Metagenomic Assembly with Ray and Velvet

Content

- Assembly Intro
  - Velvet
- Ray on single genomes (2010)
- Ray on metagenomes (2012)
- Benchmark Method
- Benchmark Conclusions

Assembly intro

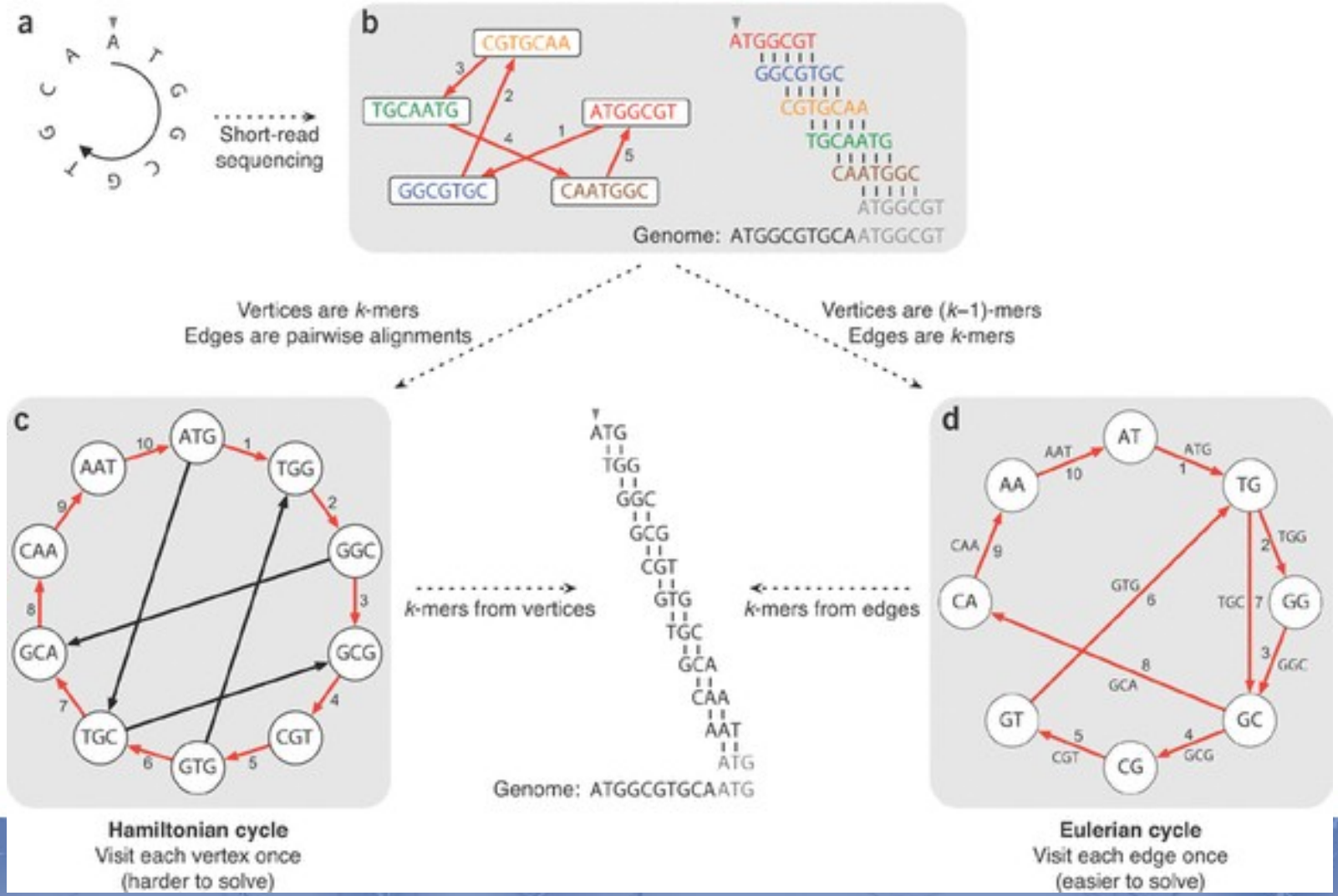
- Different technologies to sequence your sample
  - Sanger (~400-600nt, error rate ~0.001-1.0%)
  - 454 (~400nt, error rate ~1-4%)
  - Illumina (~100nt, error rate ~0.1-1%)
  - PacBio (~3k-5knt, error rate ~13-20%)
  - Ion Torrent (~200-400nt, error rate ~0.5-2.5%)
- Turn reads in to longer contigs or scaffolds
- Assembly Goal
  1) the breadth of coverage (how much of the genome is represented) is maximal
  2) the number of assembly errors (chimeric contigs, mismatches, insertions, and deletions) is minimal
  3) the number of contigs is minimal

Assembly intro

- Algorithms fit sequencing technologies
- Three general approaches
  - Overlap-Layout-Consensus (Long reads Celera 2000, Newbler 2005)
  - Greedy (Short reads, SSAKE 2007)
  - de Bruijn Graph (Short reads, Velvet 2008

Assembly intro

- De Bruijn Graph



Assembly intro

- Choosing the right K...
  - larger K more specific less coverage (span repeats, regions occurring twice, less connections in the graph)
  - Smaller K more sensitive more coverage (more connections in the graph)
  - Ideally combine both
- De Bruijn Graph potential information available
  - Overlap between kmers
  - Kmer coverage (how often does a kmer occur)
  - Read that created the kmer (choose between paths)
  - Insert size distribution between pairs (if paired reads were used)
- Programs differ in
  1) how the graph is stored
  2) how the graph is traversed

Assembly intro

- Metagenomic assembly more difficult than single genome assembly
  - Number of genomes unknown (maybe a rough idea)
  - Coverage of genomes differs (different abundances of genomes)
  - Closely related strains complicate the graph (in de Bruijn: anything that shares stretches of DNA shorter than K)

Assembly intro

- Velvet's most important parameters
  - K
    - larger K more specific less coverage (span repeats, regions occurring twice, less connections in the graph)
    - Smaller K more sensitive more coverage (more connections in the graph)
    - Ideally combine both
  - Expected coverage
    - Low coverage kmers are errors, high coverage repeats
  - Coverage cutoff
    - Low coverage kmers are errors
- Paired read data can be used to determine path in the graph

Ray Single Genomes (2010)

- Ray is a de Bruijn Graph assembler
- Three reasons for Ray

1) Parallel, can be run over multiple nodes
   - faster real time computation + distributed so memory requirements go down (Message Passing Interface)
2) Mixed sequencing data (454 and Illumina)
3) No coverage cut-off like Velvet but an approach that focuses on seed selection with global coverage and seed-extension based on local coverage, reads and paired reads

Ray Single Genomes (2010)

- Select seeds
  - Seeds are walks in the graph with a *coverage* > *(c_min* + *c_peak) / 2* and vertices af indegree and outdegree at most one (basically highly covered unitigs)
- Increase seeds based on paired information and/or read information

Ray Meta (2012)

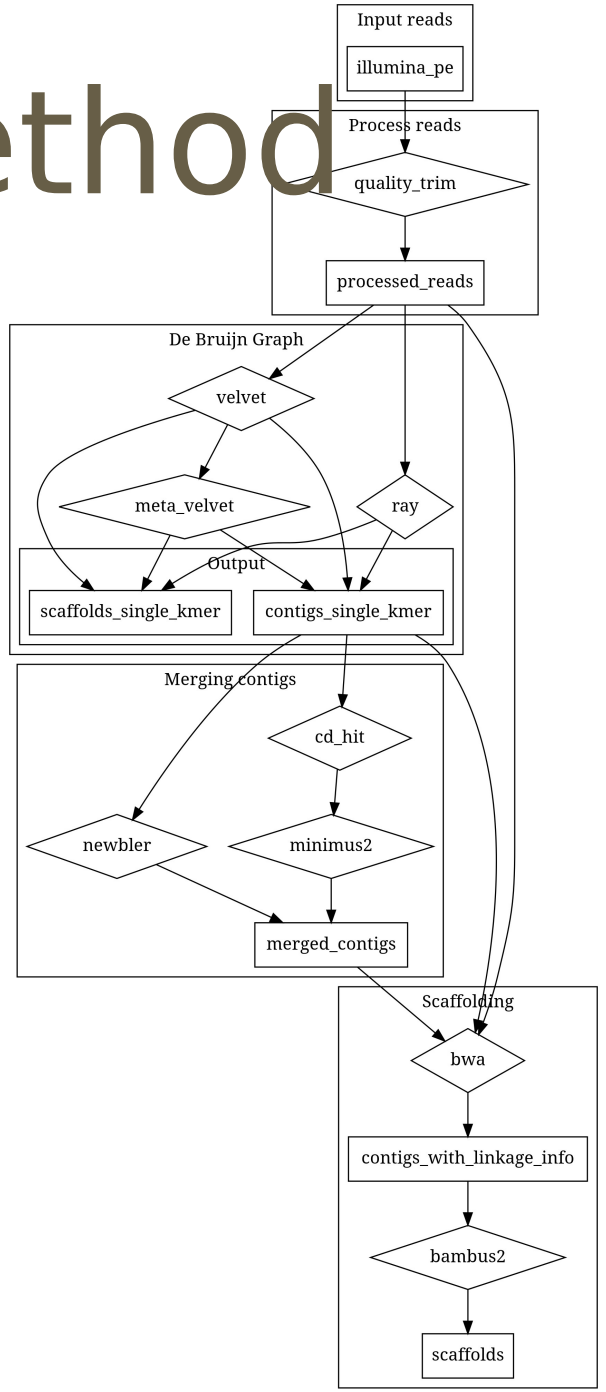
- Metagenomic sample has different abundances
- From Ray to Ray Meta
  - Seed selection changed, cov_peak and cov_min now local to unitig (walks with indegree and outdegree at most one)
- No graph simplification
  - like Meta-Velvet and Meta-IDBA based on coverage and connected components, because the graph is not easily mutable in Ray with its distributed nature

Benchmark Method

- Use an *in vitro* simulated metagenome with known species
  - 59 species, total size 195Mb
  - Two abundance distributions of genomes
    - Even, approximately the same genome copy numbers
    - Uneven, log-normal distribution of phyla similar to soil
    - Assemblies can be validated by aligning them against the reference genomes (we used nucmer for this)
- Use Illumina paired end reads since that is currently one of the most popular sequencing techniques for metagenomic samples

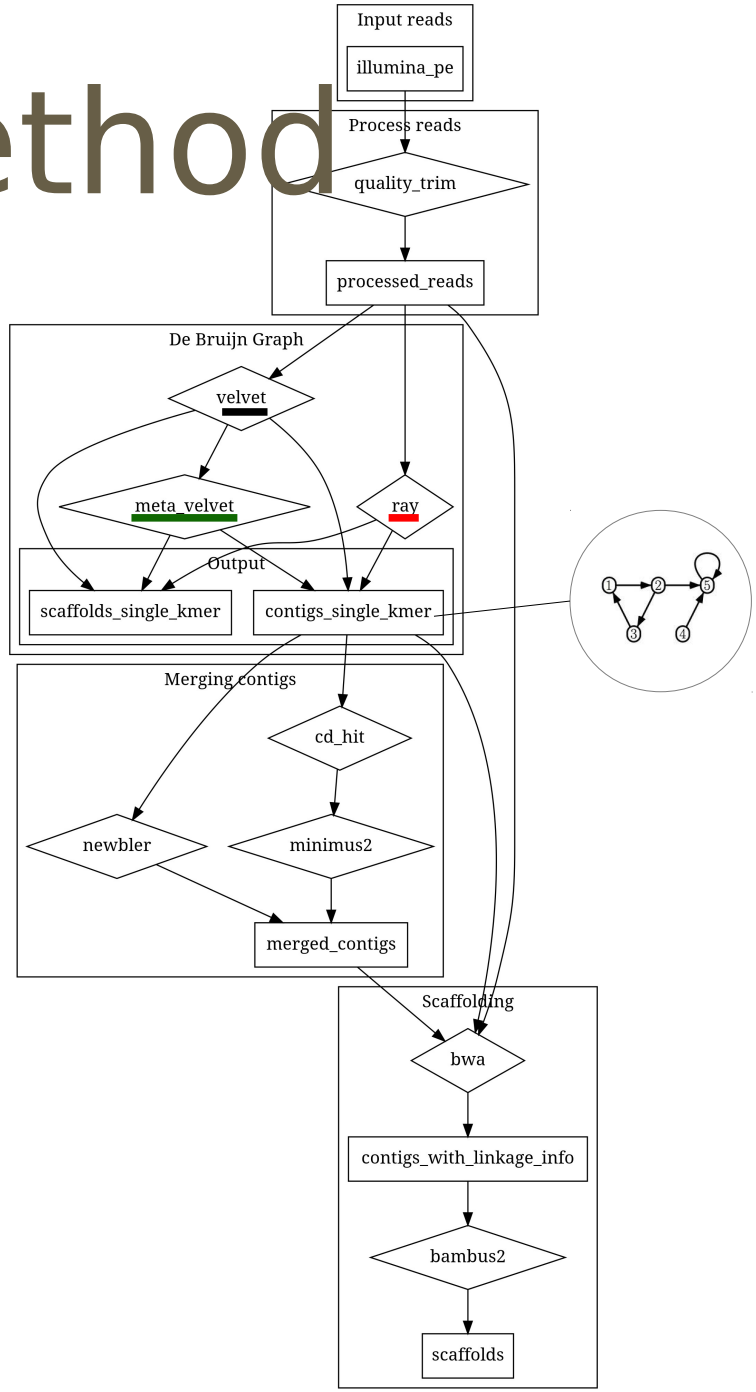
Benchmark Method

- Enormous amount of assembly strategies possible
  - Select a number of assembly strategies to test



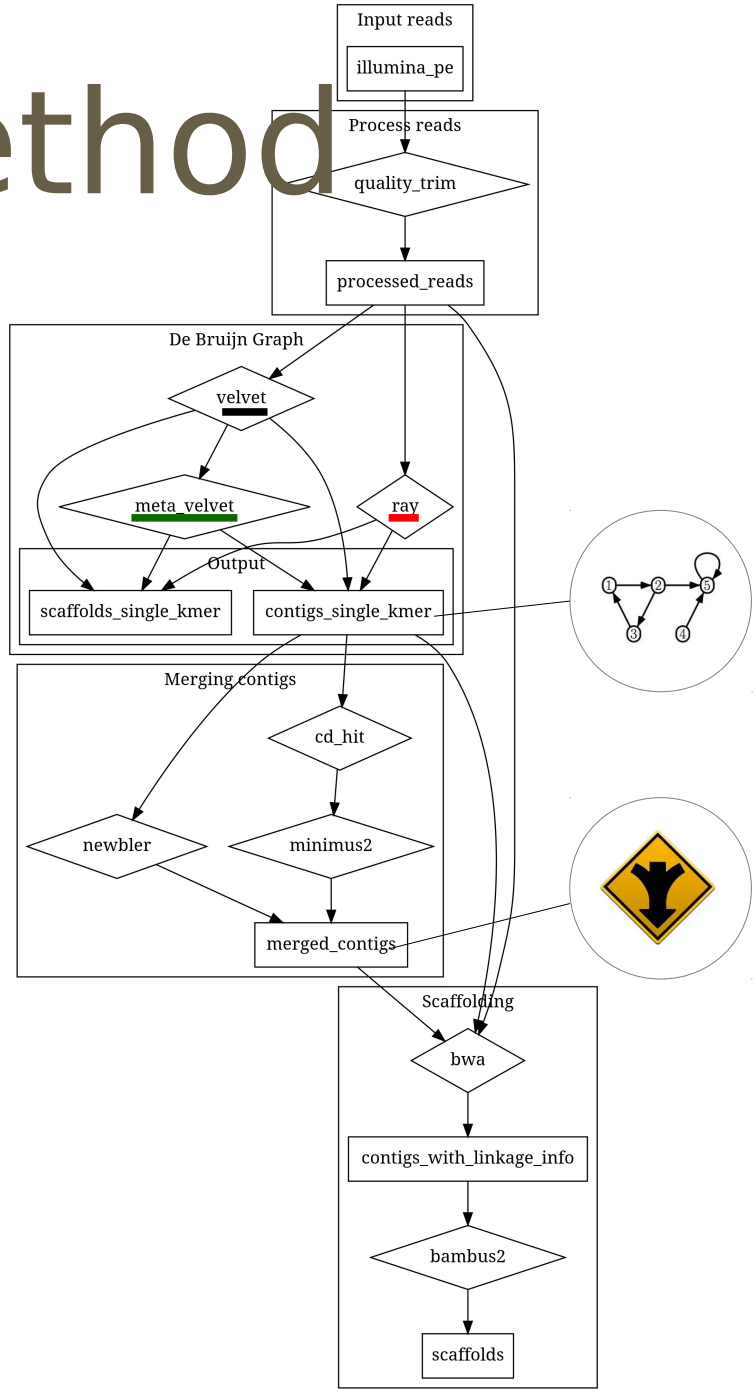
Benchmark Method

- Enormous amount of assembly strategies possible
  - Select a number of assembly strategies to test
- Contiging with
  - Velvet
  - Meta-Velvet
  - Ray



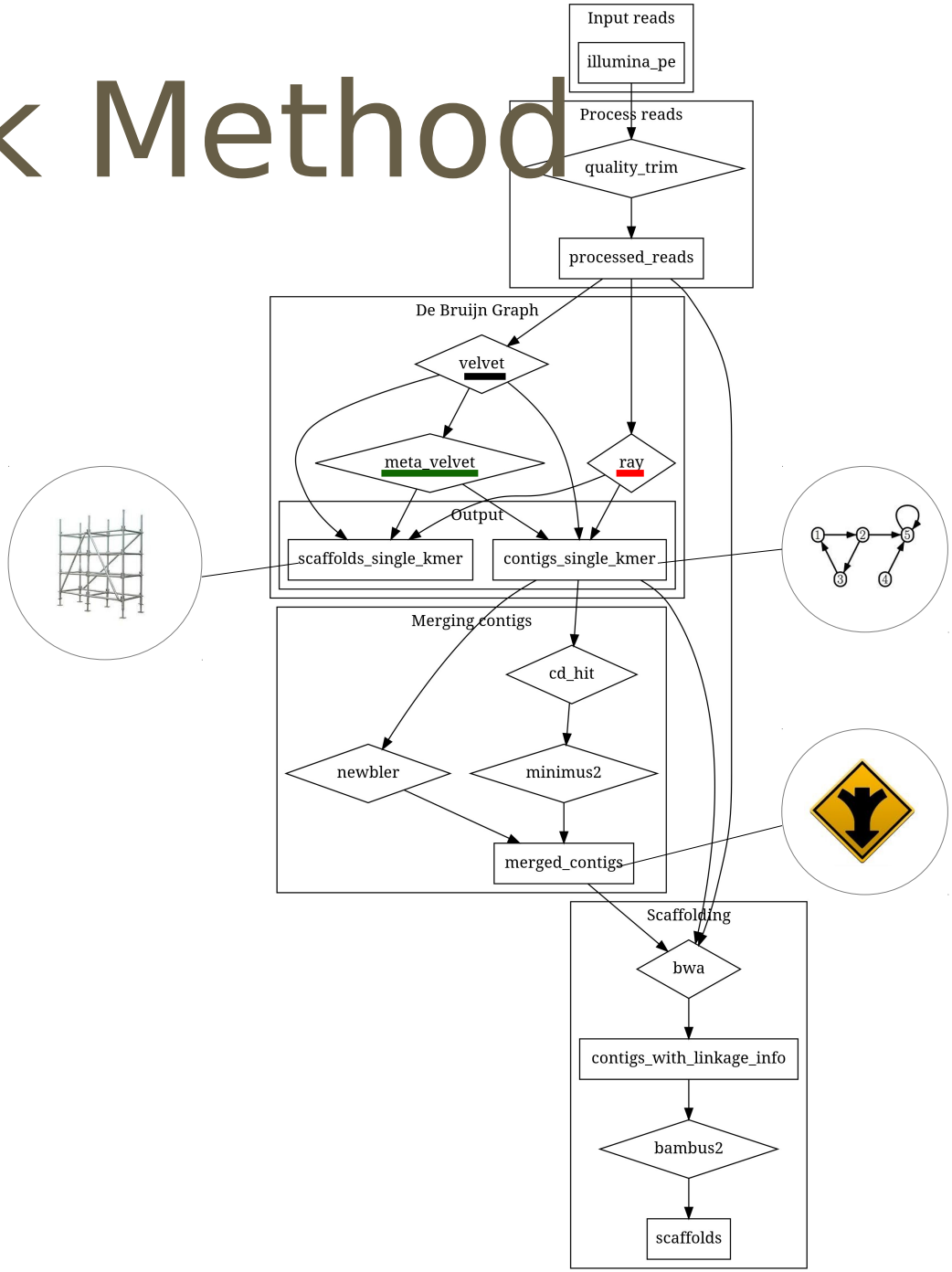
Benchmark Method

- Enormous amount of assembly strategies possible
  - Select a number of assembly strategies to test
- Contiging with
  - Velvet
  - Meta-Velvet
  - Ray
- Merging with
  - Newbler
  - Minimus2



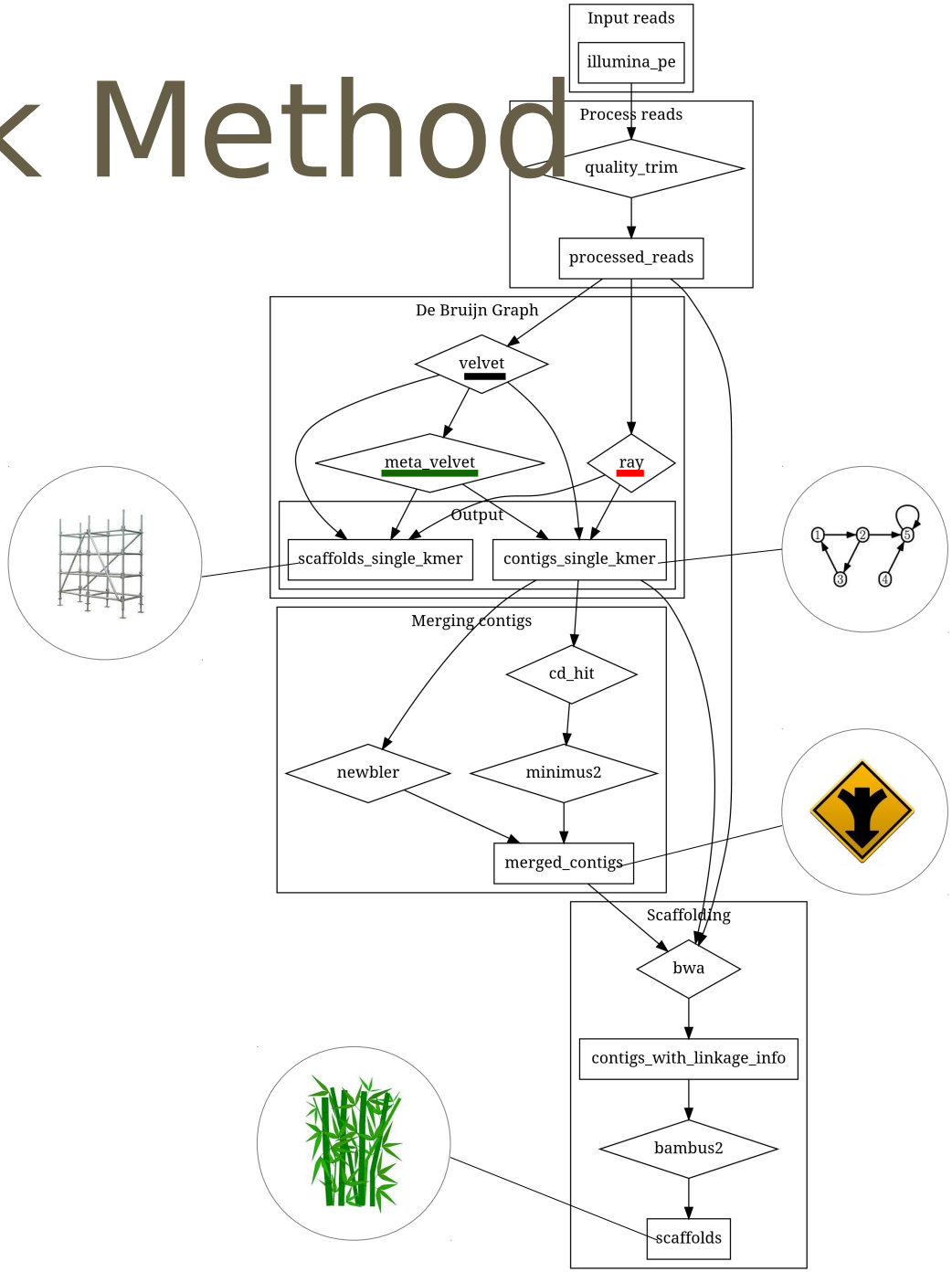
Benchmark Method

- Enormous amount of assembly strategies possible
  - Select a number of assembly strategies to test
- Contiging with
  - Velvet
  - Meta-Velvet
  - Ray
- Merging with
  - Newbler
  - Minimus2
- Scaffolding with
  - Velvet
  - Meta-Velvet
  - Ray



Benchmark Method

- Enormous amount of assembly strategies possible
  - Select a number of assembly strategies to test
- Contiging with
  - Velvet
  - Meta-Velvet
  - Ray
- Merging with
  - Newbler
  - Minimus2
- Scaffolding with
  - Velvet
  - Meta-Velvet
  - Ray
  - Bambus2



Benchmark Method

- Enormous amount of assembly strategies possible
  - Select a number of assembly strategies to test
- Contiging with
  - Velvet
  - Meta-Velvet
  - Ray

  = 3 strategies
- Merging with
  - Newbler
  - Minimus2

  = 2 x 3 stragies = 6 strategies
- Scaffolding with
  - Velvet
  - Meta-Velvet
  - Ray
  - Bambus2

  = 3 strategies

  = (6 merge + 3 contig) strategies = 9 strategies

Total of 21 strategies

Benchmark Method

- How to compare all these different strategies?
- Validation of metagenomic assembly often focuses on one or more of the following points:
  1) contig/scaffold length distribution
  2) contig/scaffold coverage of the reference metagenome
  3) chimericity and erroneousness of the contigs/scaffolds
  4) functional annotation
  5) phylogenetic classification
- We focus on the first three since those also tend to improve 4 and 5 as shown by Mende et al (2012)
- Select winner in three categories

Conclusions

- Difficult to name one true winner. Depends on what the post-processing will be like
  -  If short contigs useful: Ray or Velvet. Velvet preferred if really short contigs 100-1000 (more coverage of metagenome).
  -  If only interested in long contigs and less so in their purity, perhaps choose Meta-Velvet.
  -  Merging seems to work quite well although you lose some metagenome coverage
    - Might want to avoid scaffolding since it doesn't work that well
      -  bambus2 in particular loses a lot of metagenome coverage
      -  Internal scaffolding works better than bambus2, but merging alone does a better job

Future work

- See how results hold with functional annotation
  - How many false positives and true positives versus reference metagenome
  - What is an adequate cut-off length to use for the assemblies? (700 with Prodigal)
- Compare reference-less validation against this validation
  - Compare for instance with Feature Response Curve (Vezzi, 2012)
- Make the validation pipeline available as a web service