THE ASSEMBLATHON

A genome assembly challenge

Keith Bradnam

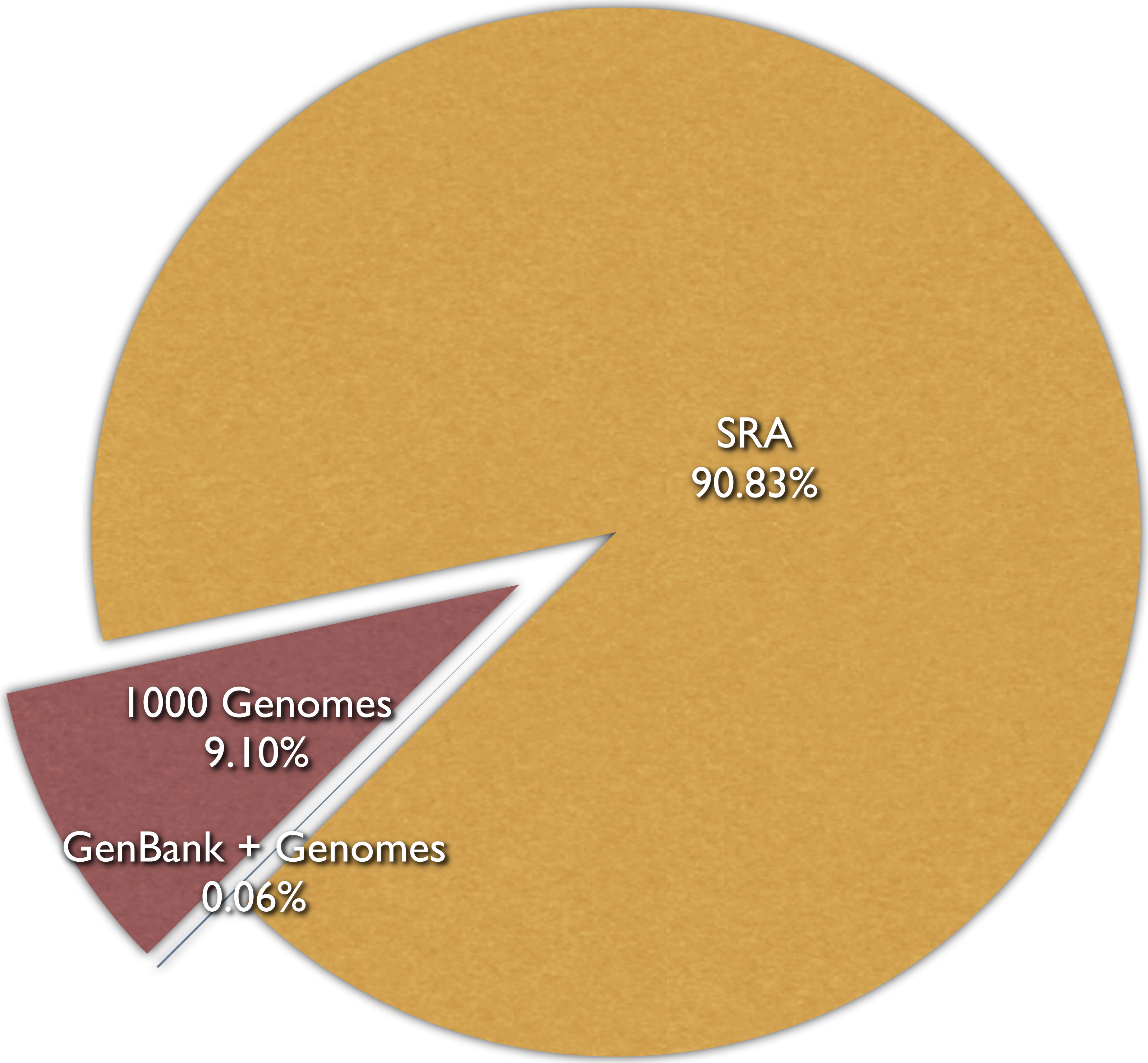
UC Davis Genome Center

assemblathon.org

@assemblathon

Why do we need an Assemblathon?

What's in the NCBI FTP site?



Remember when you used to think that GenBank was 'big'? The NCBI Sequence Read Archive (SRA) now dominates the NCBI FTP site and accounts for 1.14 Petabytes of storage (perhaps explaining why it is to be phased out). In comparison, the Genome and GenBank directories on the FTP site account for less than a tenth of one percent of all sequences. Sequences from 'traditional' sequencing methods now contribute less and less to genome projects. The so called 'short' reads – which are getting longer all the time – dominate the scene.

We have lots and lots of reads, but we want *gene* and *genome* sequences

We need better methods of genome assembly

We also need to define what 'better' means

Projects such as Genome 10K mean that the avalanche of sequence data will continue for many years. But we need to make sense of this data, and most importantly we need to turn sequences reads into genome sequences and sets of gene annotations. The Genome 10K project recognized the importance of genome assembly and the Assemblathon was born out of this. Note that there are other genome assembly assessment projects out there such as dnGASP and GAGE.

The Assemblathon

December 2010 – February 2011

Paper in preparation

The first Assemblathon took started in December 2010, and participants had until February to submit their entries. The following few months have been used to evaluate and assess the assemblies, as well as try to create some new genome assembly metrics.

UC Davis

Genome Center

Ian Korf

Dawei Lin

Keith Bradnam

Aaron Darling

Joseph Fass

Ken Yu

Vince Buffalo

UC Santa Cruz

Haussler Lab

David Haussler

Dent Earl

Benedict Paten

John St. John

Ngan Nguyen

Two groups were involved in the evaluation of the Assemblathon. The results presented in this talk were generated by all of the people mentioned. Note that other people - associated with the Genome10K project - have also played a pivotal role in getting the Assemblathon project up and running.

The challenge

1) Get teams to assemble a genome from next generation sequencing data

2) Evaluate results using existing and newly developed genome assembly metrics

Two main components to the challenge. First, we had to get people to all come together and agree to assemble the same set of genomic reads. The second part of the Assemblathon was perhaps more important. Can we come up with a variety of methods that fairly assess the 'quality' of a genome assembly? Can we also compare those metrics and evaluate which ones work well for different purposes?

The genome

112 Mbp diploid synthetic genome

Created using EVOLVER suite of software

Chromosome sequences can undergo substitutions, indels, translocations etc.

Synthetic genome was created using the EVOLVER suite of software and have an estimated divergence date of 100 million years ago. EVOLVER models many different types of evolutionary change (substitutions, indels, translocations, etc.) and has models for repeat evolution. Synthetic Illumina reads were created from the species A genome. Most groups did not use species B info. The two haplotypes are both separated by ~1 million years of evolution. There is a SNP approx. every 300 nt on average.

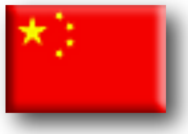
The raw data

Generated synthetic Illumina 100 nt reads using SimSeq

Species A	Species B
Paired ends, 40x, insert size = 200 bp	Diploid chromosome sequences
Paired ends, 40x, insert size = 300 bp	GFF file of gene annotations
Mate pairs, 20x, insert size = 3,000 bp	
Mate pairs, 20x, insert size = 10,000 bp	

SimSeq is software tool written by John St. John – https://github.com/jstjohn/SimSeq. Simulated reads are designed to emulate Illumina HiSeq reads. SimSeq models errors that are typically introduced by paired end and mate pair protocols. Genome coverage of reads is random, which is probably not typical for a real set of reads. Total coverage = 120x (55x per haplotype after excluding bacterial contamination reads).

Participants



17 teams

Teams A–Q

It was fantastic to get so many different teams involved from around the world. All major sequencing centers took part.

Participants

62 assemblies

21 by UC Davis

Teams V–Z

If you want to develop a good assembly metric, it is useful to have both good and bad assemblies to measure. We used three popular assemblers (CLC, Velvet, and ABySS) with the default parameters and then produced several variant assemblies that modified the settings. Some of these 21 assemblies are intentionally bad, e.g. they don't use pairing information for sets of paired reads. Bad assemblies should score badly, so these assemblies provide an extra check on how useful our metrics are.

Metrics

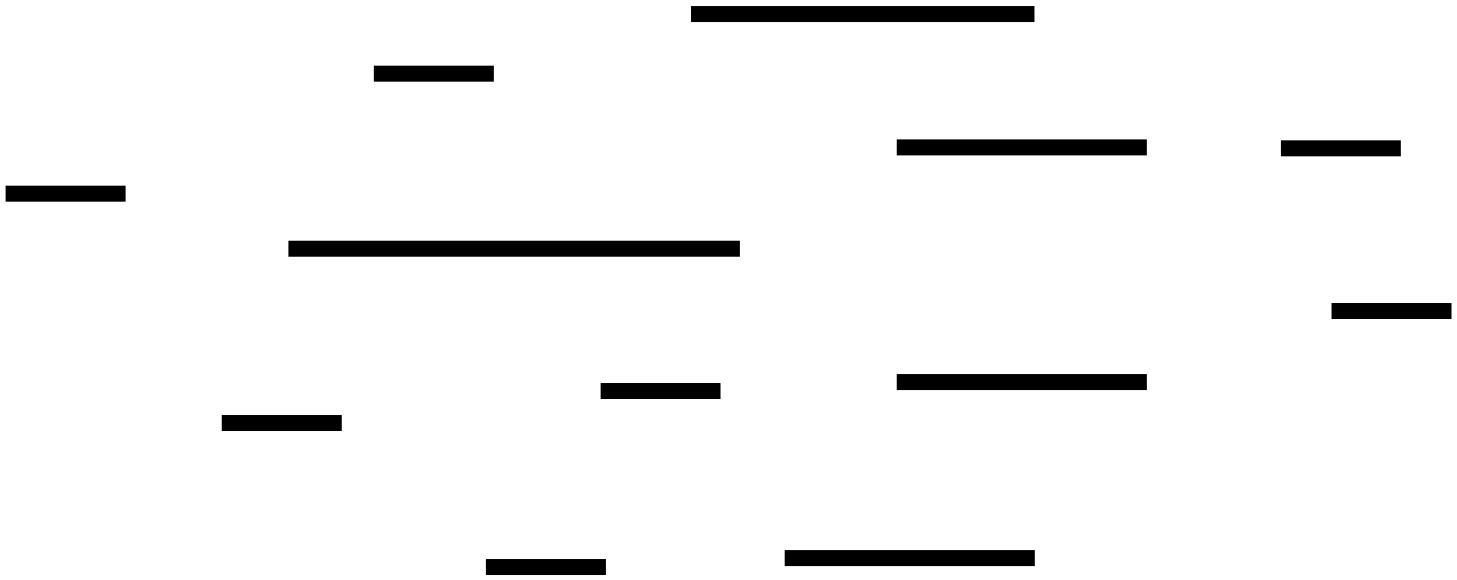
2 main types of metric

1) Traditional statistics such as N50, which don't utilize the known genome sequence

2) Metrics that make explicit use of the known genome sequence, e.g. align scaffolds to genome

Over 150 different metrics were calculated, many of which were calculated independently by the two evaluation teams. This was to ensure that we were getting the same answers. Because we know what the Species A genome looks like, we can use some metrics which would otherwise not be possible to use.

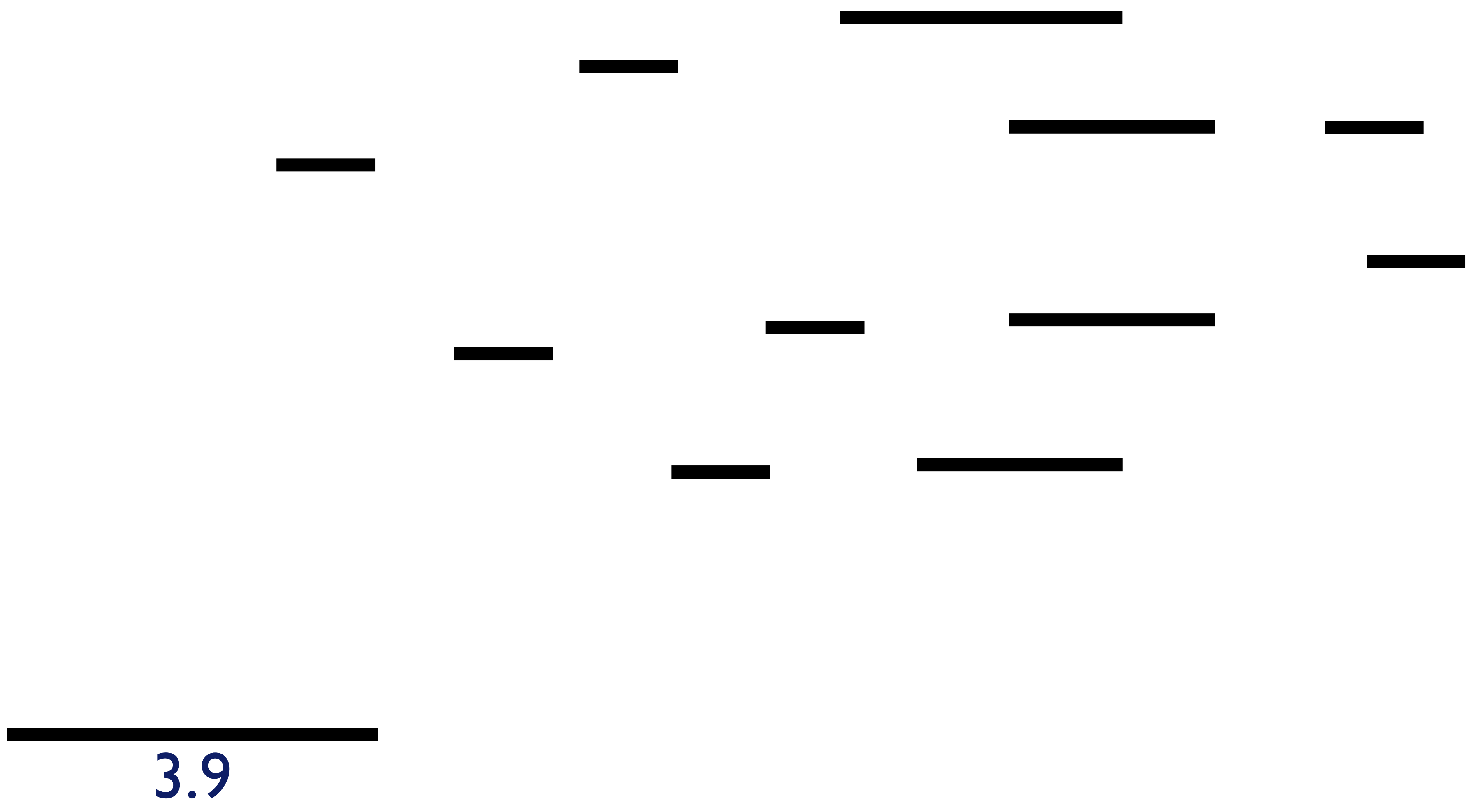
N50



Assembly size = 20 Mbp

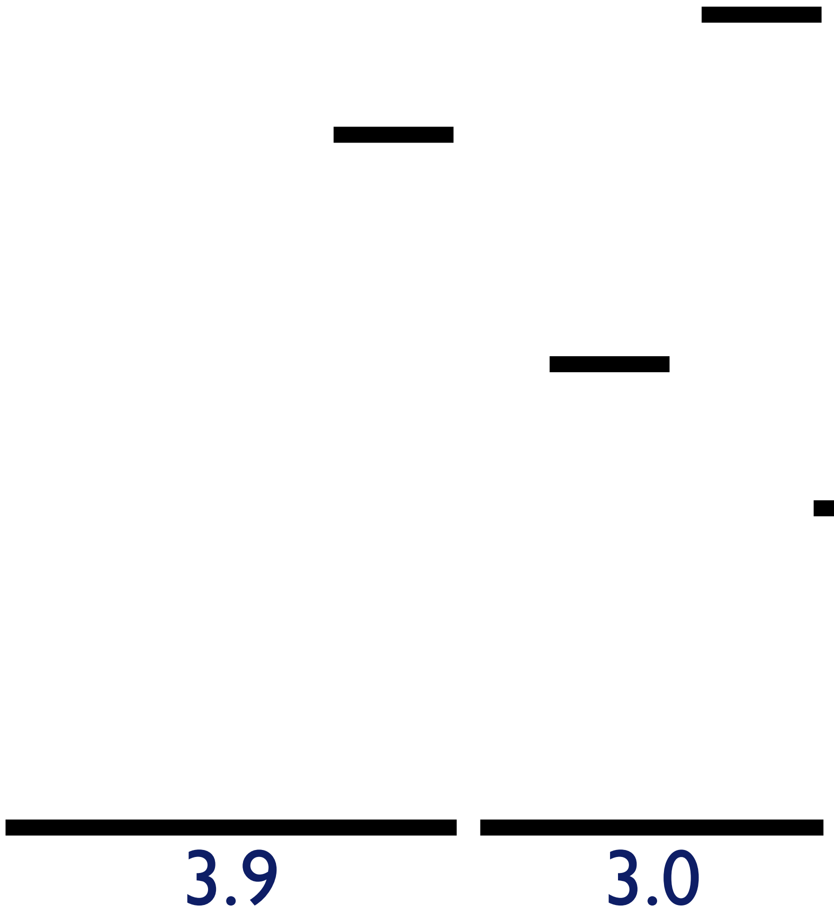
A reminder of the most traditional assembly metric. N50 is way of calculating the average length of a set of sequences. This measure can be biased if assemblies choose to exclude many short sequences from your assembly. It also can not be fairly compared to other assemblies if assemblies are different in size. In the Assemblathon, the submitted assemblies varied in size from ~80% – 200% of the known (haploid) genome size.

N50



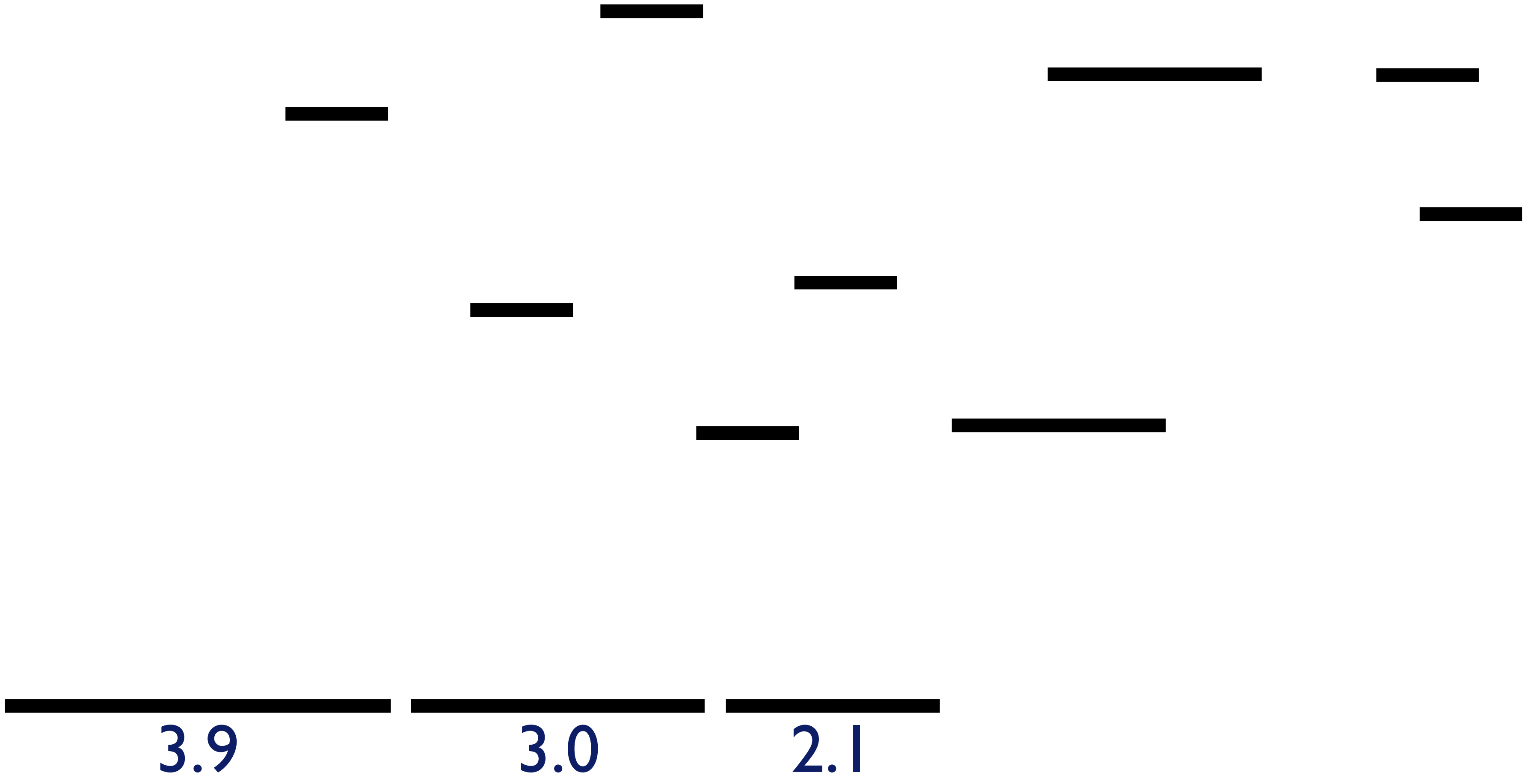
N50 can be calculated from any set of length measurements. First start with the longest sequence (be it a contig, scaffold, or any aligned length).

N50



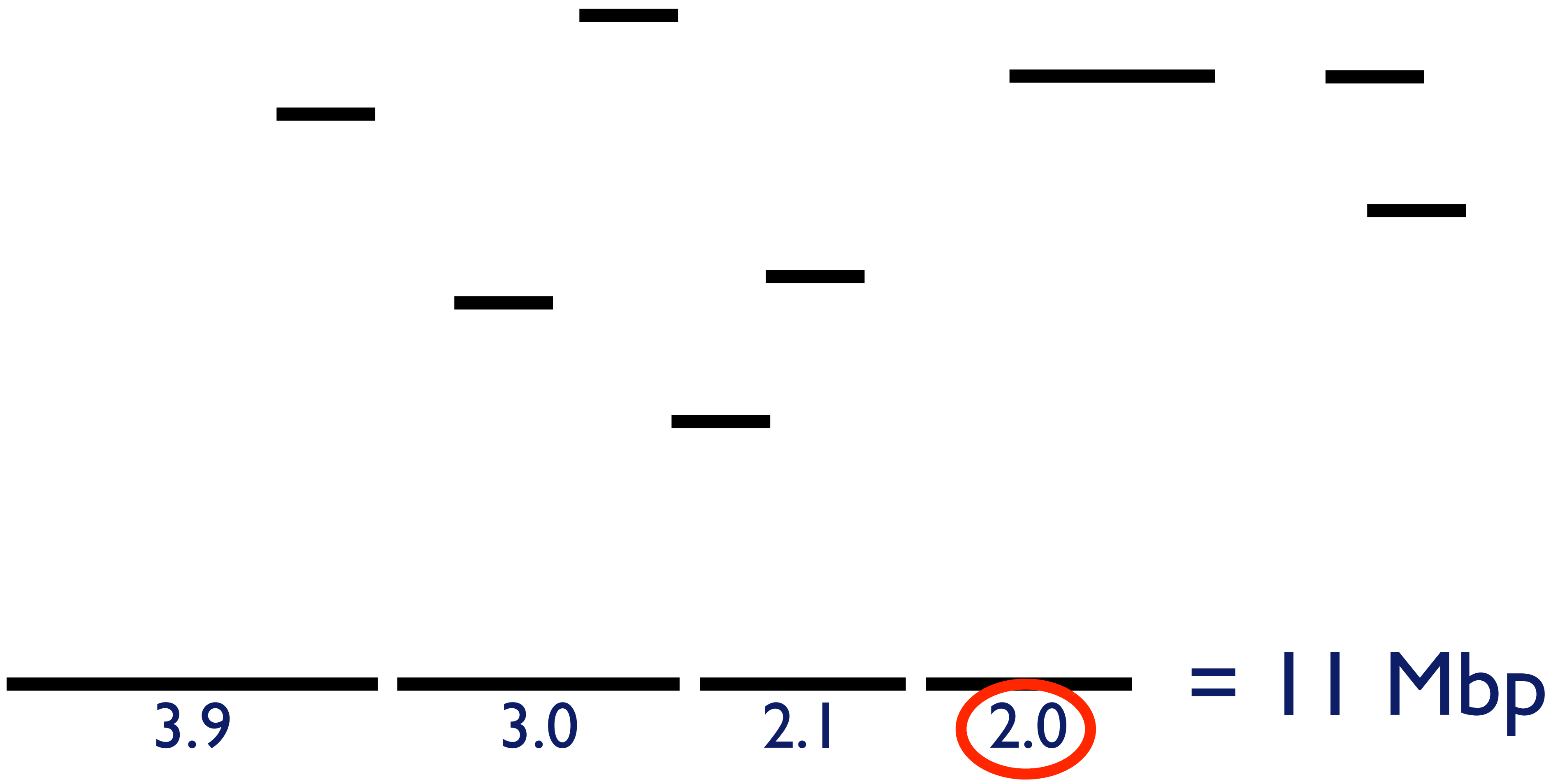
Then take the next longest sequence and add it’s length to the previous sequence (sum = 6.9)

N50



Keep adding the lengths of progressively shorter sequences (sum = 9)

N50



A point will be reached where the sum length is equal to or greater than 50% of the total assembly size. The length of the sequence that takes you pass this threshold value is the N50 length. You can also calculate lengths at other fractions. E.g. N40 would be the length of the sequence which takes you past 40% of the assembly size. Note that it can be misleading to compare N50 values for assemblies which are very different in size.

Can calculate N50 for many things

N50 talk length time = 18:2 minutes

You can even calculate the N50 value for a set of measurements such as ‘length of talks at Biology of Genomes meeting’. This result can be read as ’50% of the total talk time was made up by talks that lasted 18.2 minutes or less’.

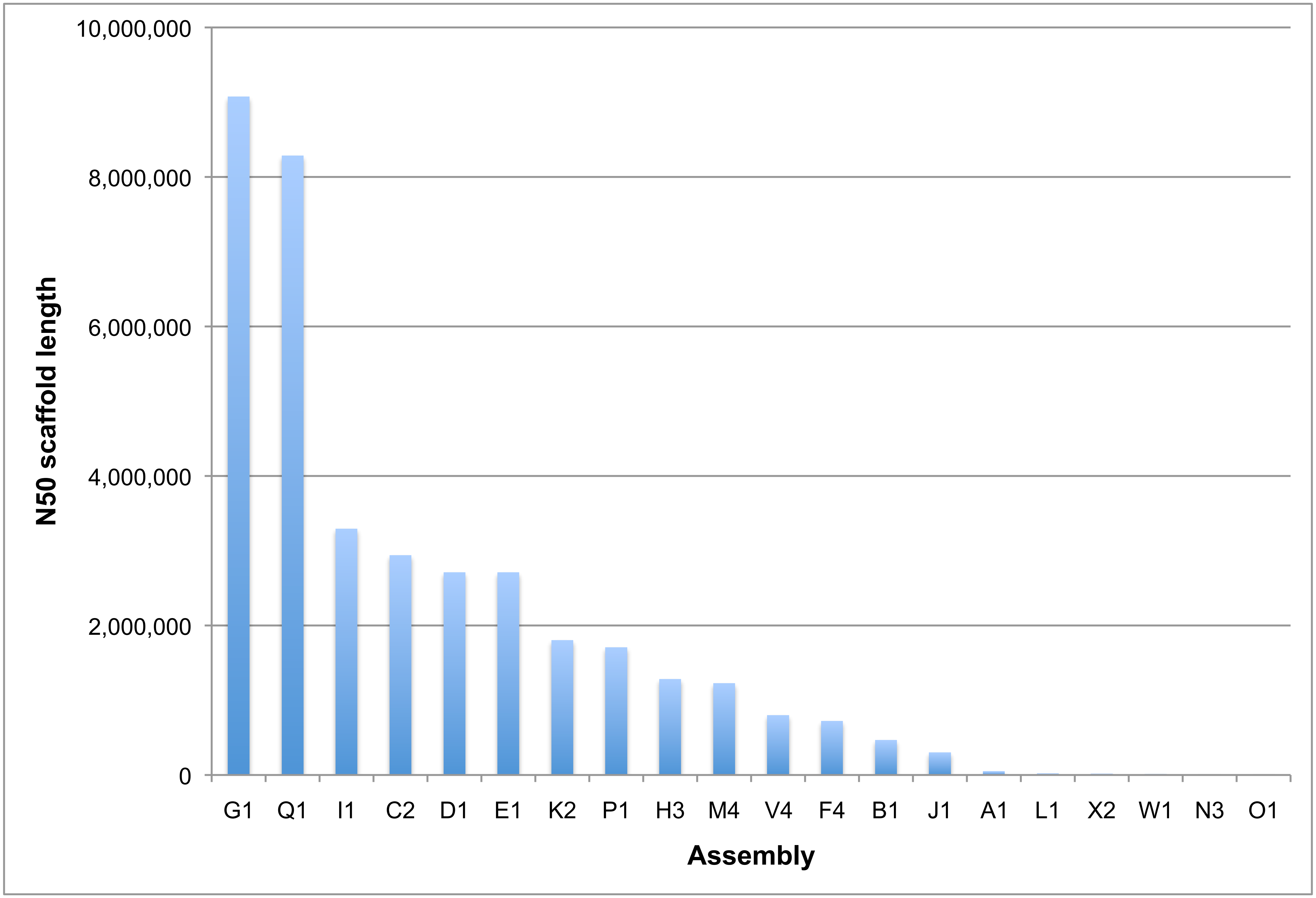
Results

Reminder, these results make up only a tiny fraction of all of the results that were generated.

```
                                         Number of scaffolds       4143
                                     Total size of scaffolds  114258553
    Total scaffold length as percentage of known genome size     101.6%
                                            Longest scaffold    4397505
                                           Shortest scaffold        100
                                Number of scaffolds > 500 nt        905  21.8%
                                 Number of scaffolds > 1K nt        316   7.6%
                                Number of scaffolds > 10K nt        125   3.0%
                               Number of scaffolds > 100K nt        116   2.8%
                                 Number of scaffolds > 1M nt         42   1.0%
                                          Mean scaffold size      27579
                                        Median scaffold size        175
                                         N50 scaffold length    1707112
                                          L50 scaffold count         24
                                        NG50 scaffold length    1716225
                                         LG50 scaffold count         23
              N50 scaffold - NG50 scaffold length difference       9113
                                                 scaffold %A      29.83
                                                 scaffold %C      19.94
                                                 scaffold %G      19.95
                                                 scaffold %T      29.86
                                                 scaffold %N       0.42
                                         scaffold %non-ACGTN       0.00
                             Number of scaffold non-ACGTN nt          0

                Percentage of assembly in scaffolded contigs      93.9%
              Percentage of assembly in unscaffolded contigs       6.1%
                      Average number of contigs per scaffold        1.1
Average length of break (>25 Ns) between contigs in scaffold        115
```

N50 scaffold lengths



Scaffold N50 lengths varied tremendously. In this case, we show only the best assembly from each team, and there were orders of magnitude difference between the highest and lowest values. If we didn't do any other analysis, we would simply conclude that assembly G1 was the 'best' based on this statistic.

Paired fragment analysis

Randomly choose *pairs* of 100 nt fragments from the known genome and search against assemblies

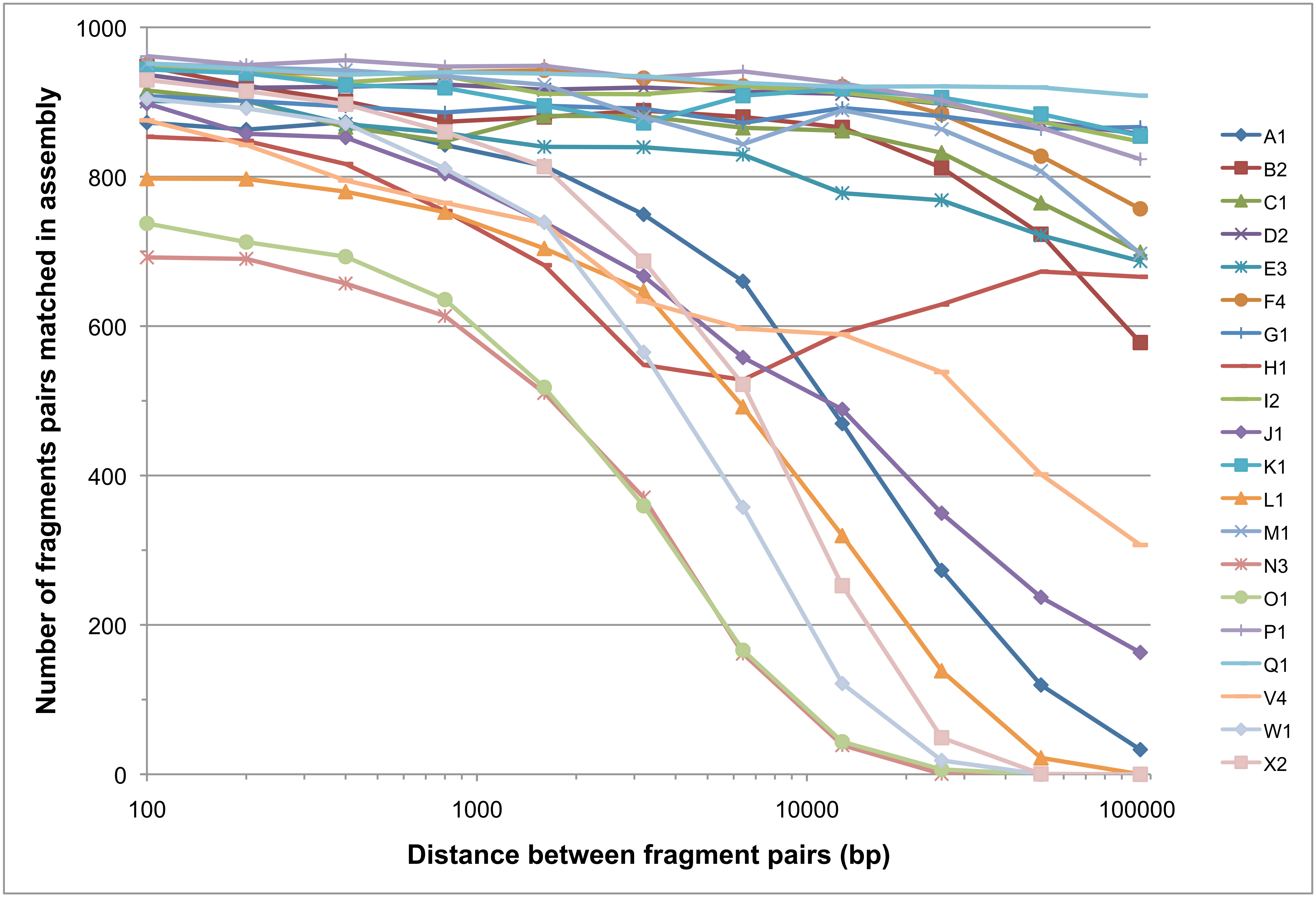
Use BLASTN, 95% identity over 95% length

Both fragments much match same strand of same scaffold

Repeat with different distances between pairs

We perform this analysis at various distances between the fragments. Can then plot the number of matching fragments as a function of the distance between fragment pairs. Fragments in the assembly must be separated by 95–105% of the distance between the pairs in the known genome. This is a quick method which makes it easy to visually compare lots of assemblies at once. It also is a great discriminator between different assemblies. We calculate results using the haplotype 1 and 2 genomes separately, and then average the results.

Paired fragment analysis



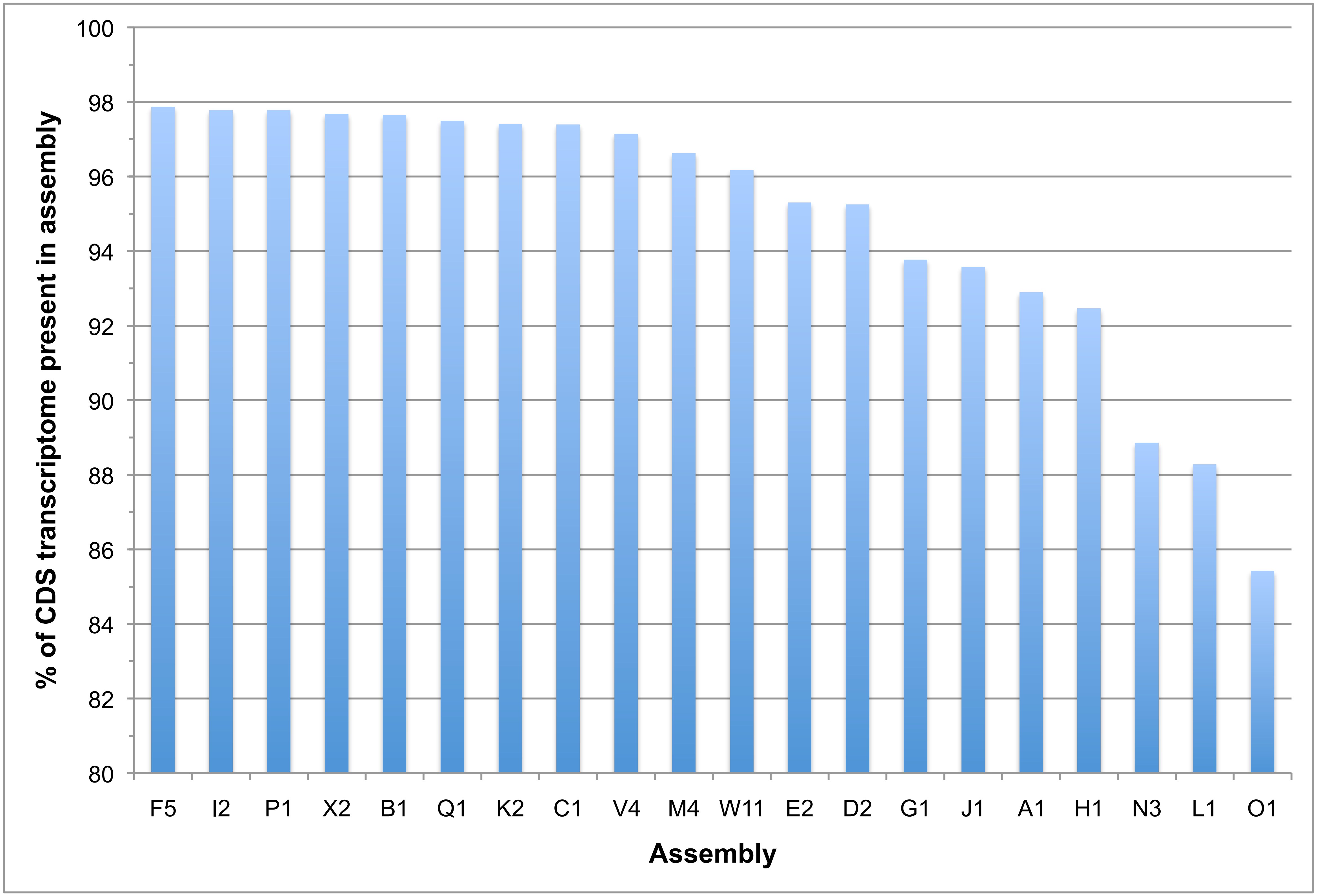
At very long distances between the pair of fragments (e.g. 100,000) there are still assemblies such as Q1 which do an excellent job at containing most matched pairs. Assembly H1 was anomalous in that the performance started to get better at increasing distance (at around D=10,000). If the graph was extended to even larger distances, the H1 assembly results start to deteriorate with increasing distance.

Gene finding

How many genes are present in each assembly?

The Species A genome contains 176 genes. For many end users of a genome assembly the genes might be all they care about, and it doesn't matter how long contigs are...as long as they contain a full-length gene.

Gene space



We used BLAST to search the CDS sequences of each gene (coding exons only, start codon to stop codon) against scaffolds of each assembly. Required any HSP match to show 95% identity. Exons from the genome were allowed to be split across multiple regions in each assembly. This suggests that most assemblies capture the CDS-based transcriptome. However, results are drastically lower when you try to find full length genes (with introns). This is because introns exhibit much more polymorphism between haplotypes (exons are typically 100% identical).

Alignment metrics

We have many, many results that all stem from aligning contigs and scaffolds of the assemblies against the known genome (or ‘genomes’ as we use both haplotypes). It’s not possible to accurately reflect the breadth of this analysis in a 15 minute talk. Many of these results are already available on the Assemblathon website and more will be available in the published paper that is currently being prepared.

Mauve, BWA, & Cactus

1) We want to know about the *accuracy* of aligned sequences

2) We want to know about the *coverage* of the Species A genome

We used three different methods independently of each other. These methods help detect gaps, errors, erroneous intra- and inter-chromosomal joins, insertions, deletions etc. They also tell us how much of the Species A genome is present in each assembly.

Dealing with polymorphism

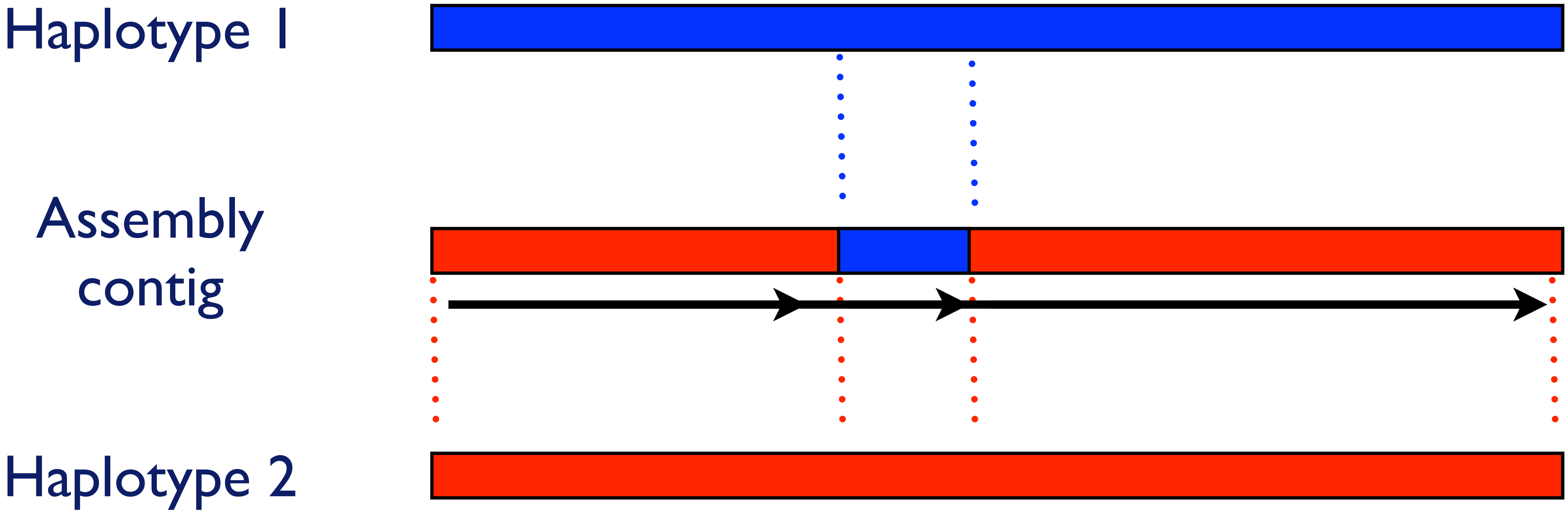
Assembles have contigs that may contain sequence from both haplotypes

Such contigs may not completely align to either haplotype sequence

We define ‘contig paths’ as portions of assembled sequence that are consistent with either haplotype

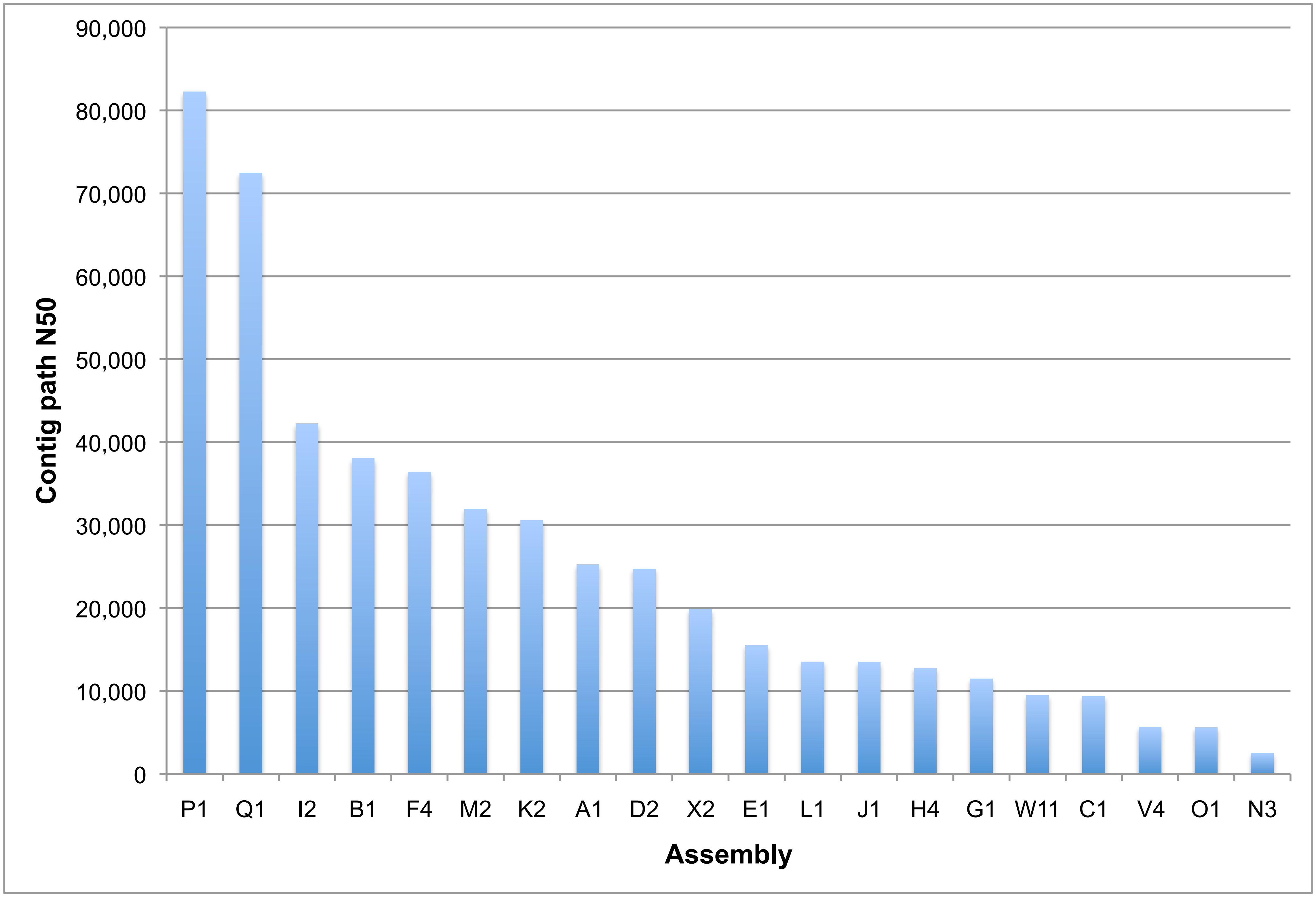
A big problem for assemblers is dealing with polymorphism between haplotypes. A contig sequence in an assembly will typically contain a mixture of sequences from both haplotypes. This means that when you align that contig to both haplotypes, there will typically be at least one break-point. We wanted to not penalize assemblies which essentially just flip-flopped between haplotypes.

Building contig paths



A contig (or scaffold) may contain a mixture of sequences from either haplotypes of the Species A genome. Because of polymorphisms between haplotypes, alignments methods might break these into separate blocks, even though the assembly is essentially correct. Contig paths try allowing for such haplotype variation. In this example, rather than produce 3 separate blocks, we would produce 1 contig path. Contig paths also allow for indels & inversions in any one haplotype. Can calculate an analogous measure for scaffold paths.

Contig path N50



We can calculate a N50 measure for the set of contig path lengths. This measure is very useful for discriminating between different assemblies. If you calculate the N50 value for the aligned blocks which are *not* allowed to switch between haplotypes, then this result is often at least an order or magnitude lower. We also calculated a similar measure for ‘scaffold path N50’. These results to do not always correlate with ‘contig path N50’ suggesting that the process of building scaffolds from constituent contigs differs a lot between different assemblies.

Winners

It is hard to pick one single winner and it would be possible to single out particular metrics which produce winners for most of the teams. Teams, C, D, E, F, G, L, M, N, O, P, and Q all come out as the ‘best’ assembly by at least one metric.

Best of the best (for synthetic genomes)

Team	Assembler	Affiliation
P	SOAPdenovo	BGI
Q	ALLPATHS	Broad Institute
D	SGA	Wellcome Trust Sanger Institute

These were the three assemblies that were consistently among the best by whatever metric was used. Note though that there were no assemblies that ranked first by the majority of metrics. It was far more common to see good assemblies consistently appear in the top 5 rankings of any particular metric. Having said that, even these assemblies would drop outside the top 10 based on certain metrics.

Which assembler should I use?

The one that is most appropriate for your time, resources, and needs.

Some assemblers better at capturing genes than others. Equally, one assembler may give you longer scaffolds, fewer errors, higher coverage, be more tolerant of repeats etc. No easy answer (unfortunately).

Trade offs

	Coverage rank	Substitution error rank
Team P	1	9
Team D	8	1

You can’t always have your cake and eat it. The assembler that was best at producing a high coverage of the Species A genome was not so good in terms of the number of substitution errors in the assembly. Conversely, the assembly that was the best at minimizing substitution errors, was not so good at producing a high coverage.

Assemblathon 2

June 1st – September 1st

The next Assemblathon is due to start very soon. Keep an eye on the Assemblathon website, or joint the Assemblathon mailing list to keep on top of the latest news.

Real data!

Species	Type of sequence	Provider
	Illumina	Illumina
	Illumina	Broad Institute
	454	Erich Jarvis (Duke)
	Illumina	BGI

Assemblathon 2 will consist of real sequence data for three different vertebrate genomes. Participants will be able to choose which genomes they wish to try to assembled.

Evaluation tools

More reads

from Pacific Biosciences

Optical maps

from David Schwartz – U. Wisconsin

Fosmid sequences

from Jay Shendure – U. Washington

Plus other methods

A NIH funding request has been submitted to cover Assemblathon 2 costs. It is likely, though not certain at this stage, that we will receive PacBio reads, optical maps, and fosmid sequences to assist in the validation of the genome assemblies. This is important because – unlike Assemblathon 1 – we will not know what the real answer is. Other evaluation methods will include holding back on some of the initial Illumina/454 reads, using any available transcript data, and utilizing a set of 'core genes' that we believe should be present in all eukaryotic genomes (see Parra, Bradnam, et al., 2009).

The best assembler of today may not be the best assembler of tomorrow

Assemblathon 3

2012

We would like the Assemblathon to become a regular – possibly annual – event. We believe there are many more areas of genome assembly to explore. More importantly, it is likely that the landscape of sequencing technologies will continue to change, and therefore there will be a need to assess new methods as and when they come to market. An NIH grant is currently being written that will try to secure funding for future Assemblathons.

Assemblathon ToDo list

Assemble transcriptomes

Assemble metagenomes

Assemble cancer genomes

Many people are interested in the assembly of other things apart from genome sequences. These are all valid areas which could be covered by future Assemblathons.

Summary

What have we learned?

It’s possible to assemble a genome to a high level of coverage and accuracy

Large differences exist between assemblies

Good assemblies tend to score well across most, but not all, metrics

Heterozygosity is a big problem for assemblers

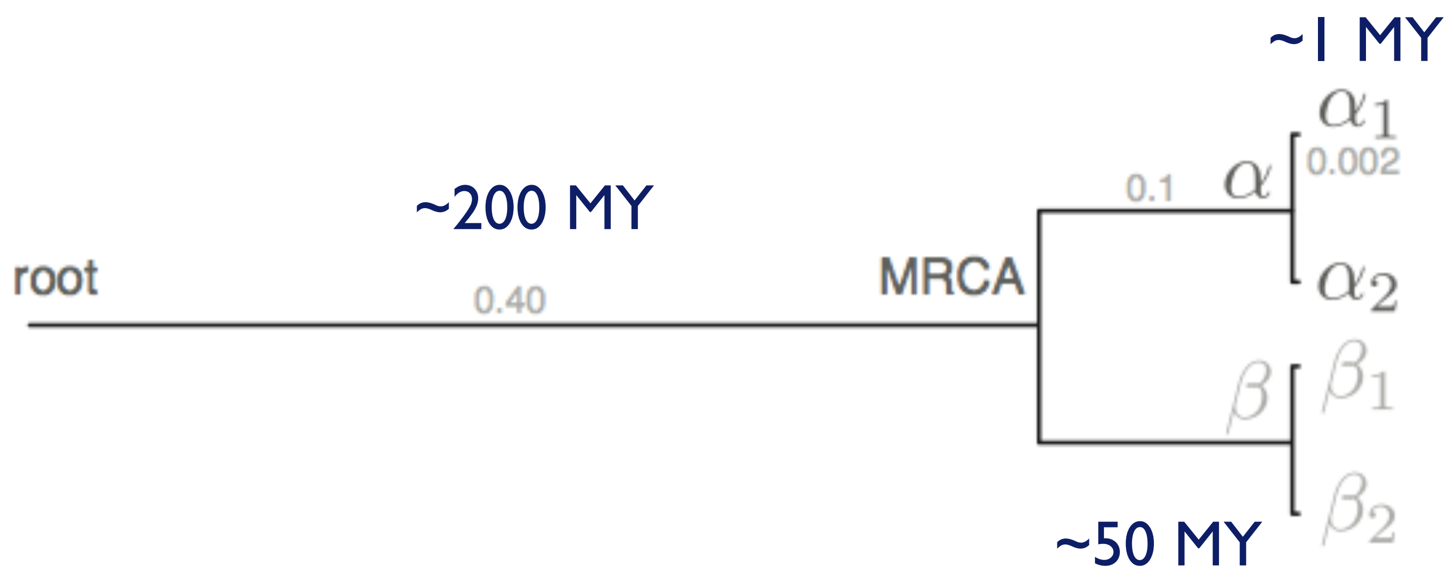
All assemblers have something to learn from this

We hope that the participants found it useful to see where their assemblers did well, and maybe to see where there are areas for improvement. The fact that no single genome assembler dominated the majority of different metrics, suggests that there is room for improvement for all participants.

assemblathon.org

Download data, results, presentations, and scripts

DVD bonus materials

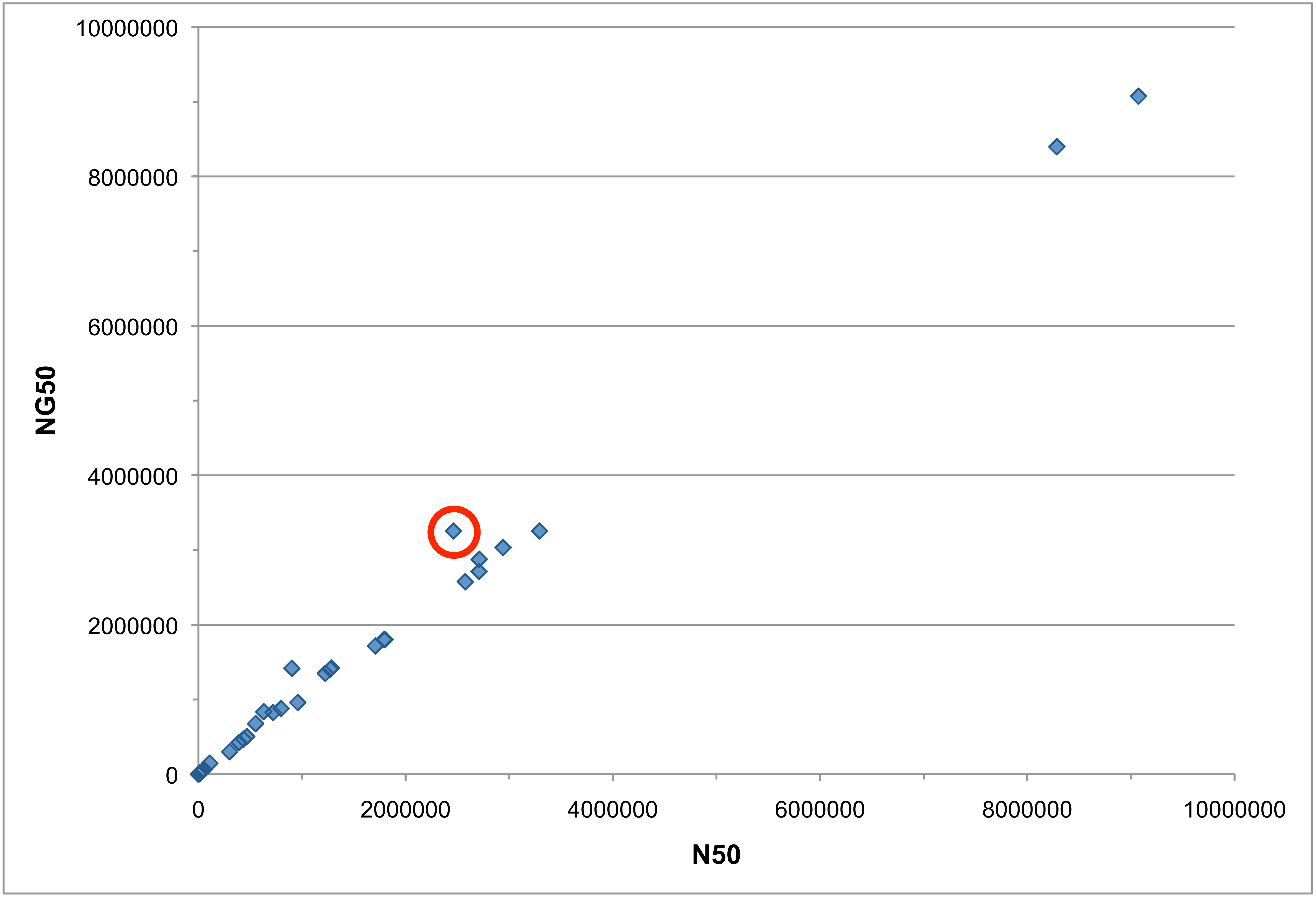


EVOLVER had a long burn in period of ~200 million years on the root sequence (human chromosome 13) to produce a most recent common ancestor (MRCA). The A and B lineages further evolved for ~50 million years before being diverged into two sub-lineages which represented each haplotype. The haplotype lineages evolved for ~ 1 million years

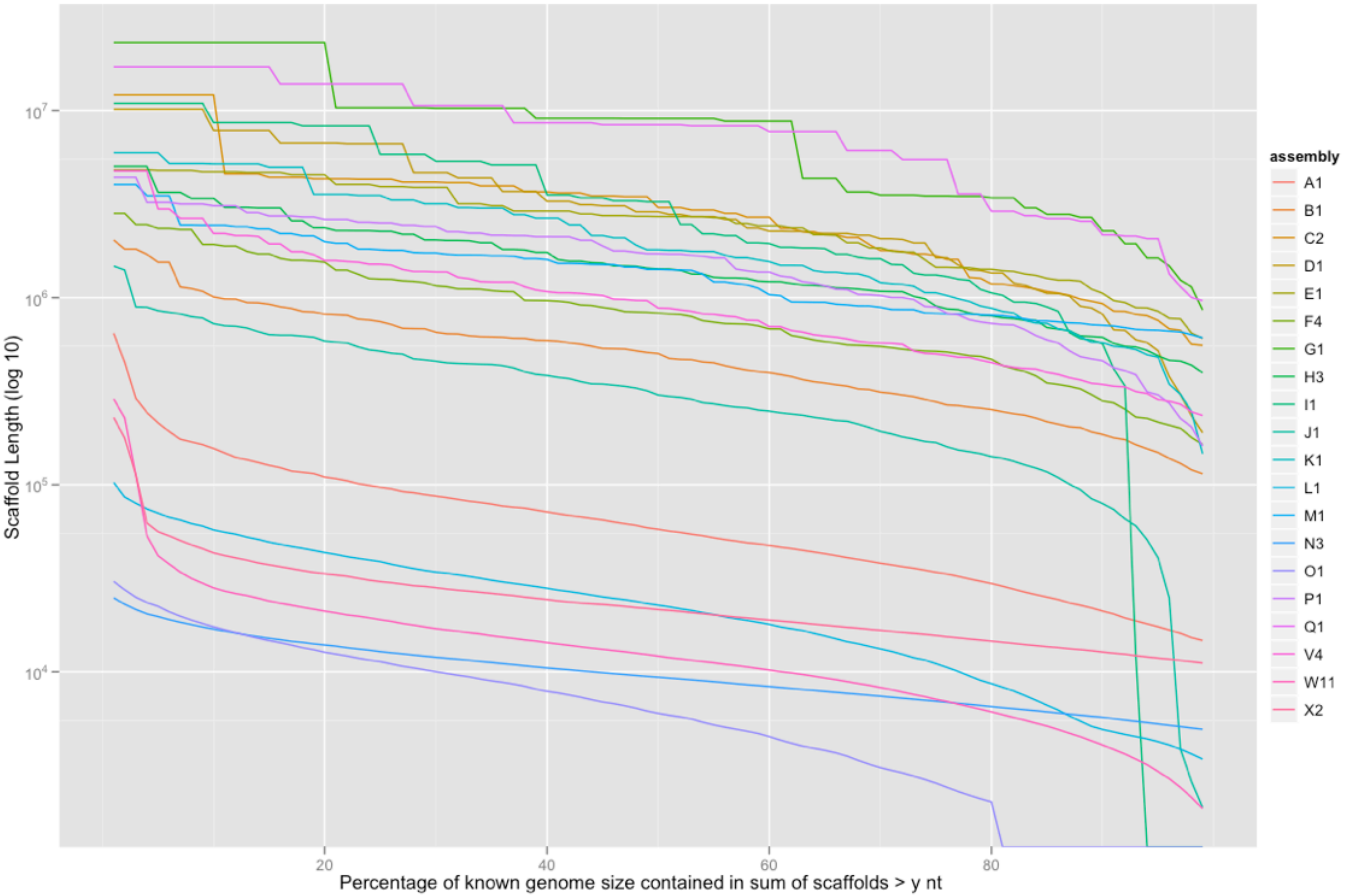
NG50

Scaffold/contig length at which you have covered 50% of total *genome* length

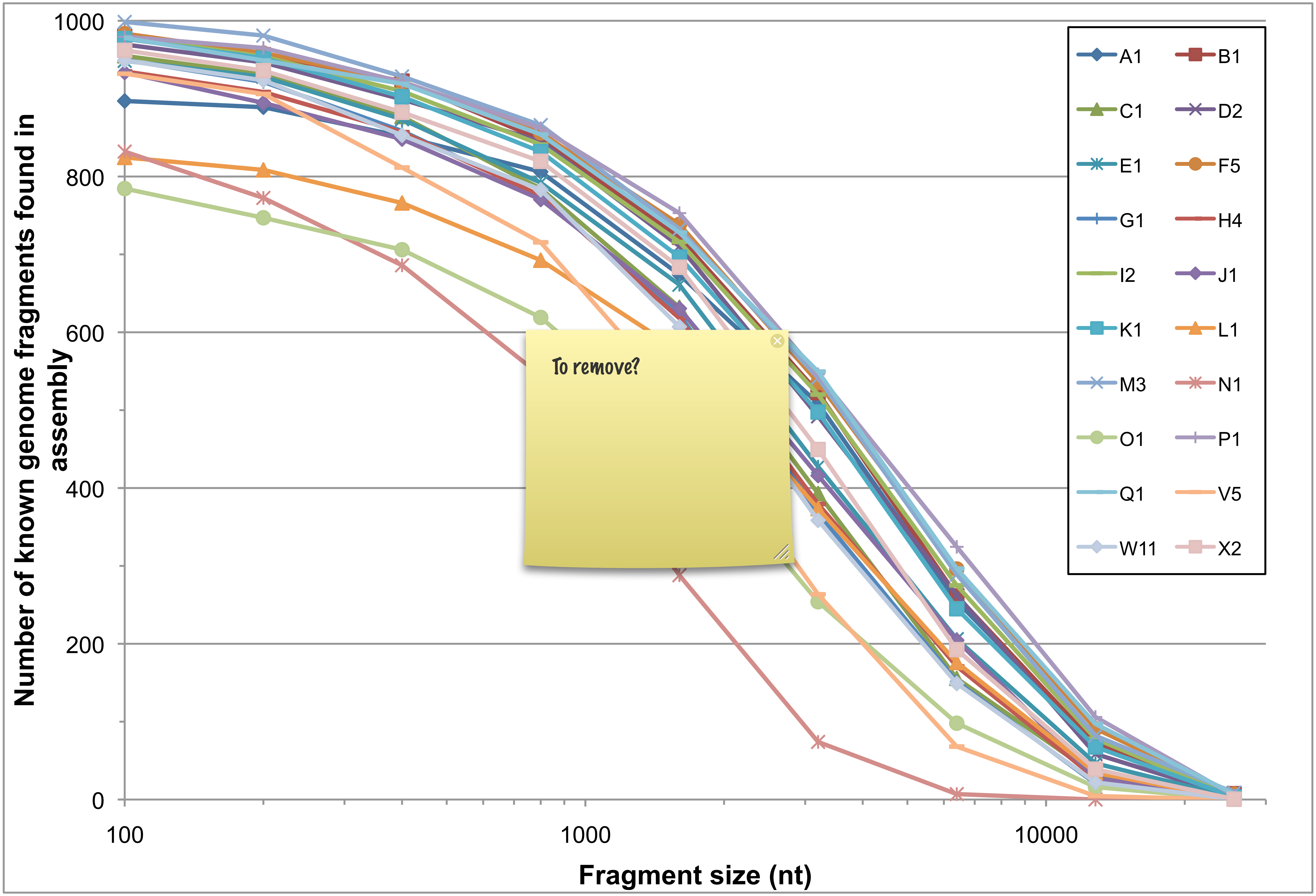
We prefer a measure that we are calling ‘NG50’. All calculations use the length of the known genome as the denominator. Can now compare assemblies to each other.



The use of NG50 rather than N50 usually makes only a small difference to the results. But some assemblies showed a bigger difference depending on which measure you used. The marked data point represents assembly I2 which had an N50 scaffold length of 2.46 Mbp, which increased to 3.25 Mbp when you instead used NG50 as the measure.



This NG(X) graph shows the values not just of NG50, but all values from NG1 through to NG99. Y-axis is on a log scale. This graph allows you compare all assemblies in a visual manner. Total area under the curve could be used as another assembly metric.



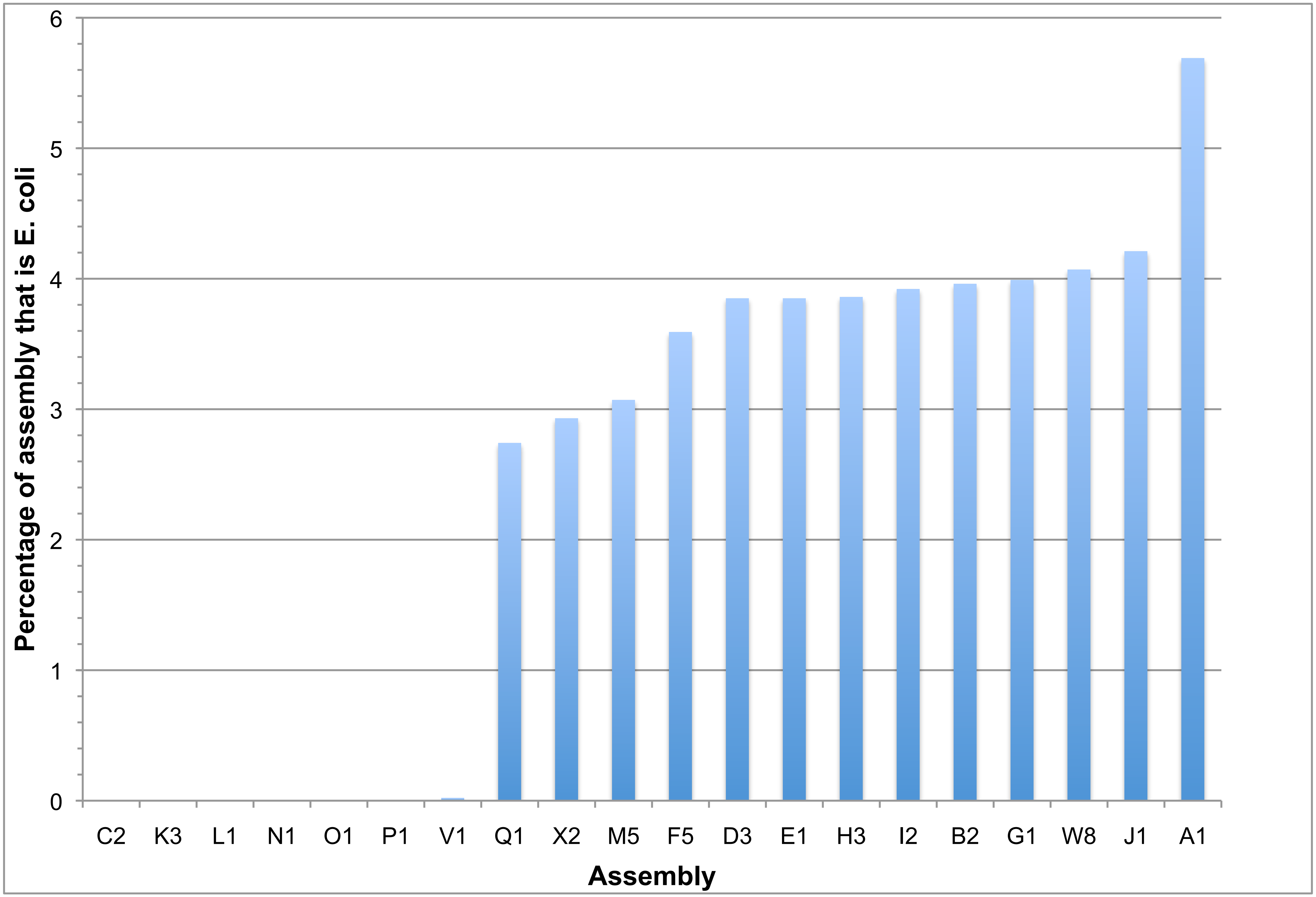
x-axis is log scale. Assemblies do relatively well at containing near-perfect regions of the known genome as long as fragments are short. Longer fragments can not be found because of errors and the problem of dealing with haplotype differences. The total area under the curve serves as a useful metric.

Contamination

“all libraries will contain some bacterial contamination”

This is a line from the README file that accompanied the set of reads that participants could download. About 5% of the reads were taken from the E. coli genome sequence, representing contamination of libraries.

E. coli contamination

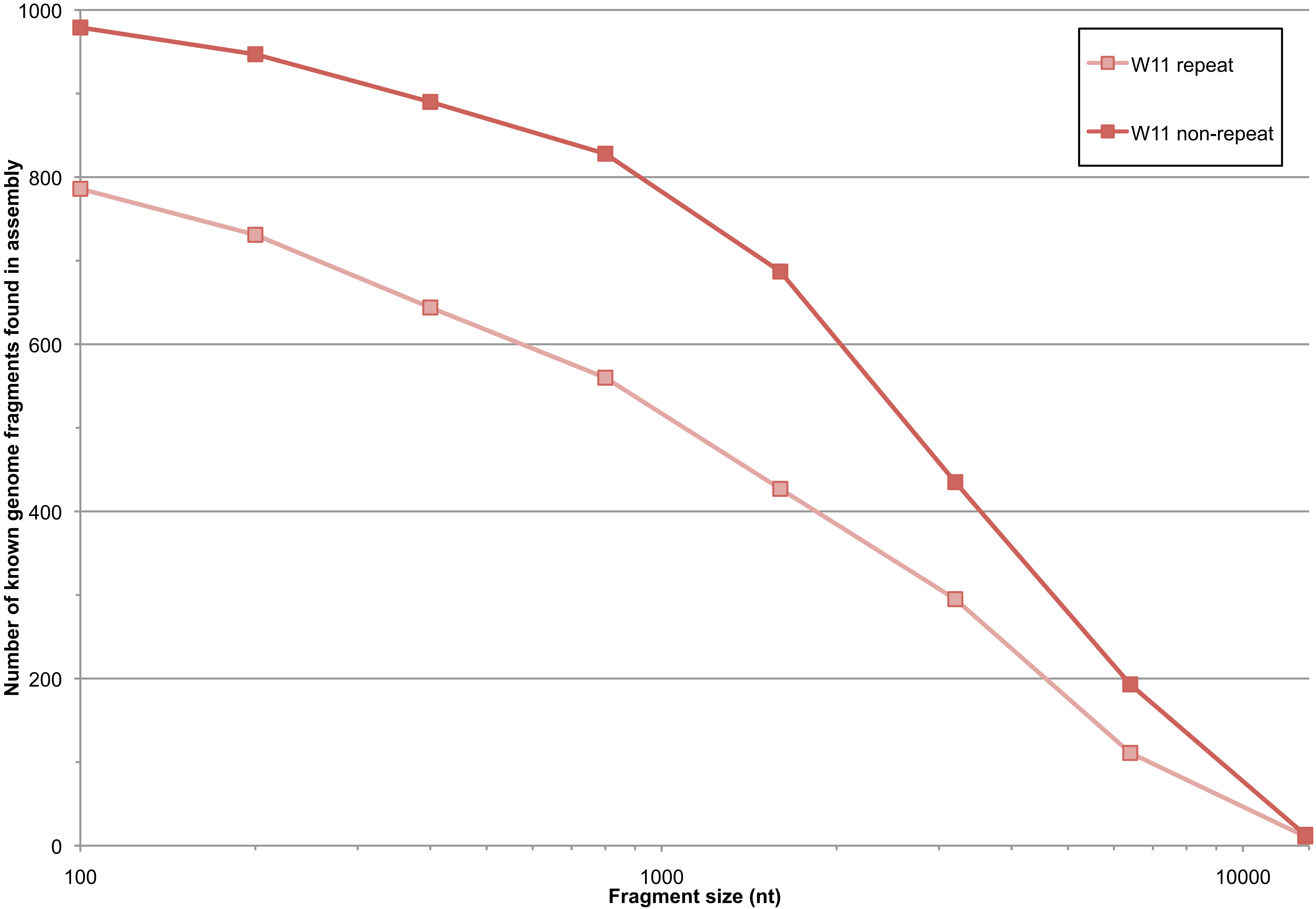


Assemblies either filtered out all contamination or they ended up with the E. coli genome sequence. All of the assemblies with contamination (apart from Q1) contained 100% of the E. coli genome sequence in the assembly.

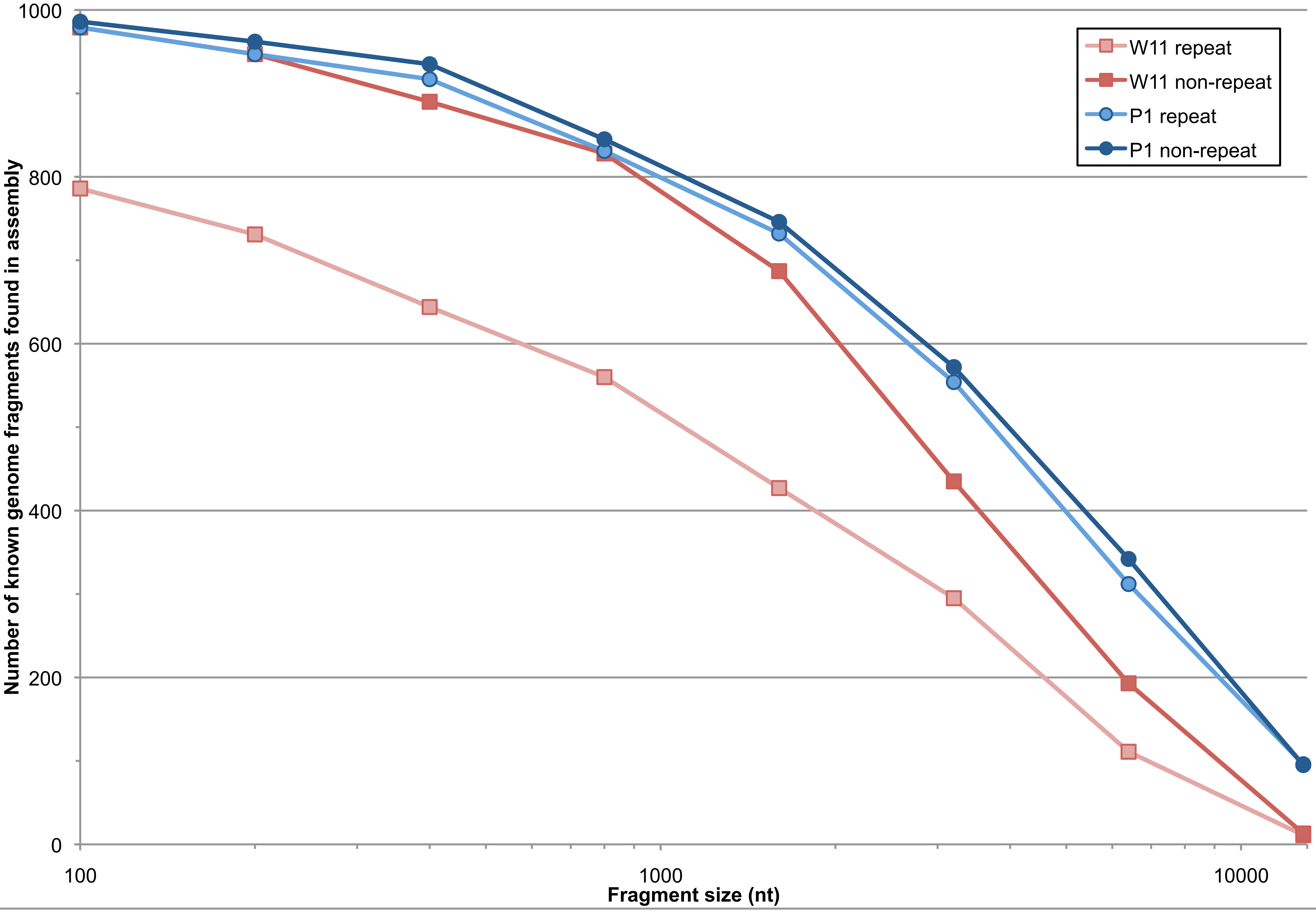
Repeat analysis

Choose fragments that either overlap or don't overlap a known repeat

Can separate out fragments on the basis of whether they overlap any sort of repeat in the known genome. This allows us to see how well different assemblers deal with assembling repeats. About 6% of the species A genome is a repeat of one kind or the other (from homopolymer runs up to transposons).



The (intentionally bad) W11 assembly does a poor job at coping with repeats, and the two lines on the above graph are far apart. The assembly contains fewer fragments of the species A genome that overlap known repeats. The sum difference between the repeat and non-repeat lines also makes for a useful metric.



In contrast, the P1 assembly does a very good job at coping with repeats, and the two lines on the above graph are very close together.

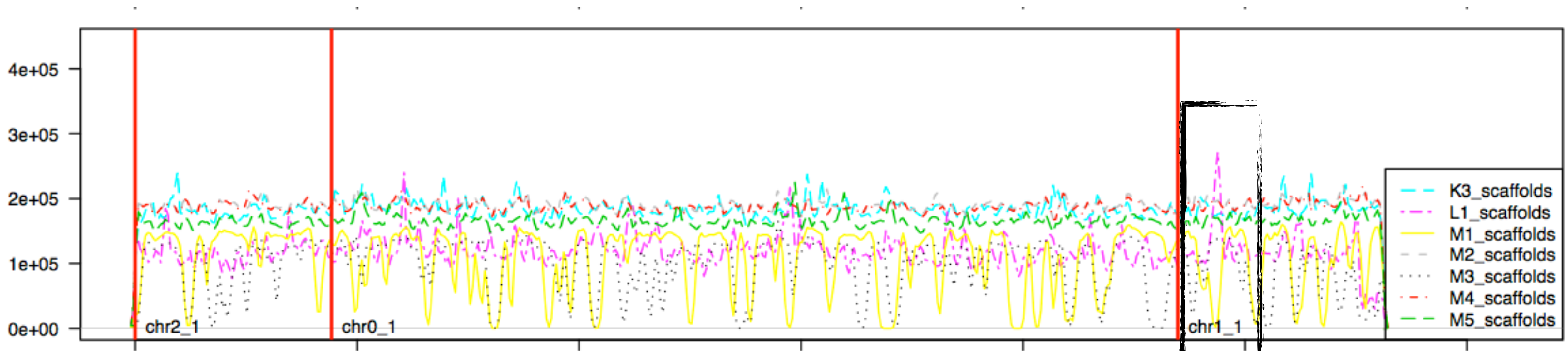
Mauve analysis

Uses whole genome alignment to reveal:

Miscalled bases
Uncalled bases
Missing bases
Extra bases
Misassemblies
Double cut & join distance

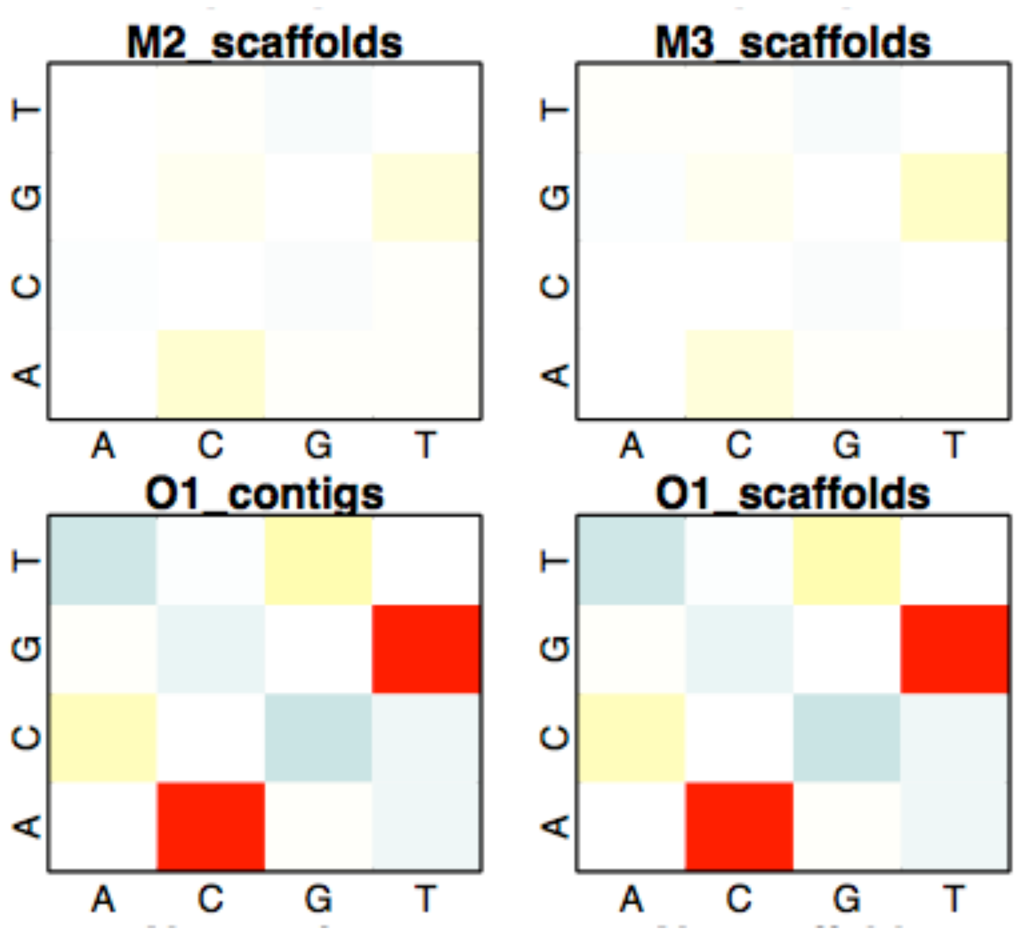
Mauve analysis by Aaron Darling.

Density of miscalled bases



Each panel shows analysis of 7 assemblies. Red lines denote boundaries between the 3 chromosomes of species A. Black box shows an area where many assemblies had a peak of miscalled bases.

Direction of miscalled bases



Most miscalled bases had no bias, but some assemblies had miscalled bases that were more likely to be a specific other base as this heat map shows.