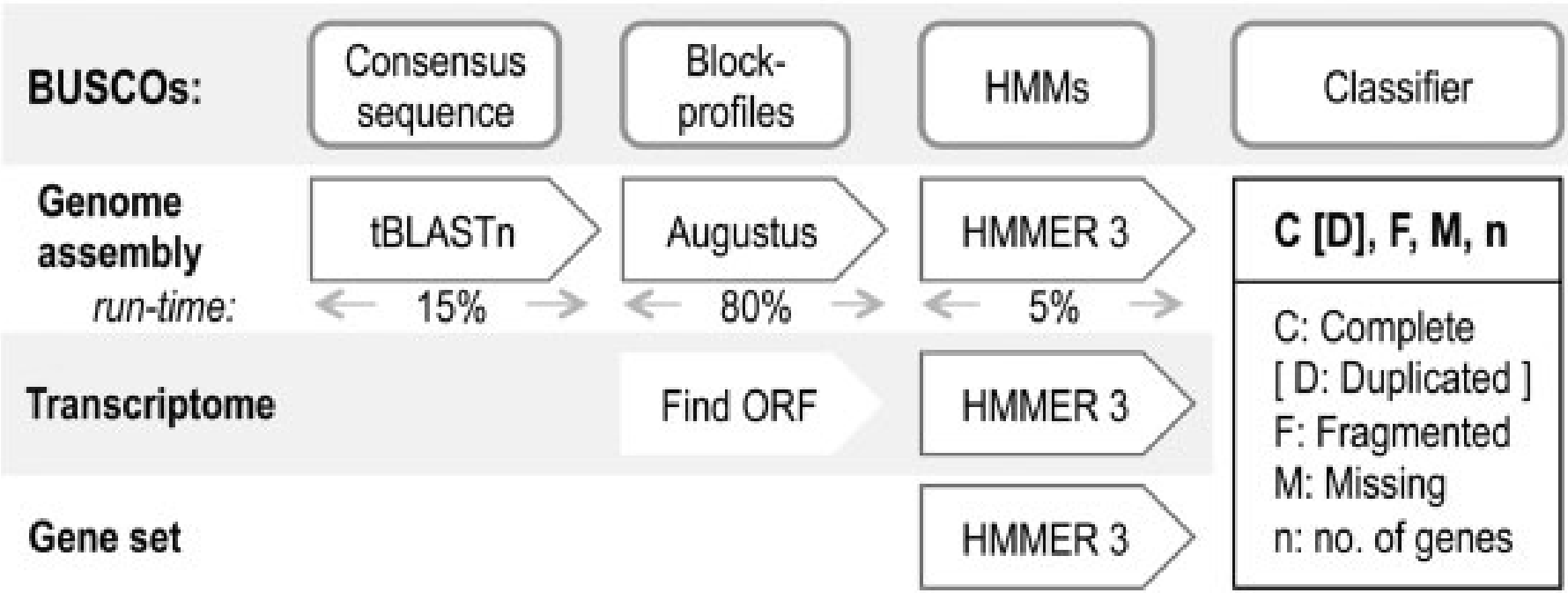
- Scaffold errors
- % of error free bases

3. Evaluate by gene content - BUSCO

BUSCO gene sets:
single-copy orthologs in at least 90% of the species in each lineage.

Arthropods Vertebrates Fungi Bacteria
Metazoans Ekaryotes Plants

BUSCO assessment workflow



Sima˜o et al. Bioinformatics, 2015, 1–3

BUSCO Output

C:complete D:duplicated F:fragmented M:missing
(Report % of genes in each category)

Species	Size	BUSCO notation assessment results
D. mela	139 Mbp	C:98% [D:6.4%], F:0.6%, M:0.3%, n:2 675
	13 918 genes	C:99% [D:3.7%], F:0.2%, M:0.0%, n:2 675
C. eleg	100 Mbp	C:85% [D:6.9%], F:2.8%, M:11%, n:843
	20 447 genes	C:90% [D:11%], F:1.7%, M:7.5%, n:843
H. sapi	3 381 Mbp	C:89% [D:1.5%], F:6.0%, M:4.5%, n:3 023
	20 364 genes	C:99% [D:1.7%], F:0.0%, M:0.0%, n:3 023
L. giga	359 Mbp	C:89% [D:2.3%], F:4.3%, M:5.8%, n:843
	23 349 genes	C:90% [D:13%], F:7.8%, M:2.1%, n:843
A. nidu	30 Mbp	C:98% [D:1.8%], F:0.9%, M:0.2%, n:1 438
	10 534 genes	C:95% [D:7.3%], F:3.8%, M:0.9%, n:1 438

Sima˜o et al. Bioinformatics, 2015, 1–3

Run BUSCO on BioHPC Lab

A copy of August in working directory (writable)

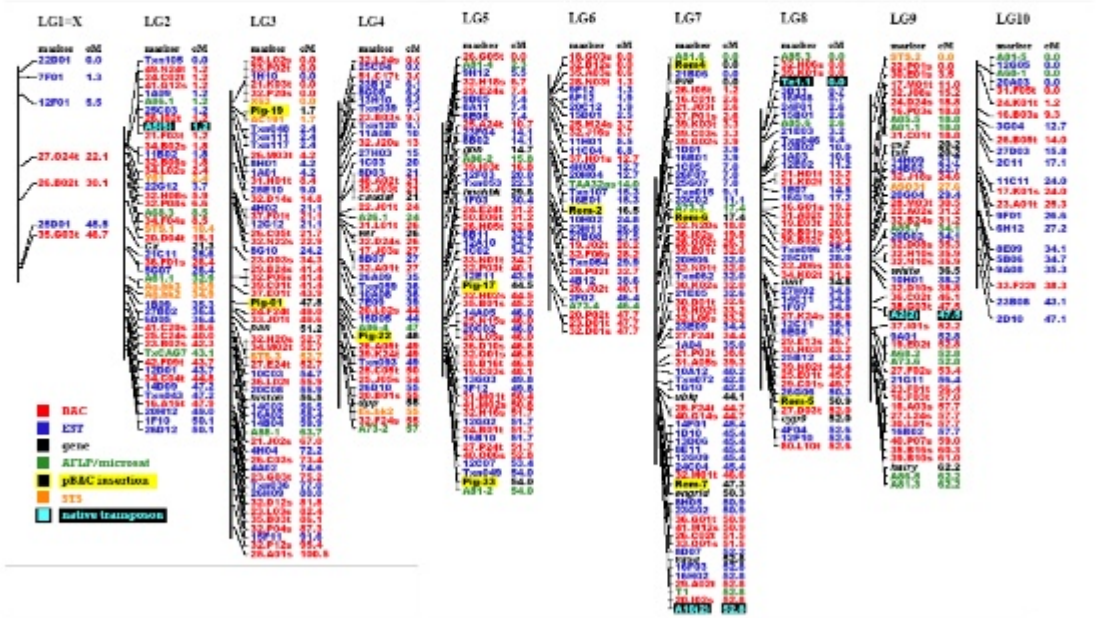
```
cp -r /programs/augustus-3.2.1 ./
# set PATH for required software
export AUGUSTUS_CONFIG_PATH=/workdir/XXX/augustus.2.5.5/config
export PATH=/programs/hmmer/binaries:/programs/emboss/bin:$PATH
export PATH=/workdir/XXX/augustus-3.2.1/bin:$PATH

python3 /programs/BUSCO_v1.2/BUSCO_v1.2.py -o SAMPLE -in
assembly.fa -l lineage_db -m genome
```

4. Evaluate by physical or genetic map

Using Physical Map to Anchor Scaffold to Chromosome

Molecular map markers used to anchor scaffolds to Chromosome builds



Few X markers, no Y, variable marker density

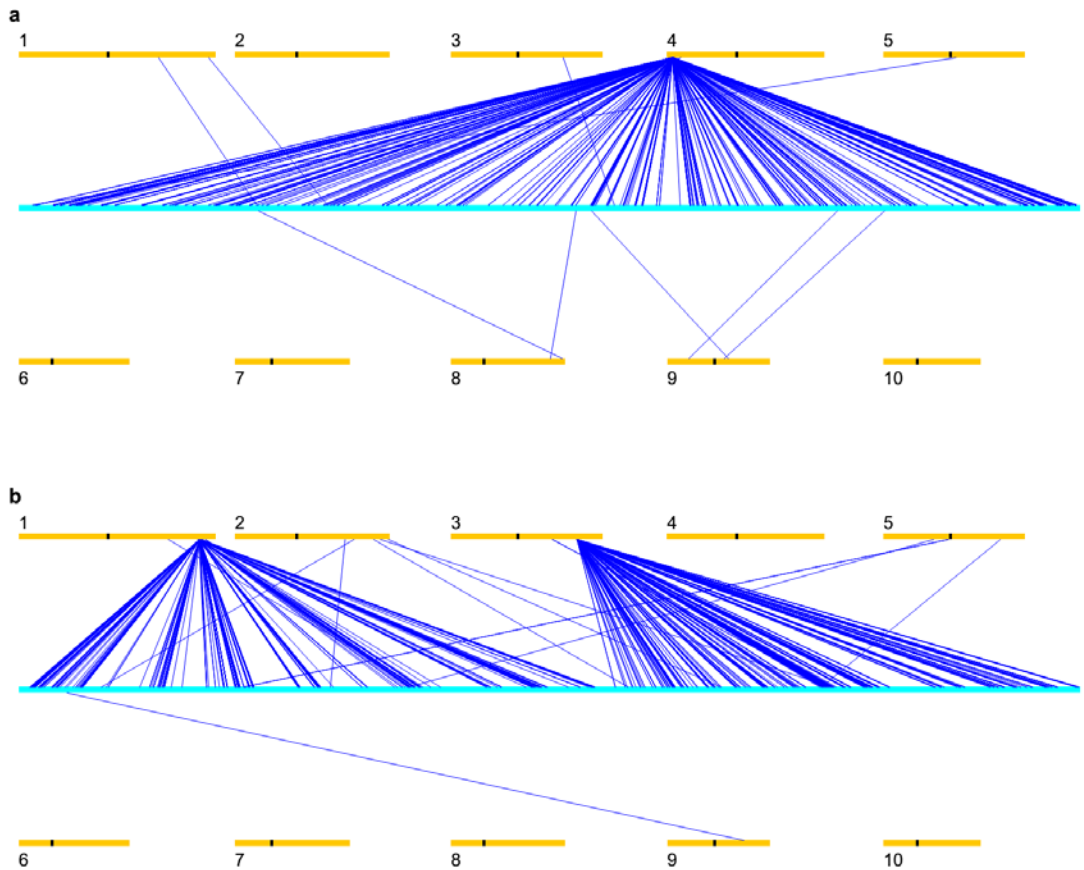
4

BioNano Map

http://www.slideshare.net/kstatebioinformatics/using-bionano-maps-to-improve-an-insect-genome-assembly

Evaluate based on genetic mapping

Use mapped GBS sequence tags to evaluate each contig



Fei Lu, Buckler lab
http://www.nature.com/ncomms/2015/150416/ncomms7914/full/ncomms7914.html

From assembled genome to annotated genome

Procaryotic genomes

↓

Genome annotation servers (web based)

1. RAST
2. NCBI

Eucaryotic genomes

↓

Gene prediction pipeline: Maker

↓

Function annotation pipeline: Blast2GO