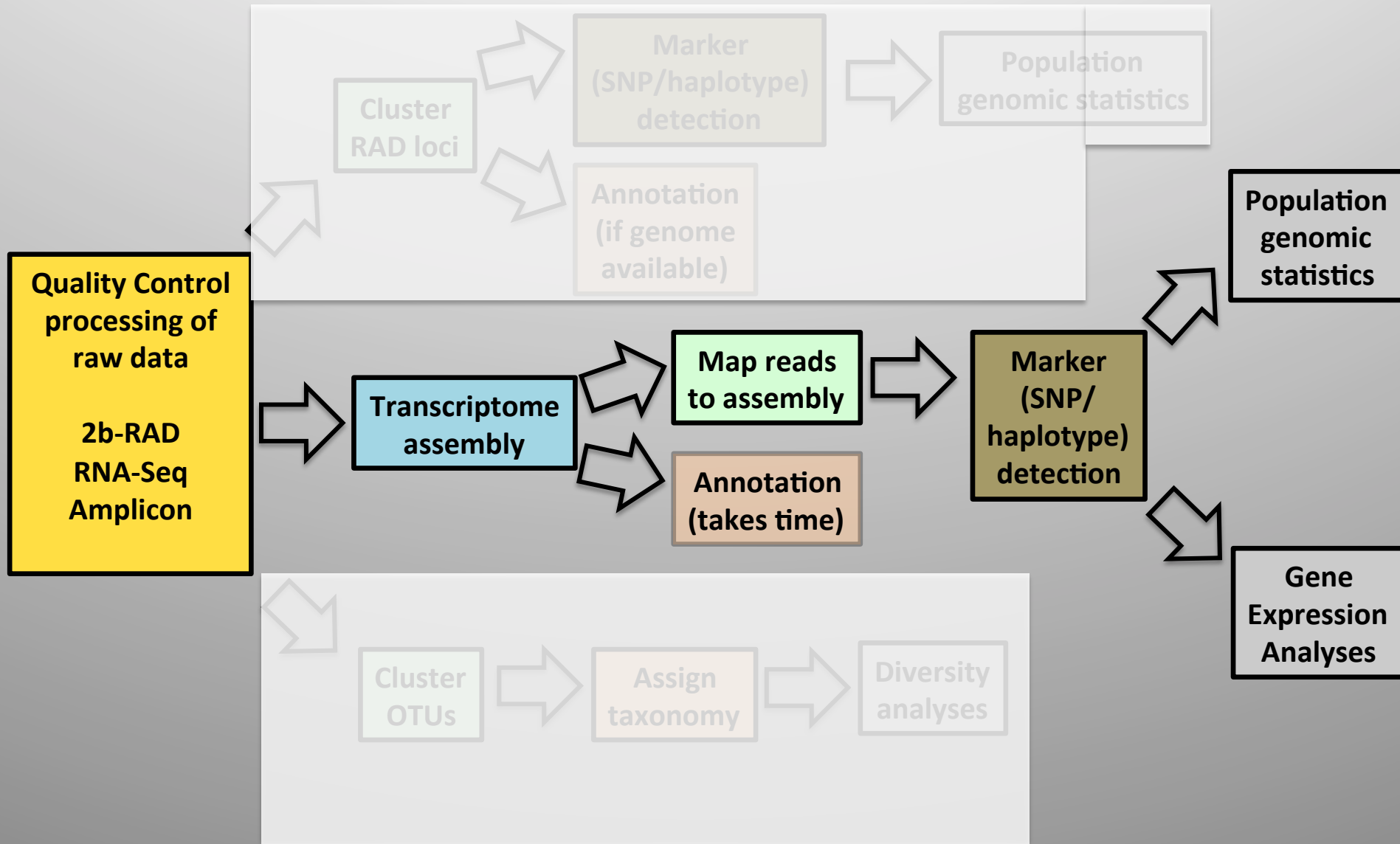




De novo assembly– software and
algorithms

Analysis pipeline for RNA-Seq data



De novo genome/transcriptome assembly

Mission: combine hundreds of millions of short reads to a genome/transcriptome (with no pre-existing genomic information available).

Method: Look for regions of overlap – create *Contigs* (=contiguous sequences)

De Bruijn graphs are excellent mathematical tools for this purpose

There are many different algorithms and programs for doing this.

3 main things to weight against each other when choosing:

- Speed
- Memory requirements
- Accuracy

See Martin & Wang 2011. *Nature Reviews Genetics*, for a great review of all caveats / issues with transcriptome assembly

De Bruijn graphs

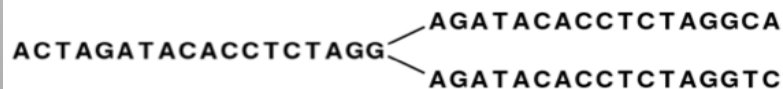
Consist of nodes (k-mers) of k length, connected by edges

$k-1$ overlap between each node.

Polymorphisms cause bifurcations in the graph



Nodes connected by one edge can then be merged to save space



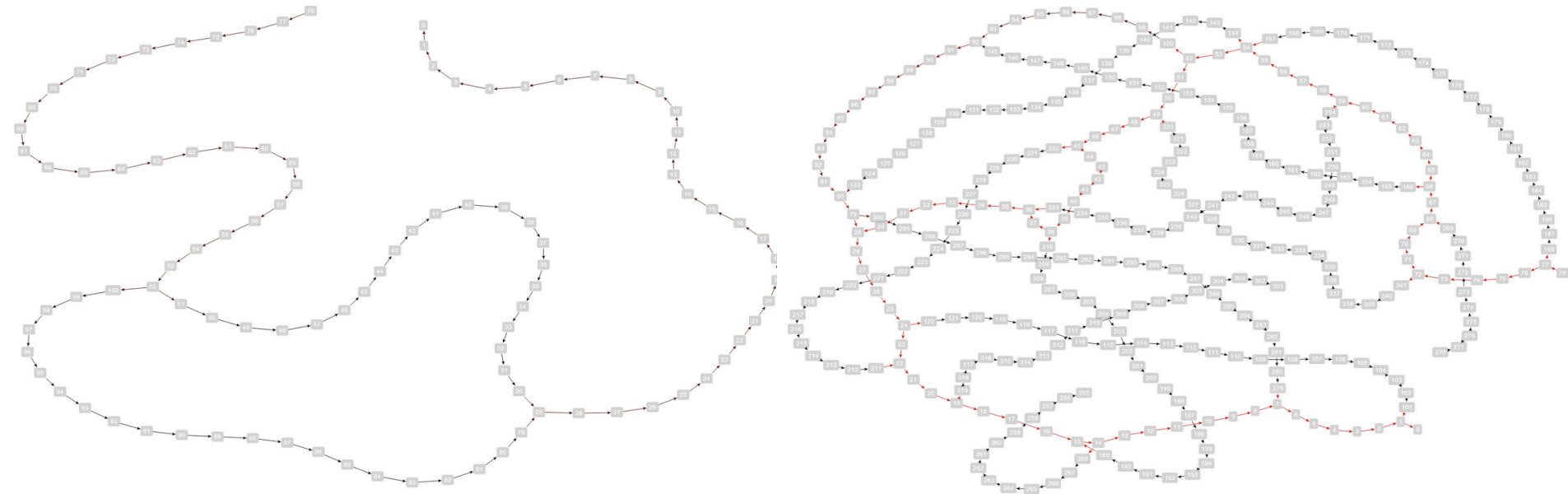
Polymorphisms result in “bubbles” with size dependent on k



De Bruijn graphs

1 SNP

5 SNPs



Polymorphisms in transcriptomes

Can be due to

- Sequencing errors
- SNPs /Indels
- Alternative splicing
- Paralogous genes

Can use base frequencies to remove sequencing errors, but there is no perfect way to distinguish true polymorphisms from false positives in some cases.

Currently, Trinity seems to be the best at finding alternative isoforms/paralogs

CLC genomics workbench is a fast low-memory (but high-cost) alternative (Bräutigam et al. 2011)

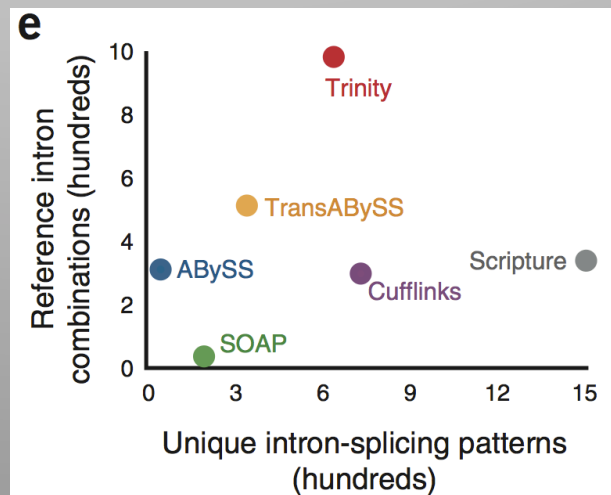
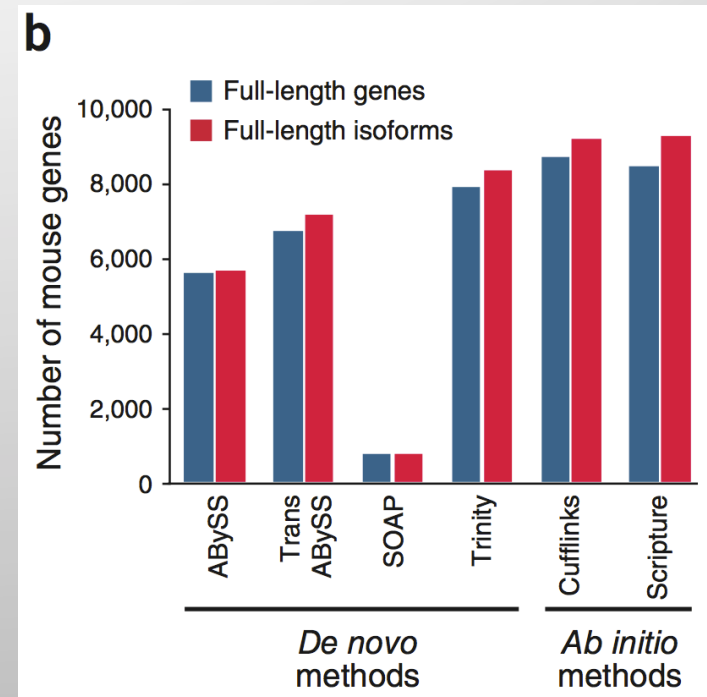


Fig 5; Grabherr et al . 2011

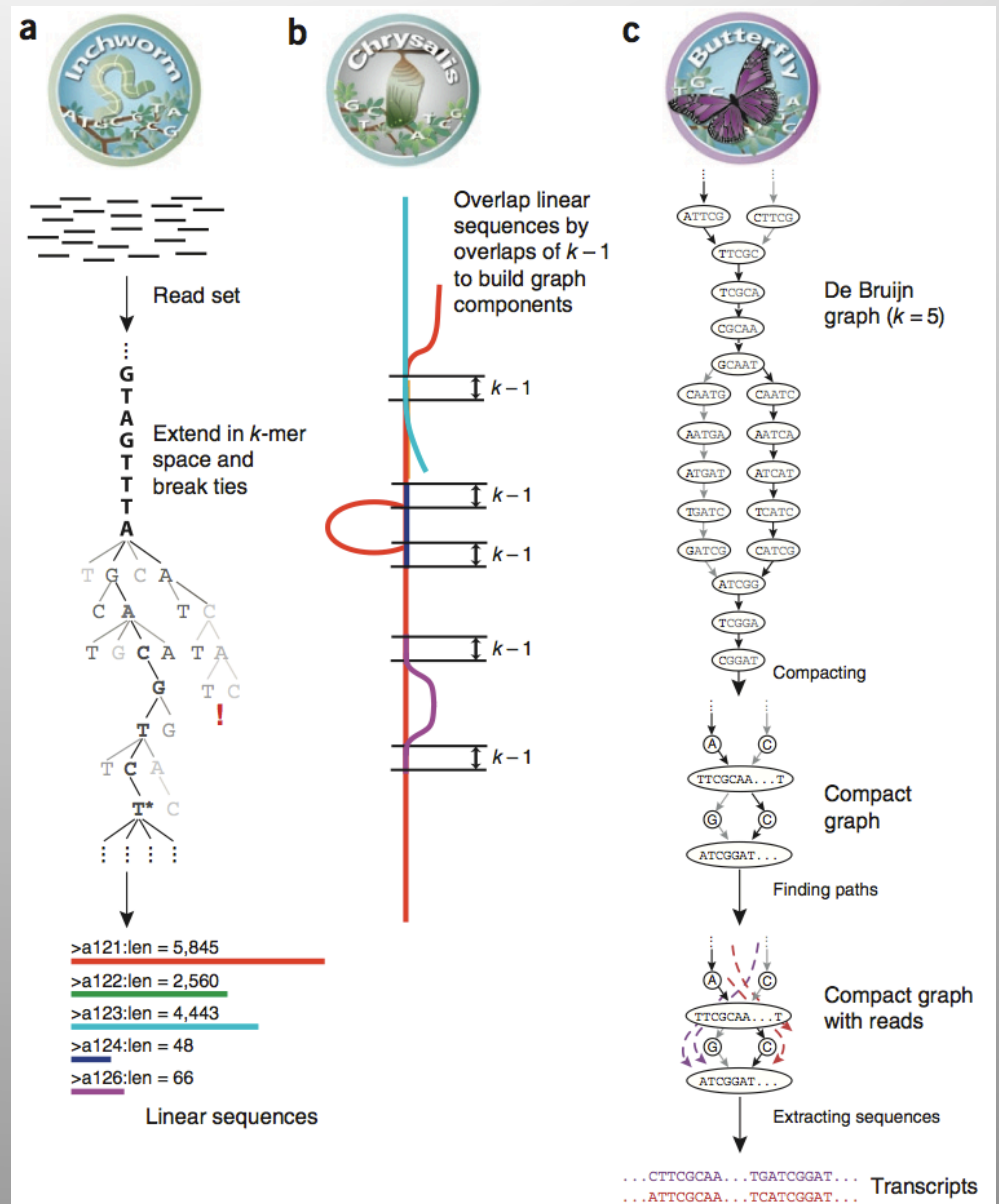
The Trinity algorithm

- Inchworm creates a table of all k-mers in the read data
- Chrysalis overlaps k-mers to create De Bruijn graphs
- Butterfly compacts graphs, saves transcripts/isoforms.

Each part can be run independently, or they can be run combined.

Currently only supports k=25 for combined runs

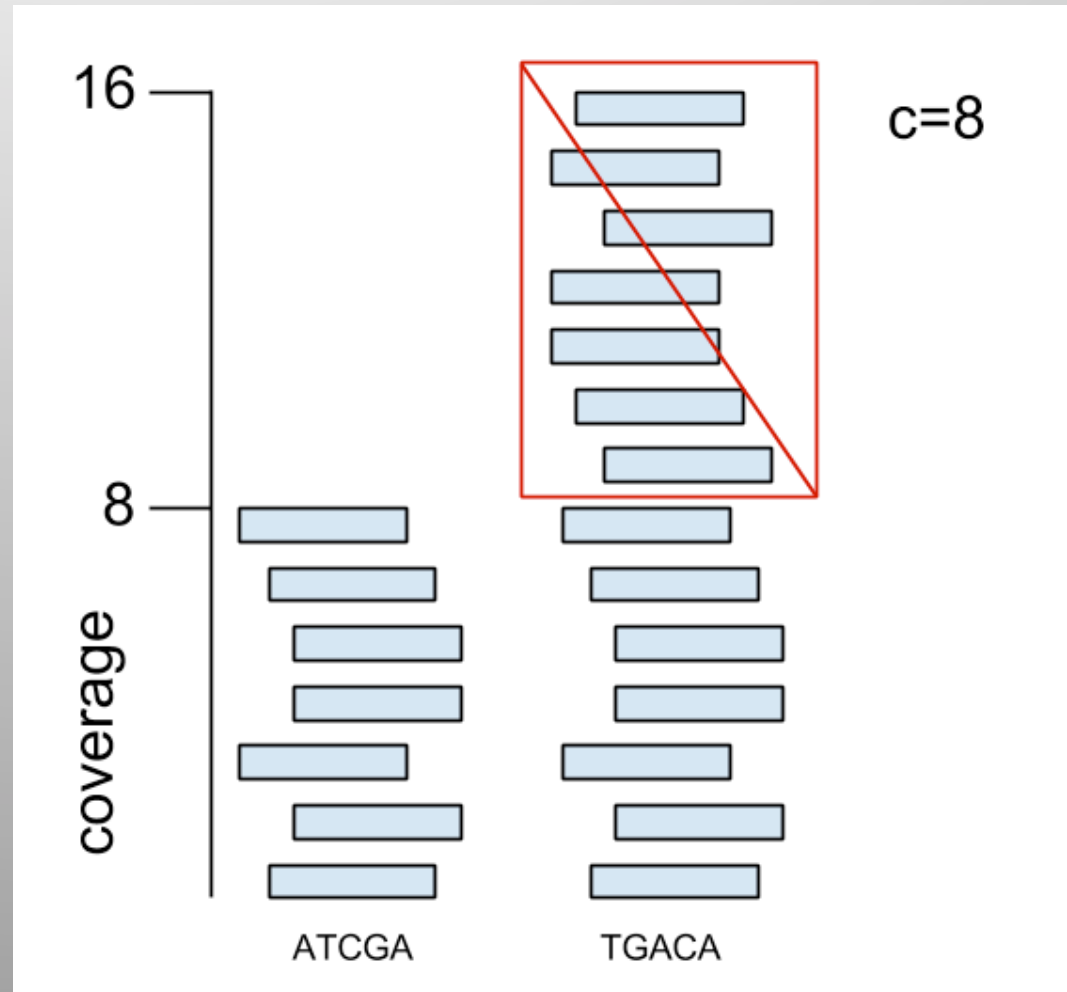
Requires ca. 1 GB RAM per million reads.



High coverage is not useful after a point

Trinity uses a digital normalization step that removes highly repeated sequences called “khmer”

Reduces computational load.



CLC genomics workbench

- Fast – low memory alternative for both genomes and transcriptomes.
- Also does many other things (QC, alignment, SNP detection, etc..)
- Uses De Bruijn graphs, but exact algorithm is proprietary
- License ca. 5000 USD /yr, but 2-week free trial versions are available for download at:

<http://www.clcbio.com/products/clc-genomics-workbench/>

Optimal workflow for transcriptome assembly (which we will NOT do today)

Sample a breadth of developmental stages and tissues.

Prepare non-normalized, strand-specific RNA-seq libraries.

Pool and sequence libraries to generate ~30-100 million paired-end, long (100bp) reads (Francis *et al.* 2013). Additional population samples can be sequenced with Illumina single-end, short reads.

Process raw sequence data for quality as well as errors (Francis *et al.* 2013).

Digitally normalize read data (Brown *et al.* 2012).

Assemble cleaned, paired-end long-read sequence data using an assembler with the ability to resolve splice isoforms and gene paralogs (e.g. Grabherr *et al.* 2011).

Prune assembled transcripts for coverage (4x), read length (600 bp), and open reading frames (ORFs) to remove or reduce DNA contamination, non-coding RNA, chimaeras, and gene fragments (Cahais *et al.* 2012).

Iterate the above steps and evaluate assemblies with both quantitative and qualitative metrics (BOX 2).

↓
Without a reference assembly: evaluate new assembly completeness by searching for conserved eukaryotic orthologous genes (COGs) (Tatusov *et al.* 2003; Parra *et al.* 2007).

↓
With a reference assembly: evaluate new assembly through BLAST comparisons to a closely related species; measure completeness, and identify and exclude or correct errors such as chimaeras, collapsed paralogs, and separated isoforms or allelic variants (Cahais *et al.* 2012).