DATA ANALYSIS

Oksana Riba-Grognuz

Getting Started

http://goo.gl/bJCJJ

```
$ mkdir /scratch/cluster/weekly/username
$ cd    /scratch/cluster/weekly/username
$ unzip
  /scratch/cluster/monthly/oribagro/summer2012_Oksana.zip
```

QC

Quality Control

illumina®

FASTQ format

http://en.wikipedia.org/wiki/FASTQ_format

illumina®

Paired and single reads

We will use FastQC

http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

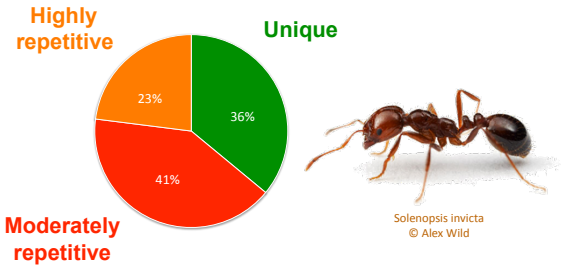
Widely used but not the only one

Our Goal

Get the best possible input for *de novo* assembler

Which data to use?
All or selected lanes?
Trimmed?
Quality filtered?

We expect



Solenopsis invicta
© Alex Wild

We expect

Genome subset:
3,252,223 bp
8 sequences

Solenopsis invicta
© Alex Wild

Input Files

Subsets of Illumina Hi-Seq lanes

Raw data

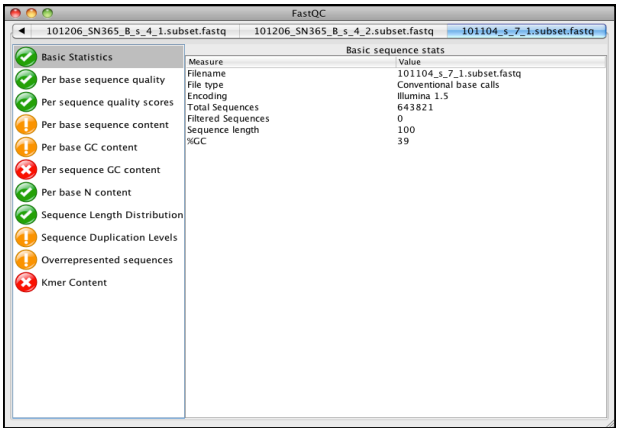
```
$ ls DNA-seq/Raw
```

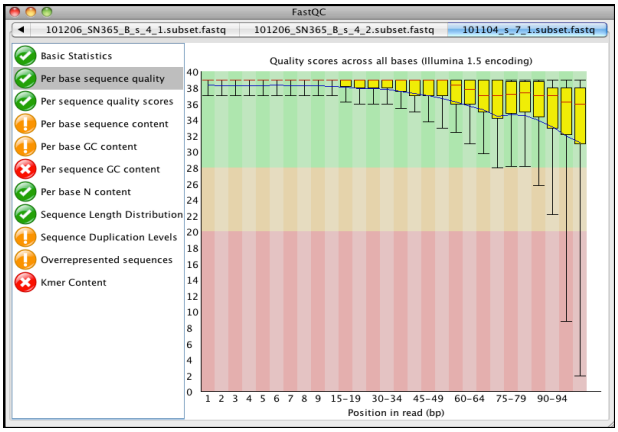
Non-redundant data

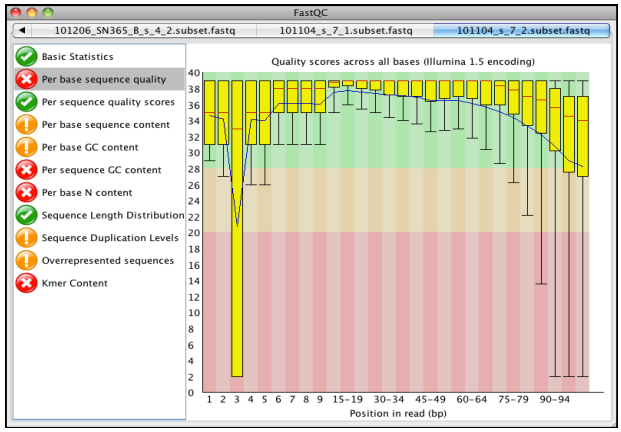
```
$ ls DNA-seq/Nr
```

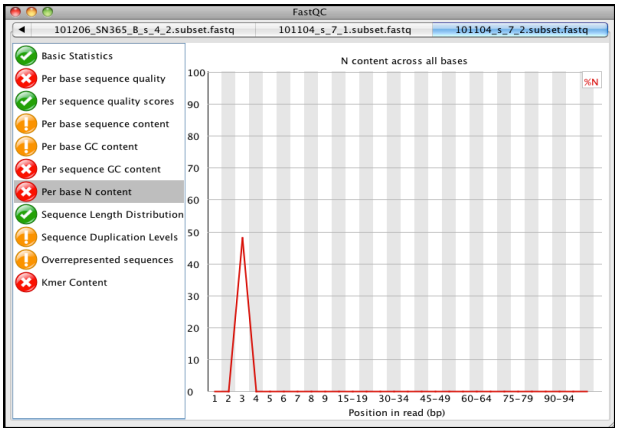
Launch FastQC

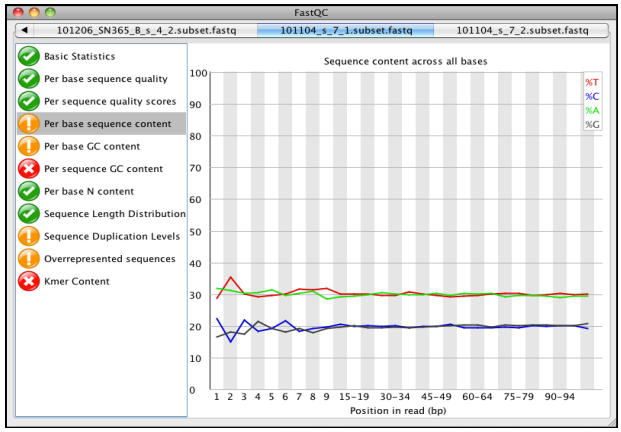
1) **Process raw data**
2) **Process non-redundant data**

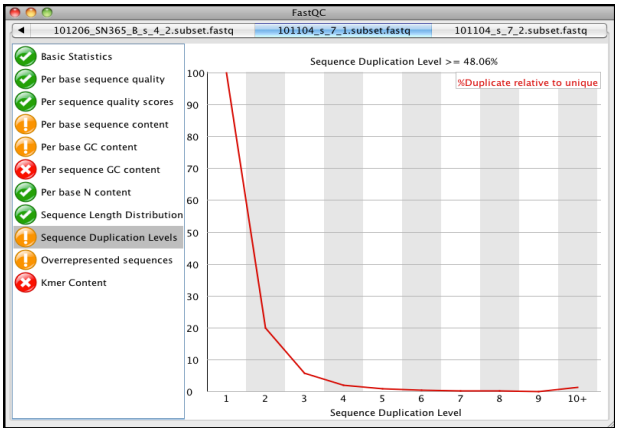












We will use FASTX-Toolkit

http://http://hannonlab.cshl.edu/fastx_toolkit/

To trim and filter Illumina reads

FASTX-Toolkit

Trim by quality

```
$ fastq_quality_trimmer –h
```

Trim by coodinates

```
$ fastx_trimmer -h
```

Input Files

Process lanes of raw OR non-redundant data

```
$ ls DNA-seq/Raw
$ ls DNA-seq/Nr
```

Output to

```
$ DNA-seq/Clean
```

Keeping Track

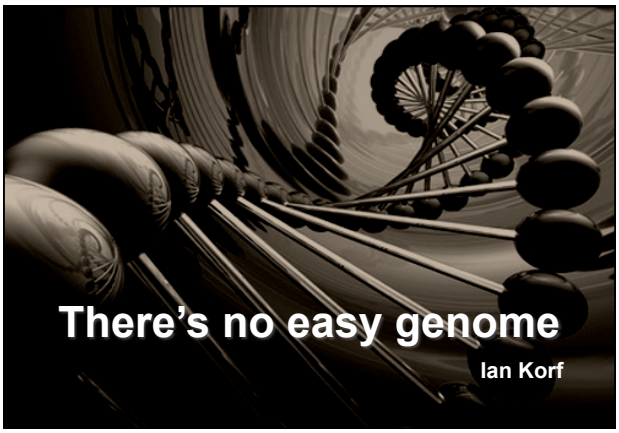
```
#!/usr/bin/env bash

currentDir=${PWD}
inDir=$currentDir/DNA-seq/Nr
outDir=$currentDir/DNA-seq/Clean

mkdir -p $outDir

myFiles=(`find $inDir -name *.fastq`)

for inFile in ${myFiles[@]}
do
   outFile=${inFile##*/}
   fastx_trimmer –f XXX -l YYY -i $inFile -o $outDir/$outFile
done

exit 0
```

Genome Assembly

De Novo

Best Practices

http://assemblathon.org/



In Resume

Every genome is a special case

So, know your genome as much as you can **BEFORE** doing sequencing

Choose appropriate strategy based on what is known

Red Fire Ant

Very repetitive genome with some repeats being very long

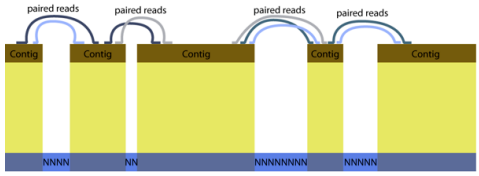
Ideally: use libraries with insert size > repeat size

The Hardest Way

We used long insert 454 libraries for the official genome release

You will try to see what happens when such libraries are not available

Defintions



Five contigs are joined into a single scaffold thanks to paired read information

Contigs and scaffolds

Assembly

We will use SOAPdenovo

http://soap.genomics.org.cn/soapdenovo.html

SOAPdenovo

The package consists of 4 programs: pregraph, contig, map and scaff

With paired end data the simplest is to run all programs in a single command

```
$ SOAPdenovo-127mer all -s config_file -o output_prefix
```

Config file

```
#maximal read length
max_rd_len=100
[LIB]
#average insert size
avg_ins=344
#if sequence needs to be reversed
reverse_seq=0
#in which part(s) the reads are used
asm_flags=3
#cut the reads from the current library to this length
rd_len_cutoff=100
#in which order the reads are used while scaffolding
rank=1
# cutoff of pair number for a reliable connection (default 3)
pair_num_cutoff=3
#minimal mapped length to contigs for a reliable read location (default 32)
map_len=60
```

Config file

```
q1=lane4_pair1.fastq
q2=lane4_pair2.fastq
q1=lane7_pair1.fastq
q2=lane7_pair2.fastq
```

All our reads come from the same sequencing library, so we define only 1 LIB

Keeping track

To find optimal parameters run multiple assemblies.

We will do at least 2 using launch scripts

```
$ ls SOAPdenovo/Assembly
conf01-lanes47_maplen60
Run_conf01-RL200D.sh
```

```
#change THIS
myconf=conf01-lanes47_maplen60

#DO NOT CHANGE
mypref=`echo $myconf | cut -f1 -d'-'`
runDir=$PWD
myconf=$runDir/$myconf

#change THIS
L=200

#change THIS
for K in 35 65
do
  # DO NOT CHANGE
  mydir="$mypref"_K"$K"_R_L"$L"_D
  mkdir -p "$mydir"
  cd $mydir
    SOAPdenovo-127mer all -p 1 -s "$myconf" -K "$K" -L "$L" -R -D -o out &> LOG
  cd ..
done
```

K-mer

The most influential parameter

Should be at least 1/3 of read length

Additional criteria may apply depending on the software used

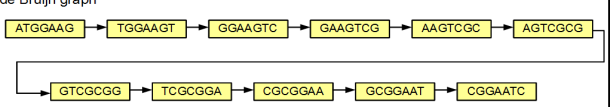
de Bruijn graph Construction

sequence ATGGAAGTCGCGGAATC

7mers

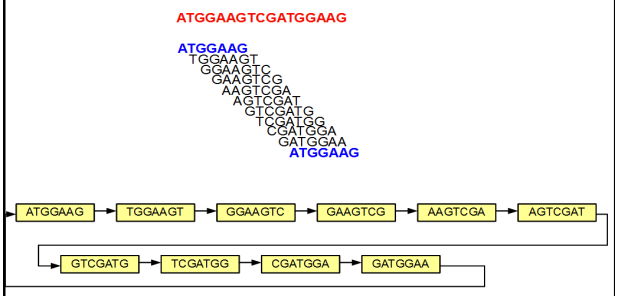
ATGGAAG
TGGAAGT
GGAAGTC
GAAGTCG
AAGTCGC
AGTCGCG
GTCGCGG
TCGCGGA
CGCGGAA
GCGGAAT
CGGAATC

de Bruijn graph



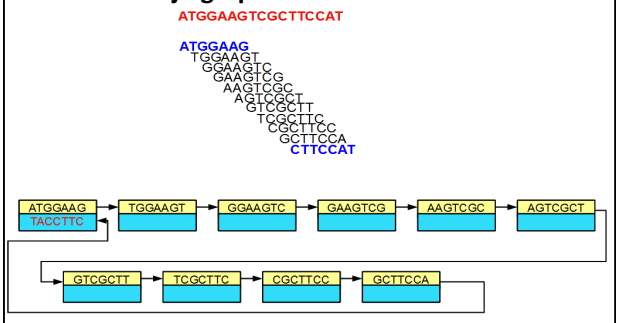
Source: http://www.homolog.us/blogs/2011/07/28/de-bruijn-graphs-i/

A loop in de Bruijn graph



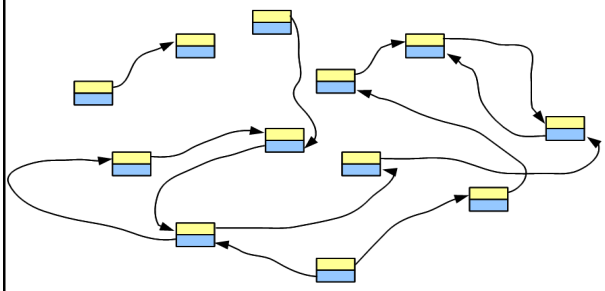
Source: http://www.homolog.us/blogs/2011/07/28/de-bruijn-graphs-i/

de Bruijn graph is double stranded



Source: http://www.homolog.us/blogs/2011/07/28/de-bruijn-graphs-i/

Assembly = golden path through de Bruijn graph



Source: http://www.homolog.us/blogs/2011/07/29/de-bruijn-graphs-ii/

Config File

Keep config file name of the format
conf01-whatever

Run Time

```
$ cd SOAPdenovo/Assembly
$ ./Run_conf01-RL200D.sh
```

Validation

Assembly Metrics

N50 etc

We will use script from Abyss

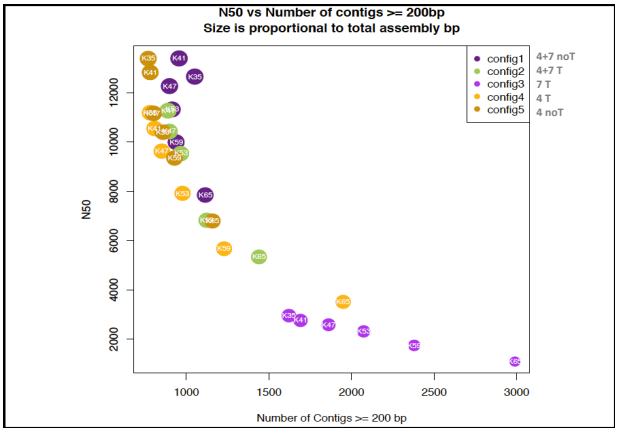
$ abyss-fac -t 200 scaffold.fasta

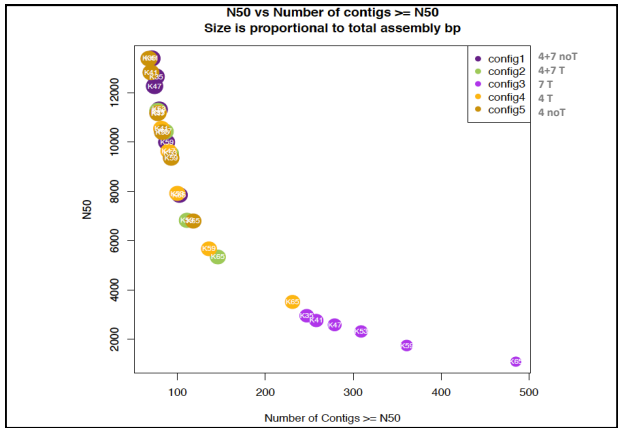
$ cd SOAPdenovo/Assembly/
$ ./get.stats.sh > ../Statistics/stats

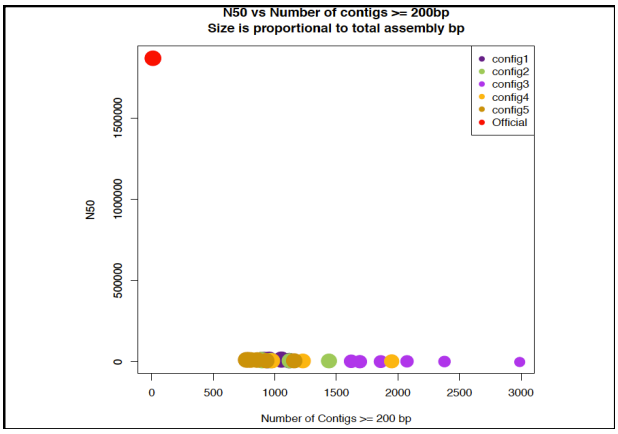
50 assemblies

Different data and configs

$ ls /scratch/cluster/monthly/oribagro/summer2012/SOAPdenovo/Assembly/







Validation

Independent Data

What is in there?

Use available independent sequences:

Conserved gene sets (for eukaryotes http://korflab.ucdavis.edu/Datasets/cegma/)

Sequences in public databases

Independently assembled transcriptome

De Novo Transcriptome

We will use Trinity to assemble the transcriptome

http://trinityrnaseq.sourceforge.net/

Unique portions of alternatively spliced transcripts

Clustering De-brujin graphs

Full-length alternatively spliced transcripts

Add this line to .bashrc

```
$ echo "source /mnt/common/DevTools/DevTools.bashrc" >> .bashrc
```

RNA-Seq Data

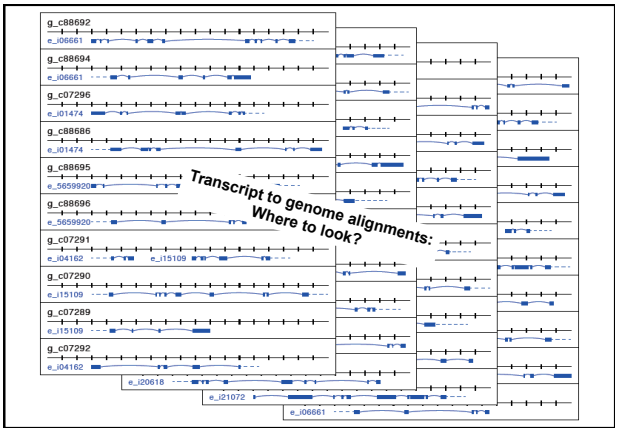
```
$ cd /scratch/cluster/weekly/username
$ ls RNA-seq/Raw/
```

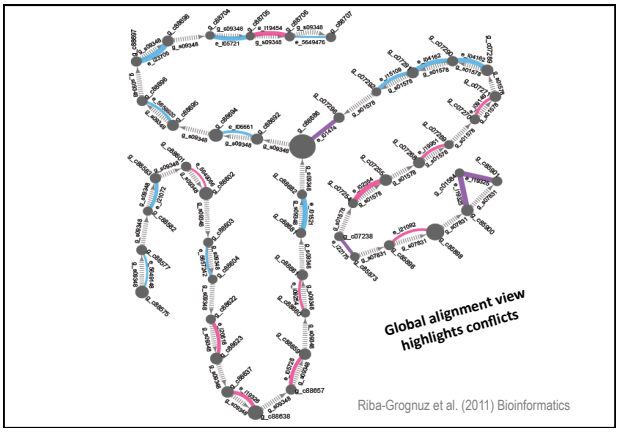
Pick 1 file to run Trinity on it

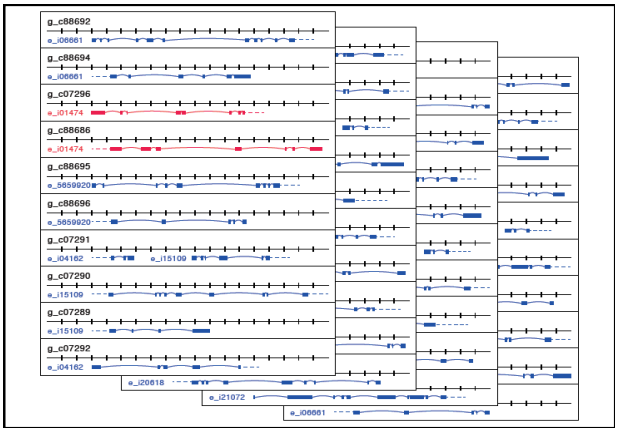
```
$ outDir=Trinity
$ mkdir -p $outDir
$ cd $outDir
$ Trinity.pl -h
```

Cross validation

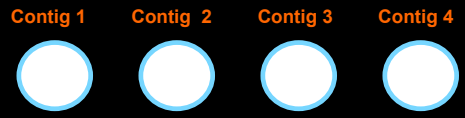
Genome to transcriptome alignments



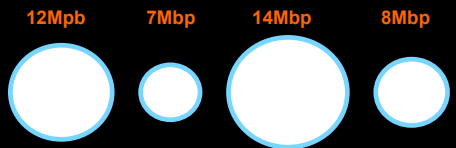




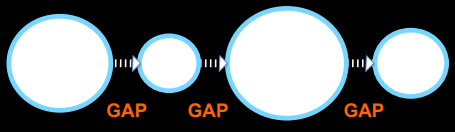
Assembly graphs



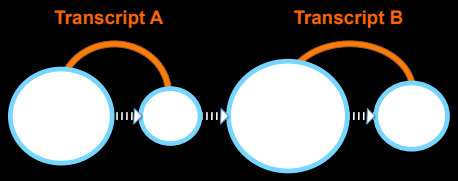
Circle size = Contig size



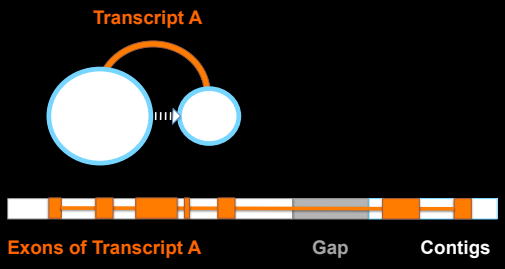
Connect by scaffold



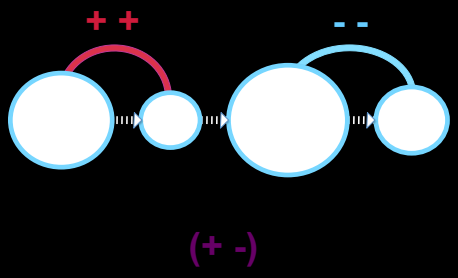
Connect by transcript

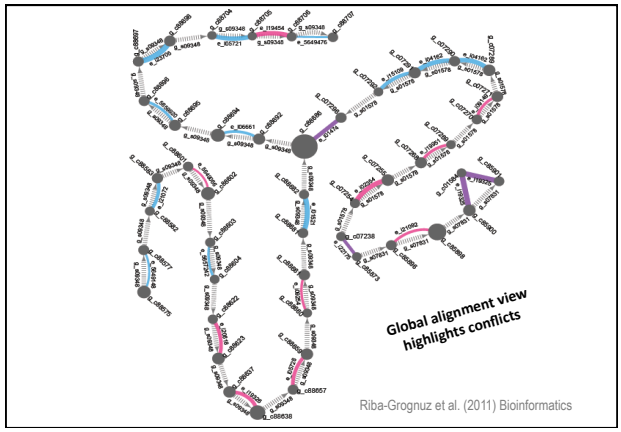


What does this mean?

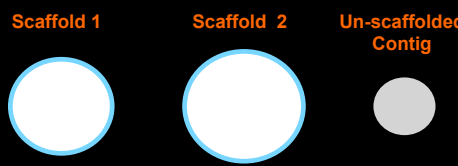


Color by strand



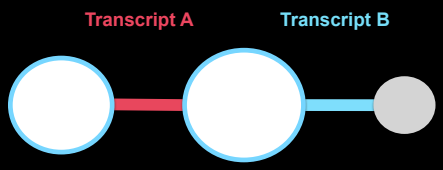


Scaffold Level

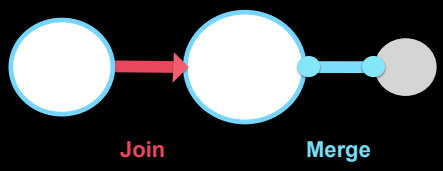


Compact view

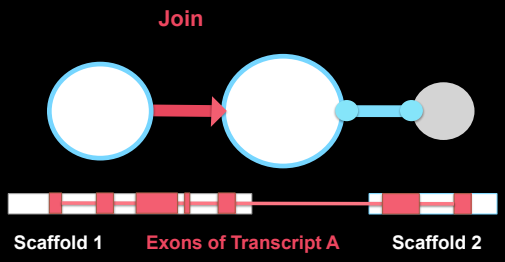
Connect by transcript



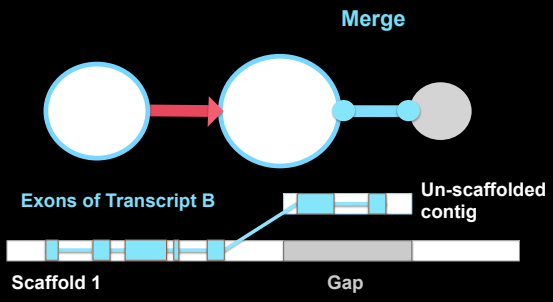
Show connection type



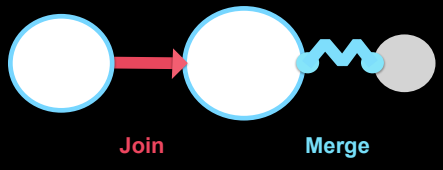
What does this mean?



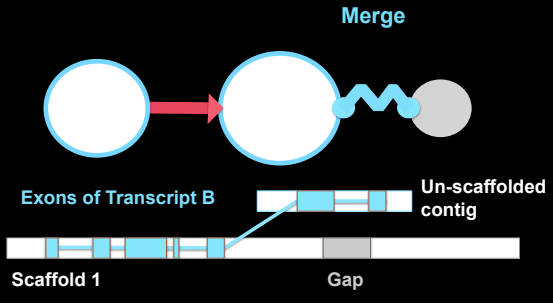
What does this mean?

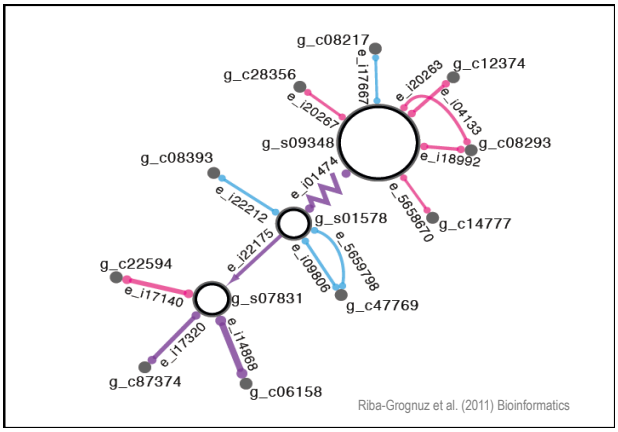


Highlight inconsistencies



What does this mean?




g_c08217
e_i17667
g_c28356
e_i20267
g_s09348
e_i20263
g_c12374
e_i04133
g_c08293
e_i18992
g_c08393
e_i22212
e_i01474
g_s01578
e_5659670
g_c14777
e_i22175
e_i09806
e_5659798
g_c47769
g_c22594
e_i17140
g_s07831
e_i17320
e_i14868
g_c87374
g_c06158
Riba-Grognuz et al. (2011) Bioinformatics