Genome Assembly

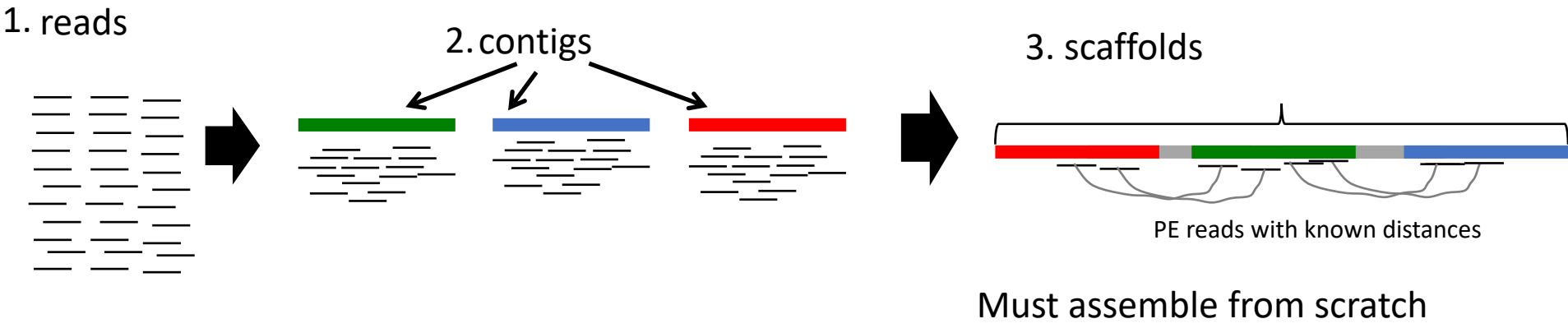
Marc Tollis

Comparative Genomics

as.sem.bly (n) – computational reconstruction of a longer sequence from smaller sequence reads.

Elkblom and Wolf (2014). ***A field guide to whole-genome sequencing, assembly and annotation.*** *Evolutionary Applications*

De novo assembly ('from scratch')



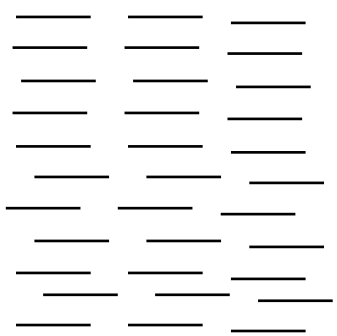
Must assemble from scratch

Reference-based assembly

Map reads with bwa, bowtie etc…

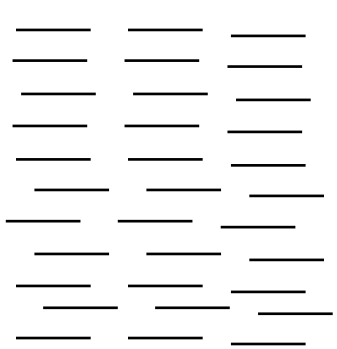
De novo Assembly Basics

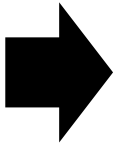
1. Shotgun reads



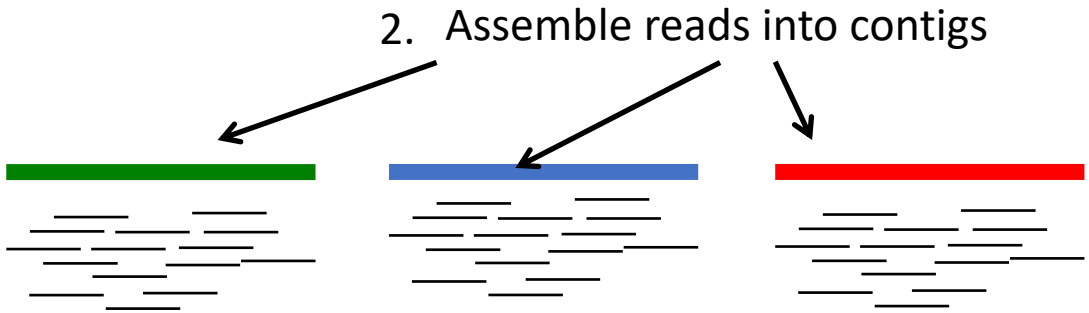
De novo Assembly Basics

1. Shotgun reads



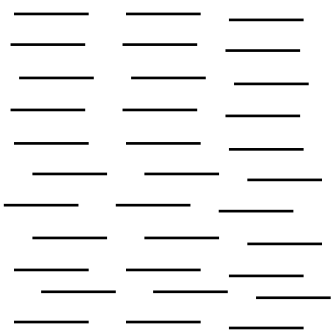


2. Assemble reads into contigs

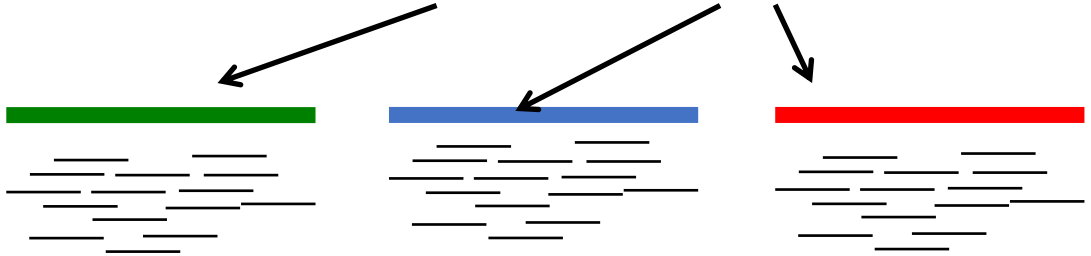


De novo Assembly Basics

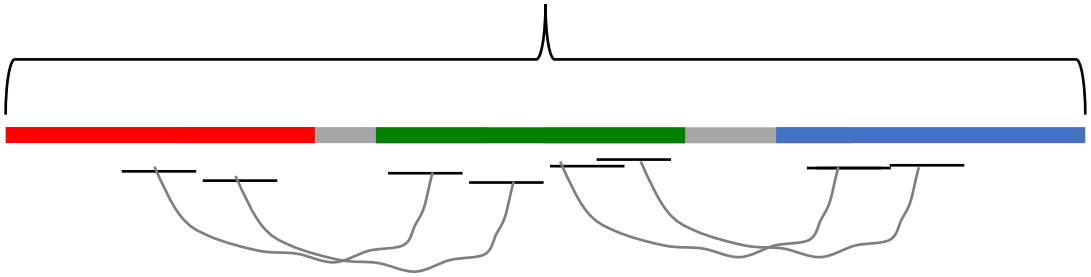
1. Shotgun reads



2. Assemble reads into contigs



3. Order contigs onto a scaffold

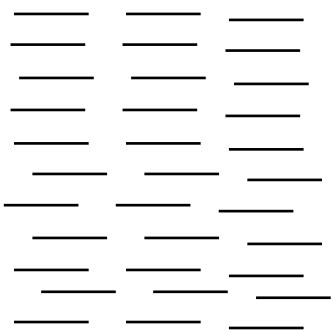


PE reads with known distances

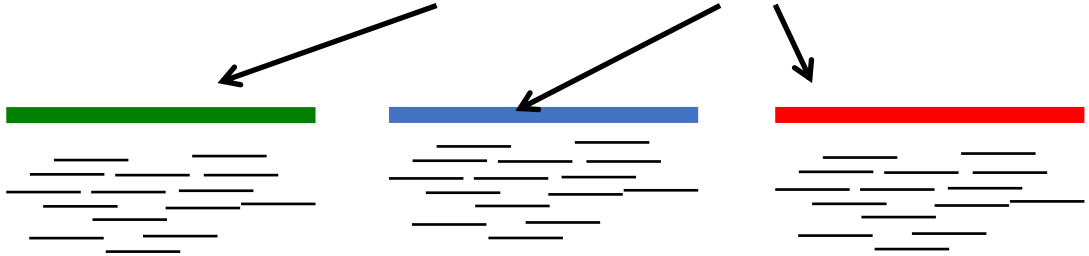
Use paired-end info to determine order of (and distance between) contigs

De novo Assembly Basics

1. Shotgun reads

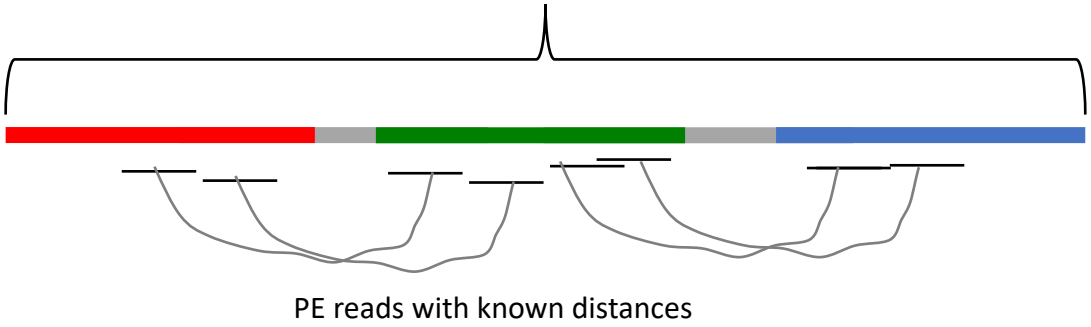


2. Assemble reads into contigs



3. Order contigs onto a scaffold

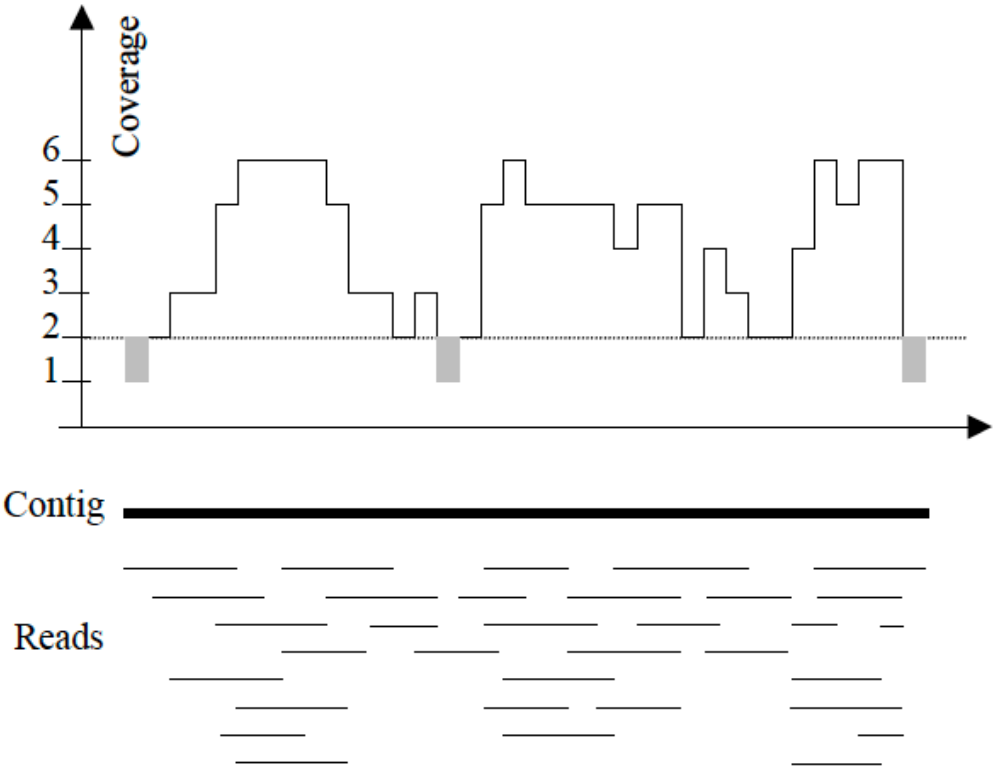
Use paired-end info to determine order of (and distance between) contigs



PE reads with known distances

4. (optional) gaps between contigs are filled in by mapping reads back to the scaffolds

Typical sequencing coverage



Imagine raindrops on a sidewalk

We want to cover the entire sidewalk but each drop costs $1

If the genome is 10 Mbp, should we sequence 100k 100bp reads?

Mike Schatz, CSHL

Core algorithmic concepts: similarity and extension

Identify ***similarity*** by identification of similar seed sequences in different reads

- expensive: each read compared to every other read
- Exact matches are identified quickly
  - but depends on error rate
- FALCON – assembles contigs from PacBio reads

Extension then searches for regions of similarity outside the seed.

- less expensive than similarity search

Core algorithmic concepts: K-mers

Short sequences of length *k* (usually 15-70 nucleotides)

Very cheap

- defined memory footprint
- Depending on number of errors, focuses on exact matches
- Easy to code

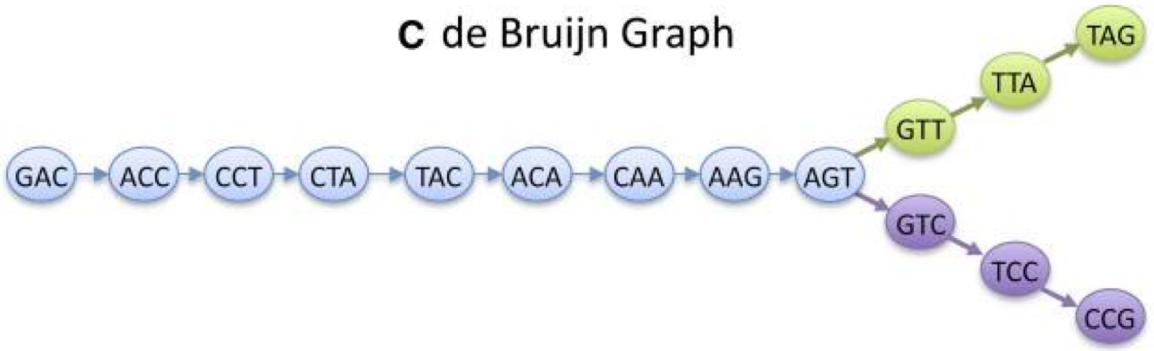
Hash tables are sorted lists of k-mer sequences with a count of how many times the sequence exists

de Bruijn Graph Construction

- Reads are decomposed in to *k*-mers
- *K*-mers become nodes in a graph.
- Edges are drawn between *k*-mers which overlap by *k*-1 bases.
- Non-branching paths in the graph form unambiguous stretches of sequence.

A Read Layout

```
R1: GACCTACA
R2:   ACCTACAA
R3:    CCTACAAG
R4:     CTACAAGT
A:       TACAAGTT
B:        ACAAGTTA
C:         CAAGTTAG
X:       TACAAGTC
Y:        ACAAGTCC
Z:         CAAGTCCG
```



Pop Quiz 1

Assemble these reads using a de Bruijn graph approach (k=3):

ATTA

GATT

TACA

TTAC

Pop Quiz 1

Assemble these reads using a de Bruijn graph approach (k=3):

```
ATTA: ATT -> TTA
GATT: GAT -> ATT
TACA: TAC -> ACA
TTAC: TTA -> TAC
```

Pop Quiz 1

Assemble these reads using a de Bruijn graph approach (k=3):

```
ATTA: ATT -> TTA
GATT: GAT -> ATT
TACA: TAC -> ACA
TTAC: TTA -> TAC
```

Pop Quiz 1

Assemble these reads using a de Bruijn graph approach (k=3):

ATTA: ATT -> TTA

GATT: GAT -> ATT

TACA: TAC -> ACA

TTAC: TTA -> TAC

GATTACA

Mike Schatz, CSHL

Some de Bruijn Assemblers

Canada's Michael Smith Genome Sciences Centre

Beijing Genomics Institute

ALLPATHS-LG

BROAD INSTITUTE

DISCOVAR

Genome variant discovery through assembly

Platanus Kajitani et al., 2014

PLATform for Assembling NUcleotide Sequences

Meraculous – Joint Genomes Institute

MaSURCA – Maryland Genome Assembly Group

ABySS

- Three-step process:
  1. Unitig – single end assembly (no PE)
  2. Contig – contigs using PE
  3. Scaffolding

- Optimize the parameter *k* by running the unitig process as an array
  - i.e., every odd value between 27 and 99
  - Uses Message Passing Interface (MPI)
  - For instance, run the array on 72 cpus across 6 nodes

Assembling contigs using multiple libraries in ABySS

```
abyss-pe k=64 name=ecoli lib='pe200 pe500' \
    pe200='pe200_1.fa pe200_2.fa' pe500='pe500_1.fa pe500_2.fa' \
    se='se1.fa se2.fa'
```

Library pe200 has reads in two files, `pe200_1.fa` and `pe200_2.fa`.

Library pe500 has reads in two files, pe500_1.fa and pe500_2.fa.

Single-end reads are stored in two files, se1.fa and se2.fa.

Scaffolding using ABySS

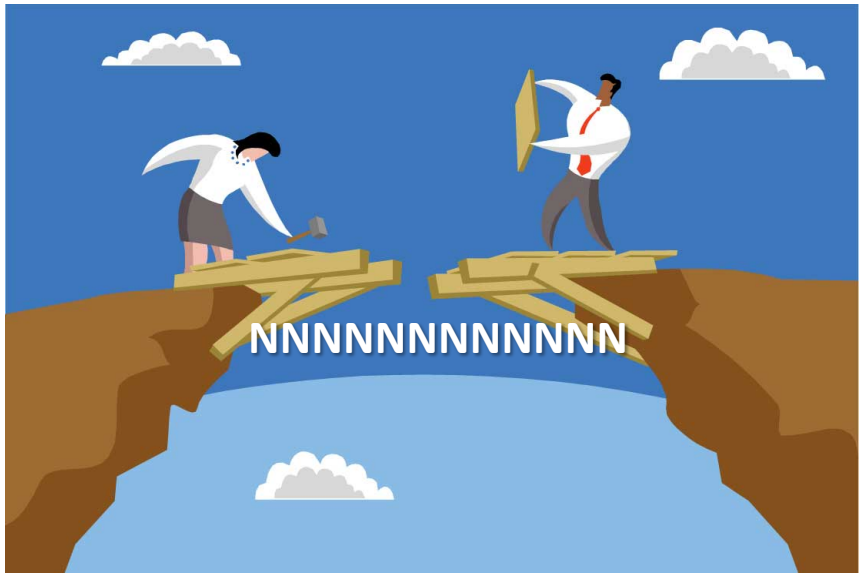
```
abyss-pe k=64 name=ecoli lib='pe1 pe2' mp='mp1 mp2' \
    pe1='pe1_1.fa pe1_2.fa' pe2='pe2_1.fa pe2_2.fa' \
    mp1='mp1_1.fa mp1_2.fa' mp2='mp2_1.fa mp2_2.fa'
```

Mate pairs are used for scaffolding, so do not add to coverage

ABySS will estimate the true distances between reads using it's own mapper

Gap closing

- Every time two contigs are joined on a scaffold, a gap is created (NNN).
- Reads can be mapped to the edges of gaps to close them.
- Memory-intensive process but can reduce Ns in your genome.
  - Hundreds of GB RAM
  - A few days time



- Gap Filler (Boetzer and Pirovano 2012 *Genome Biology)*
- GapCloser from SOAPdenovo
- Sealer (Paulino et al. 2015 *BMC Bioinformatics*)

Rescaffolding

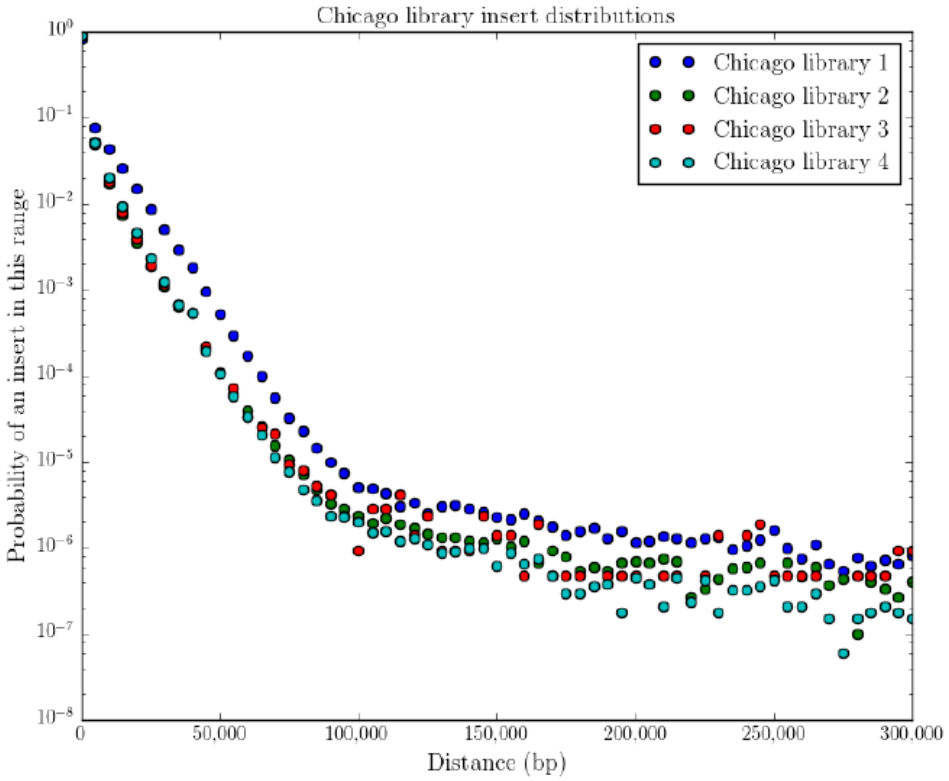
SSPACE (Boetzer et al., 2010)

- Map reads to existing contigs/scaffolds with bowtie
- Position and orientation of each mapped pair is stored in hash
- Read pairs across contigs are joined
- Done iteratively using libraries of increasing size
- RAM-intensive

Best with new data (mate pairs or long reads)

ABySS and SOAPdenovo2 have standalone scaffolders too

Dovetail Genomics



- Chicago Library method – reconstituted chromatin
- Vast insert sizes
- Sequence the libraries at ~95X coverage
- Used to rescaffold Illumina assemblies

Organism	Genome Size (Mbp)	Fold Physical Coverage (in 1-50 kbp bins)	Input N50 (kbp)	Final N50 (kbp)	Fold N50 Improvement
Vampire Bat	2,088	82x	5,498	13,814	3x
Cichlid	845	118x	1,208	3,395	3x
Potato	729	208x	755	5,868	8x
Butterfly	322	55x	143	3,707	26x
Pigeon	1,086	38x	70	3,739	54x
Prairie Chicken	897	111x	136	11,320	83x
Chimp	3,349	88x	72	9,969	138x
Human	3,086	39x	178	26,337	148x
Alligator	2,157	73x	81	21,540	265x

Why Are Some Genomes So Hard To Assemble?

1. Biological reasons:
   - ploidy, heterozygosity, repeat content

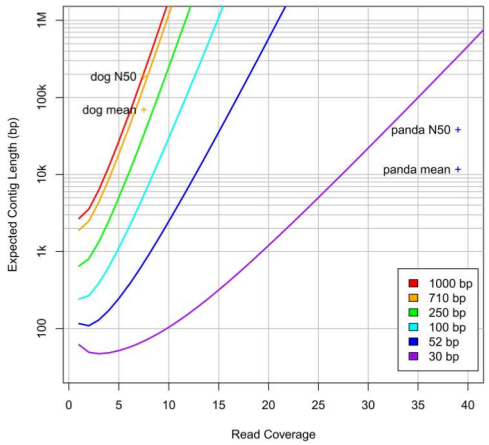
2. Sequencing reasons:
   - large genomes, errors/biases in sequencing

3. Computational reasons:
   - Large genomes, complex structure

4. Accuracy:
   - Hard to assess correctness

Ingredients for a good genome assembly

Coverage



High coverage is required

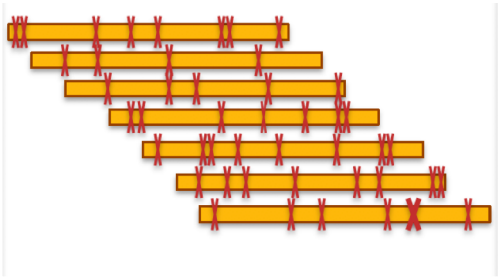
- Make sure every base is sequenced with overlapping reads
- Biased coverage will fragment the assembly

Read Length

Reads and mates must be longer than the repeats

- Short reads will have false overlaps forming problems for assembly graphs
- If reads are long enough, you can assemble entire chromosomes into contigs

Quality



Errors should obscure the overlaps

- Reads are assembled by finding kmers shared in pairs of reads
- High error rates require very short seeds, increasing complexity and forming bubbles and/or other problems

So Is Your Assembly Any Good?

¯_(ツ)_/¯

Contig or Scaffold N50

Most widely used statistic for genome assemblies

Measure of contiguity

Take all contigs and sort them from shortest to longest. The N50 is the length of the contig for which half of the assembly is comprised of contigs at least this length.

More informative than mean

Contig or Scaffold N50

1,1,1,1,1,1,1,1,2,2,3,4,6,6,8,9,9,9,10,24

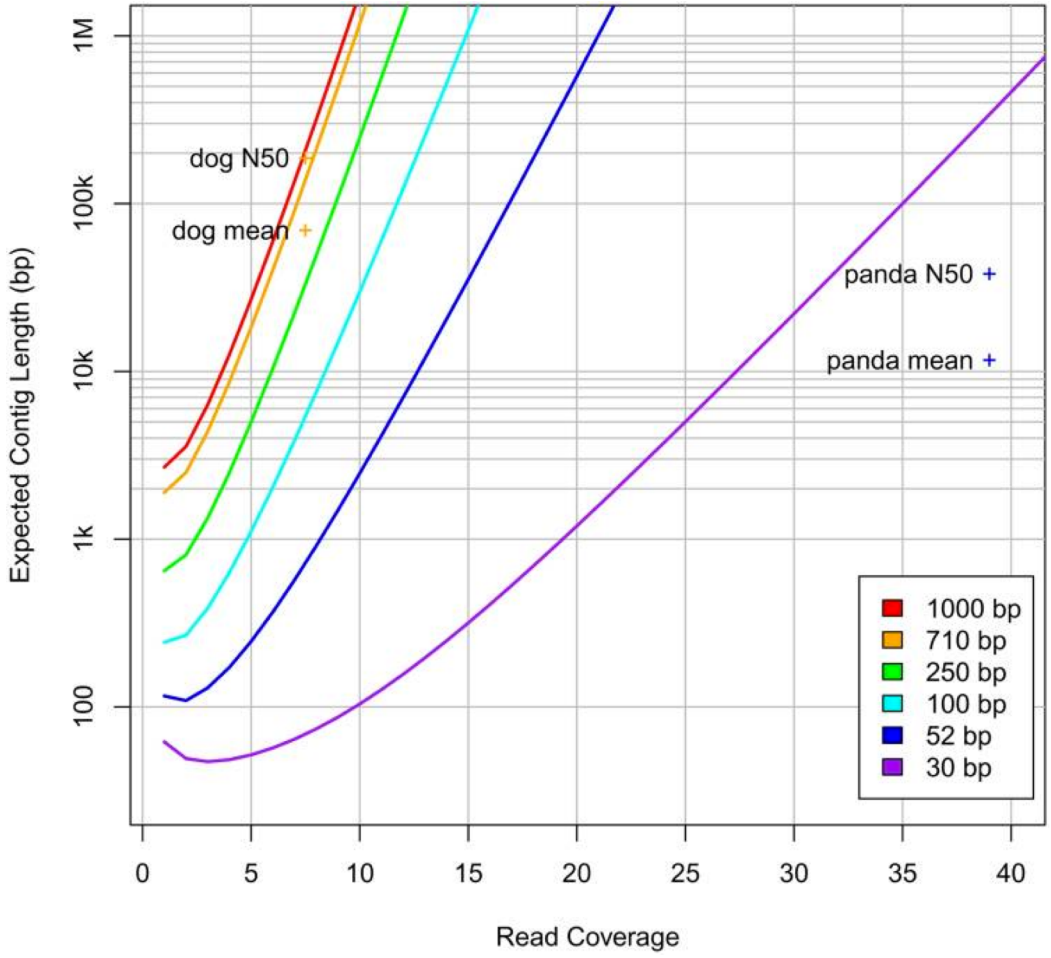
- Total = 100
- Mean = 5
- N50 = 9

N50 can be manipulated if you eliminate small contigs

- Which may be useless anyway

NG50 – uses genome size instead of assembly length

Contig length is a function of read length and coverage



Schatz et al. (2010) *Genome Research*

Assessing Gene Space

Core Eukaryotic Genes Mapping Approach (CEGMA) – Parra et al. (2007)

- Found 458 genes highly conserved across eukaryotes in the eu**K**aryotic **O**rthologous **G**roups (KOG) database
- tblastn of CEGs to your genome
- Refines gene models using HMMs

Proportion of 248 of the most highly conserved single-copy CEGs can be used to estimated how many genes you have in your assembly

Orthologous genes are homologs descended from a common ancestor due to speciation.

Benchmarking Universal Single-Copy Orthologs (BUSCO)

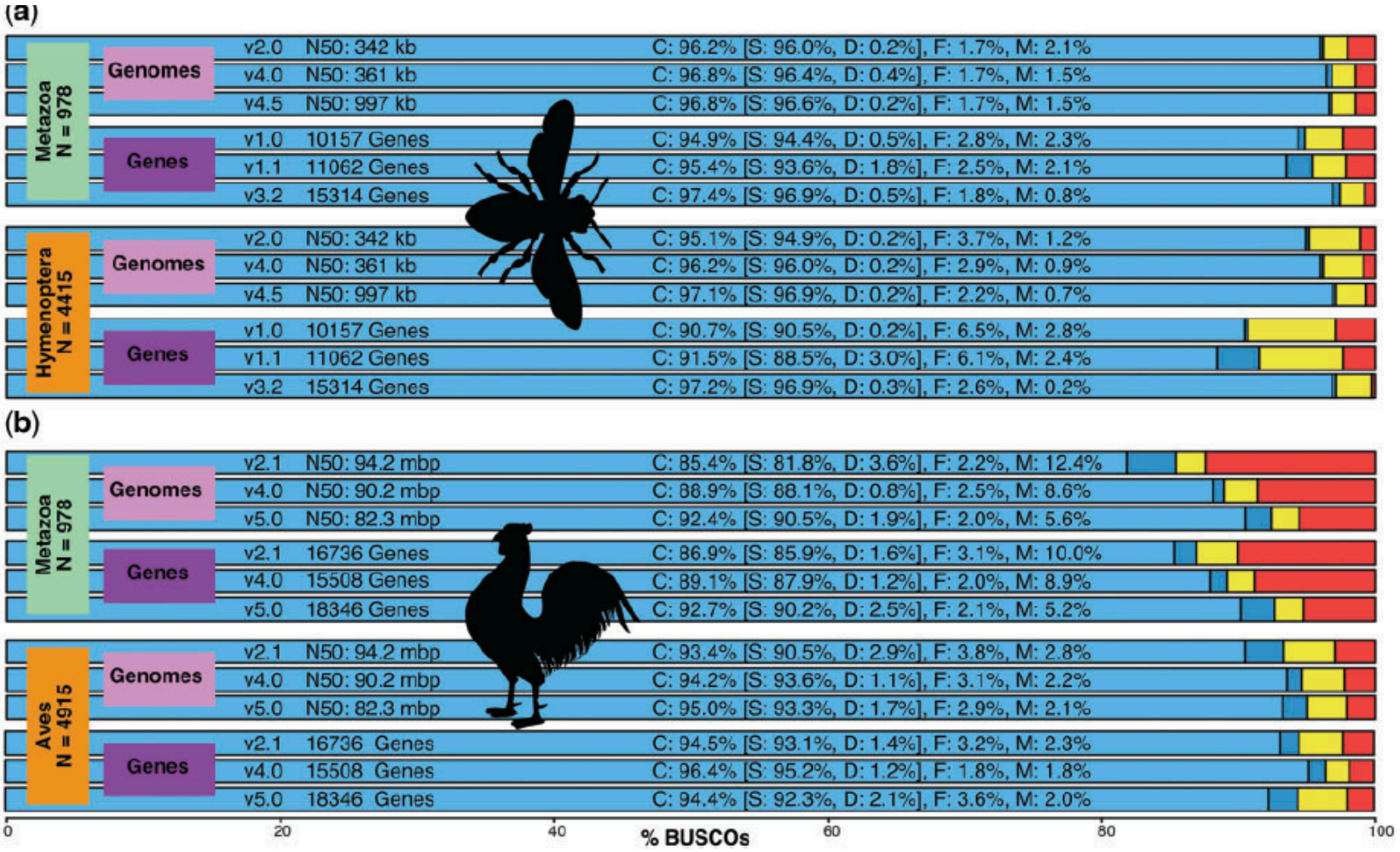
- Similar to CEGMA
- Based on OrthoDB instead of outdated KOGs database
- Clade-specific conserved single-copy orthologs

Computed BUSCO Assessments for Vertebrate Species

Species	Data Type	BUSCO Benchmarks
Human	Genome	C:89% [D:1.5%], F:6.0%, M:4.5%, n:3023
	Gene set	C:99% [D:1.7%], F:0.0%, M:0.0%, n:3023
Mouse	Genome	C:78% [D:3.0%], F:19%, M:2.5%, n:3023
	Gene set	C:99% [D:2.5%], F:99%, M:0.1%, n:3023
Platypus	Genome	C:55% [D:0.8%], F:25%, M:18%, n:3023
	Gene set	C:72% [D:1.1%], F:19%, M:8.2%, n:3023

Felipe A. Simão, Robert M. Waterhouse, Panagiotis Ioannidis, Evgenia V. Kriventseva, and Evgeny M. Zdobnov. 2015. ***BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs.*** *Bioinformatics*

BUSCO Completeness Assessments for Genomics Quality Control



Waterhouse et al. 2018. ***BUSCO Applications from Quality Assessments to Gene Prediction and Phylogenomics.*** *Mol Biol Evol.*

Choose genome, gather info

↓

DNA Library preparation

↓

Sequencing

↓

Quality check

↓

Trimming

↓

Error correction

↓

Merge overlapping reads

↓

ASSEMBLE!

Choose genome, gather info

DNA Library preparation

Sequencing

Quality check

Trimming

Error correction

Merge overlapping reads

ASSEMBLE!

ASSEMBLE!

Assemble again and again (different tools, kmers)

Fill gaps

Evaluate assembly contiguity

Evaluate assembly gene content

Choose a final assembly

Re-scaffold

ANNOTATE!

Choose genome, gather info

DNA Library preparation

Sequencing

Quality check

Trimming

Error correction

Merge overlapping reads

ASSEMBLE!

ASSEMBLE!

Assemble again and again (different tools, kmers)

Fill gaps

Evaluate assembly contiguity

Evaluate assembly gene content

Choose a final assembly

Re-scaffold

ANNOTATE!

Bradnam *et al. GigaScience* 2013, **2**:10
http://www.gigasciencejournal.com/content/2/1/10

(GIGA)^n SCIENCE

RESEARCH Open Access

Assemblathon 2: evaluating *de novo* methods of genome assembly in three vertebrate species

Three vertebrate species

Multiple international teams

Many different assemblers

To answer: which is the best assembler?

The Assemblathon 2 Cast

Budgeriagar
Melopsitaccus undulatus
1.2 Gbp genome

Data:
285X Illumina p.e. & m.p
16X 454
10X PacBio

Lake Malawi cichlid
Maylandia zebra
1.0 Gbp genome

Data:
192X Illumina p.e. & m.p.

Boa constrictor
Boa constrictor
1.6 Gbp genome

Data:
125X Illumina p.e. & m.p.

What they did with the assemblies

Analyses

Contiguity

Presence of Core Genes (CEGMA)

Alignment of fosmids to scaffolds

- COMPASS analysis of validated fosmid regions (VFRs)
- Short range accuracy in VFRs

Optical map analysis

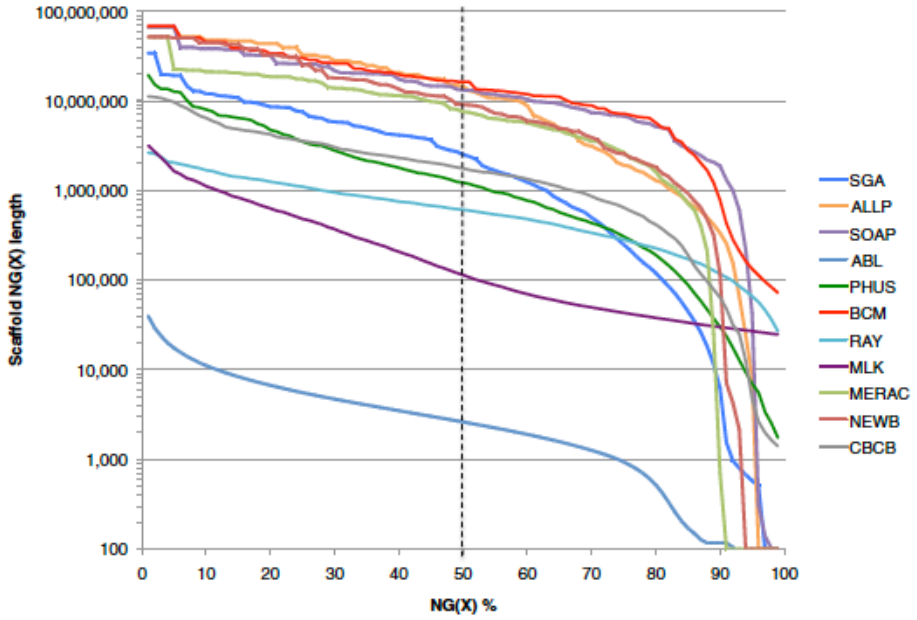
REAPR analysis

These metrics were incorporated

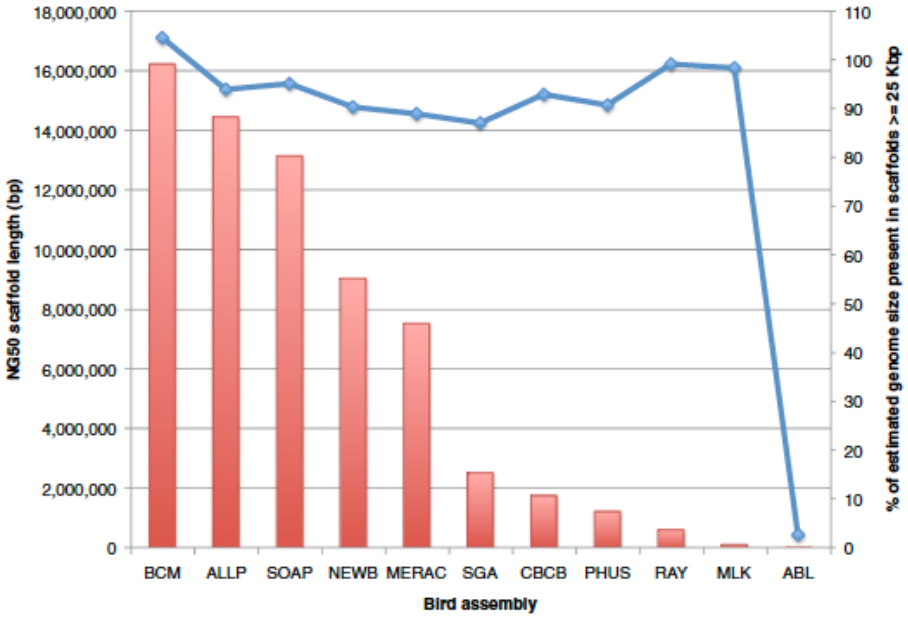
Ranking of assemblies using z-scores

analysis of metrics using linear regressions

Scaffold NGX

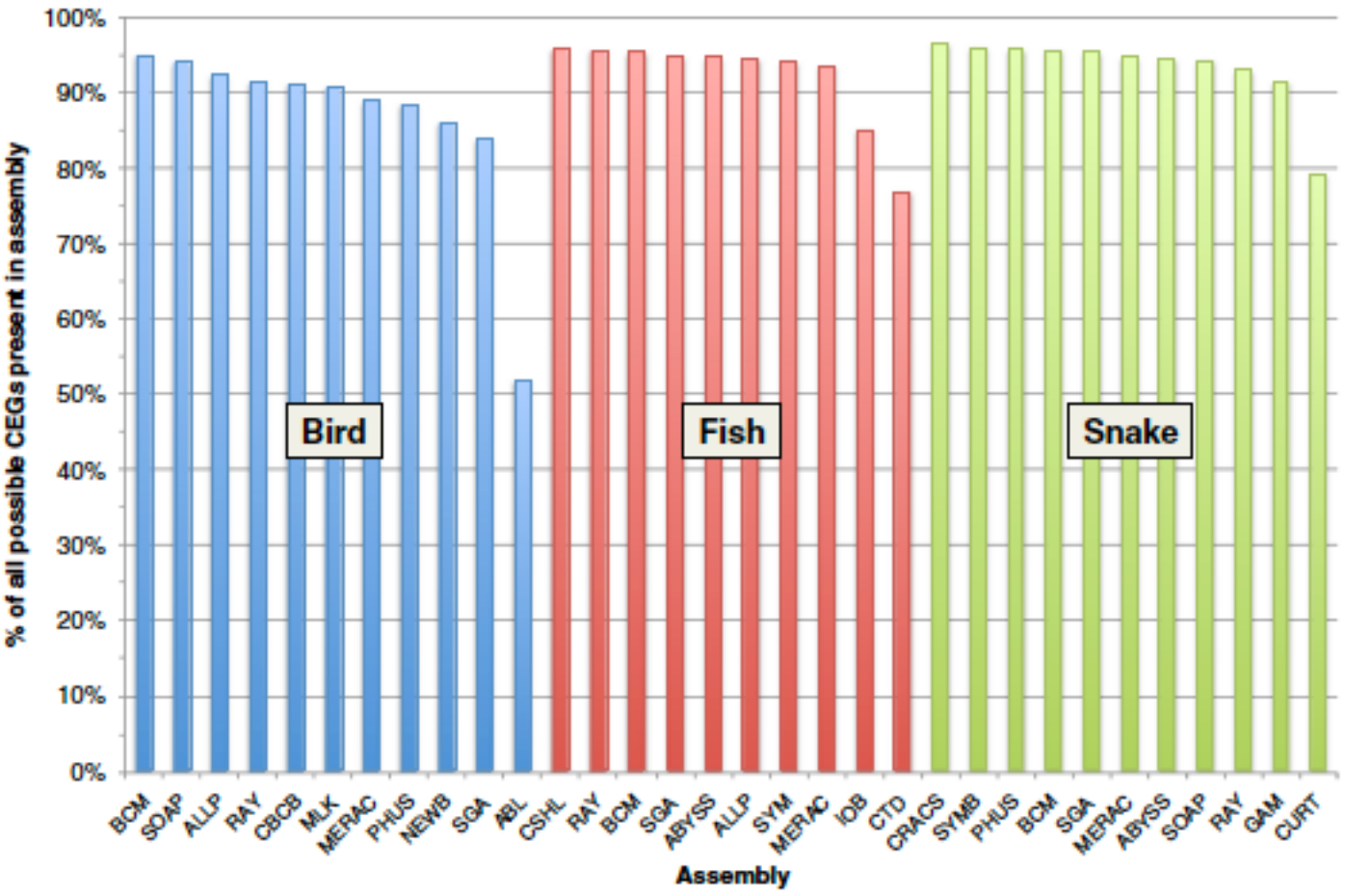


NG50 varied widely

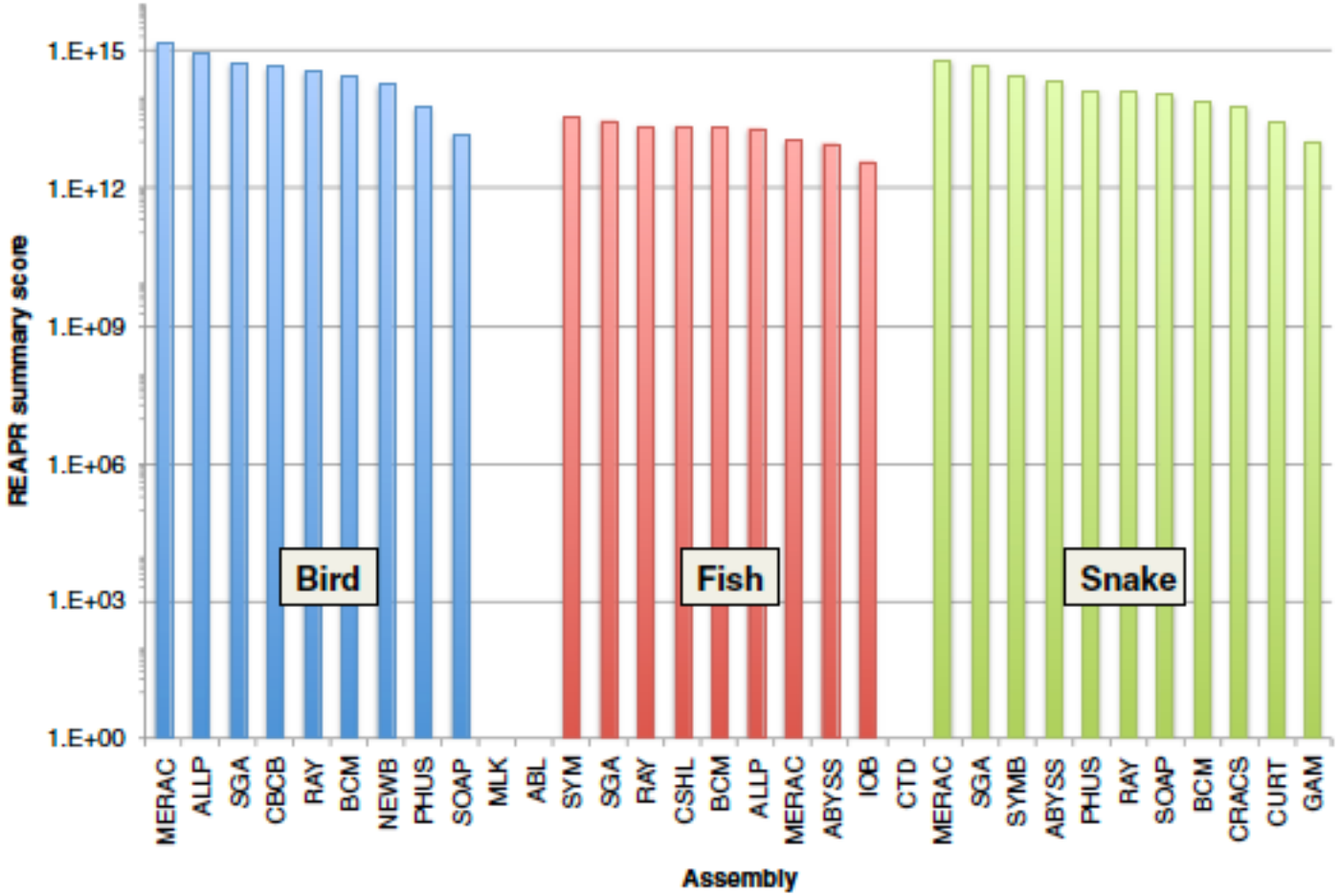


Comparing NG50 to prop. of assembly found on gene-sized scaffolds

Presence of Core Genes

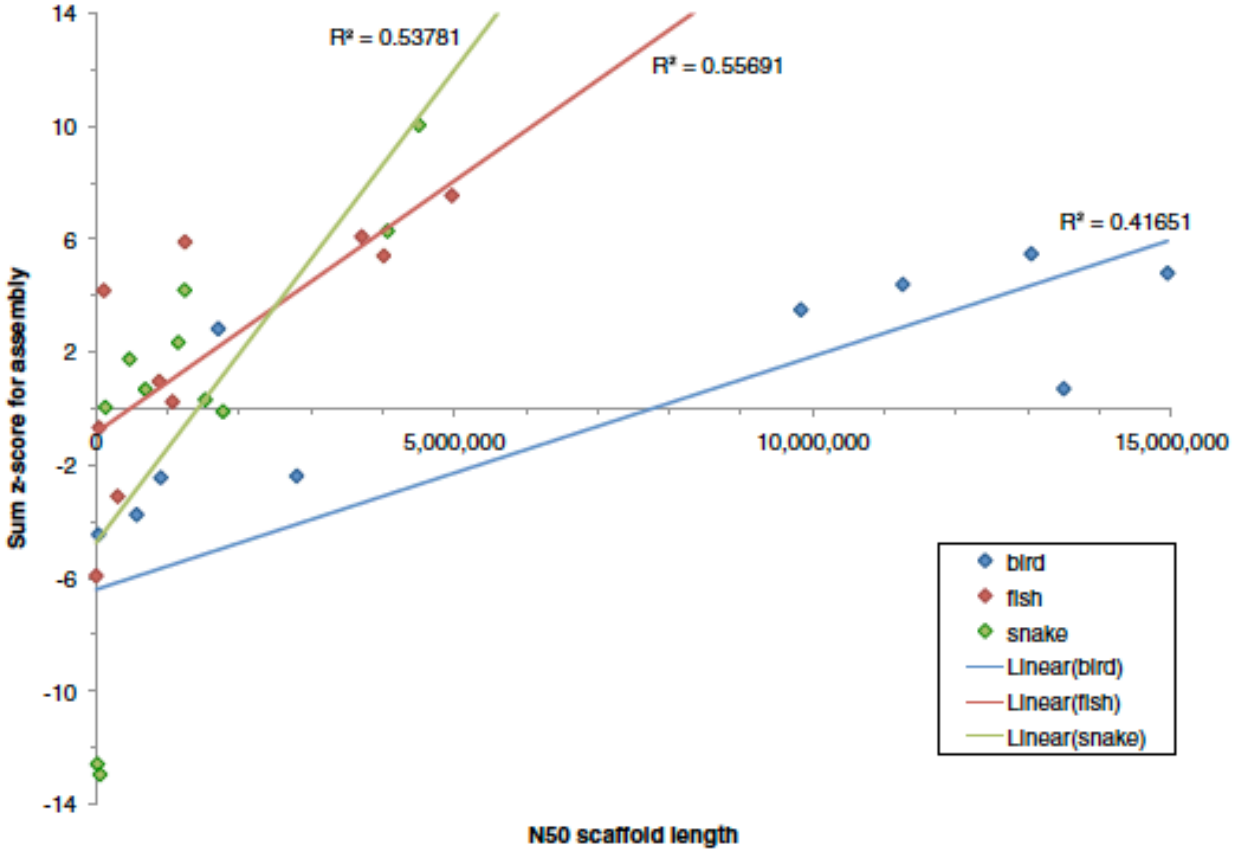


REAPR – remapping paired reads back to assembly to obtain base-by-base metrics



Hunt et al. ***REAPR: A universal tool for genome assembly evaluation.*** 2013. *Genome Biology*

Correlation between z-score ranking and scaffold N50



Bradnam et al. GigaScience 2013, 2:10
http://www.gigasciencejournal.com/content/2/1/10

(GIGA)n SCIENCE

RESEARCH Open Access

Assemblathon 2: evaluating *de novo* methods of genome assembly in three vertebrate species

1. Don't trust the results of a single assembly.
   - Generate several assemblies with different assemblers and/or parameters.
2. Assemblers that work well with data from one species may not work well in another.
3. Scaffold contiguity may not be related to gene coverage.
4. Heterozygosity of target genome is a big issue.
5. Don't place too much trust in a single metric.