Plateforme Bioinformatique Midi-Pyrénées

# RNA-Seq de novo assembly training Day 3

# Session organisation: Day 3

## Morning:

- Assembly quality common problems
  - Simple cleaning
  - Frame-shifts
  - Chimeras
- Assembly quality assessment using biological knowledge
  - CEGMA
  - Close reference

## Afternoon:

- Example of assembly pipeline
- Meta-assembly
- Contigs to unigenes
- Publishing your transcriptome in TSA

# Objectives for this third day

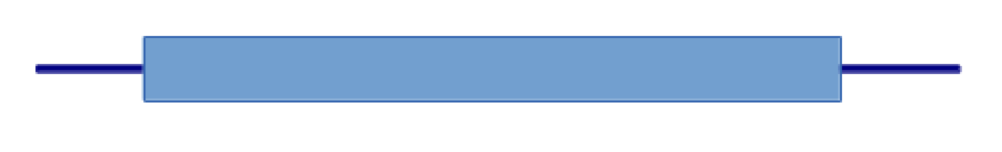
Answer the following questions :

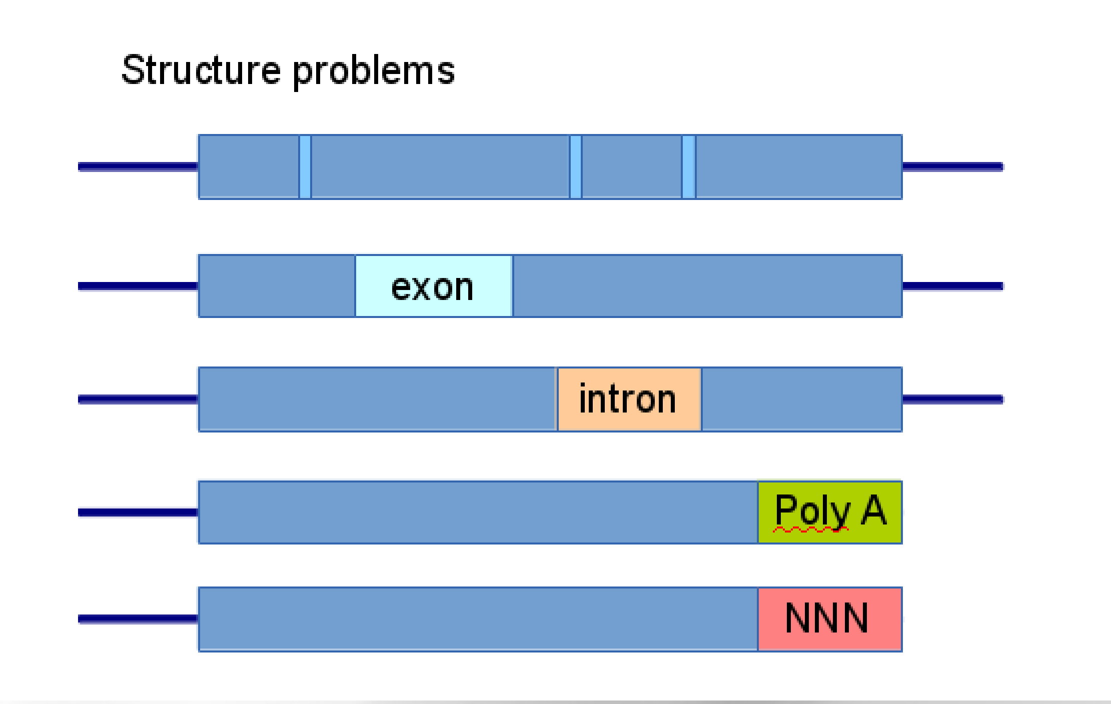
- What are the common errors found in the assemblies?
- How do I get rid of those errors?
- How do I validate my assemblies?
- How do I choose the best assembly?
- How to merge assemblies?

# What are the classical errors found in the contigs?

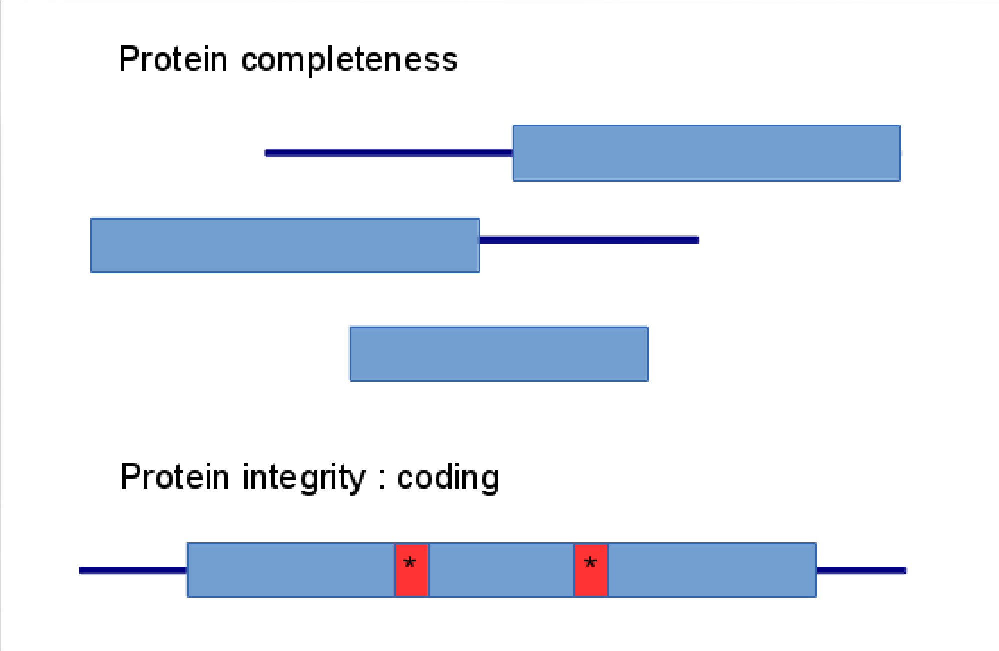
# Common errors

Ideal contig



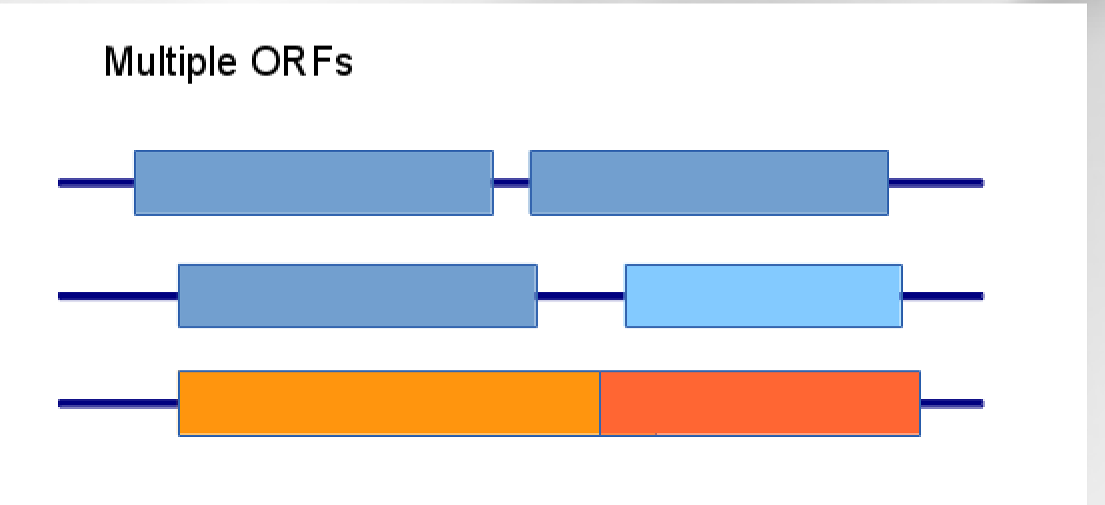


# ORF errors



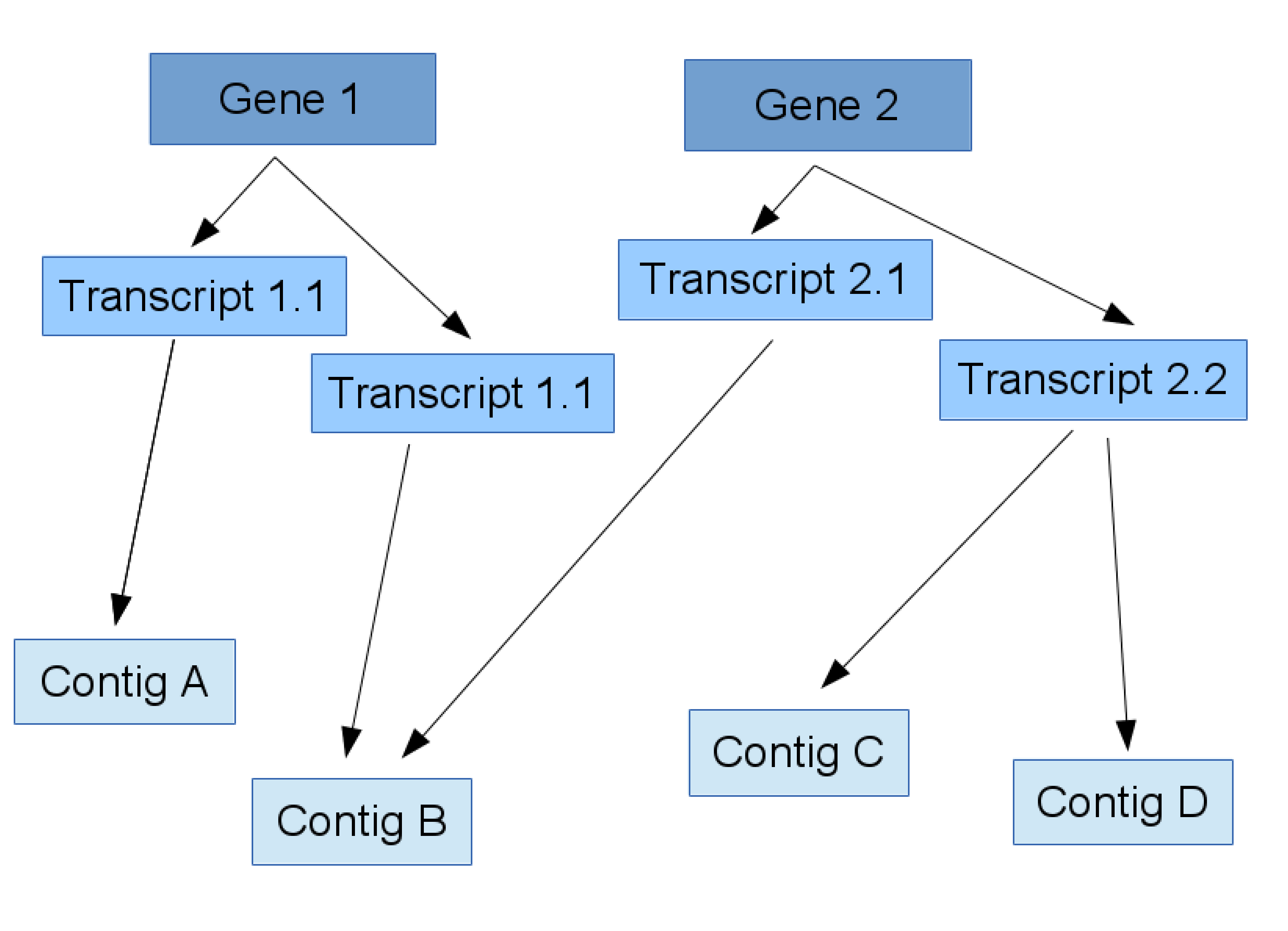
Protein completeness

Protein integrity : coding



Multiple ORFs

# From genes to contigs



# How do we clean our transcriptome assemblies?

# Classical cleaning steps

- cleaning polyA tails, terminal N blocks, low complexity areas
- cis or trans-chimera detection
- insertion/deletion correction using the alignment
- low fold coverage filtering (graph data)
- low expression filtering
- possible filtering of contigs which do not have a long enough ORF (phylogenomy)

# Simple cleaning steps

Remove remaining polyA tails

Remove blocks of Ns located at the extremities

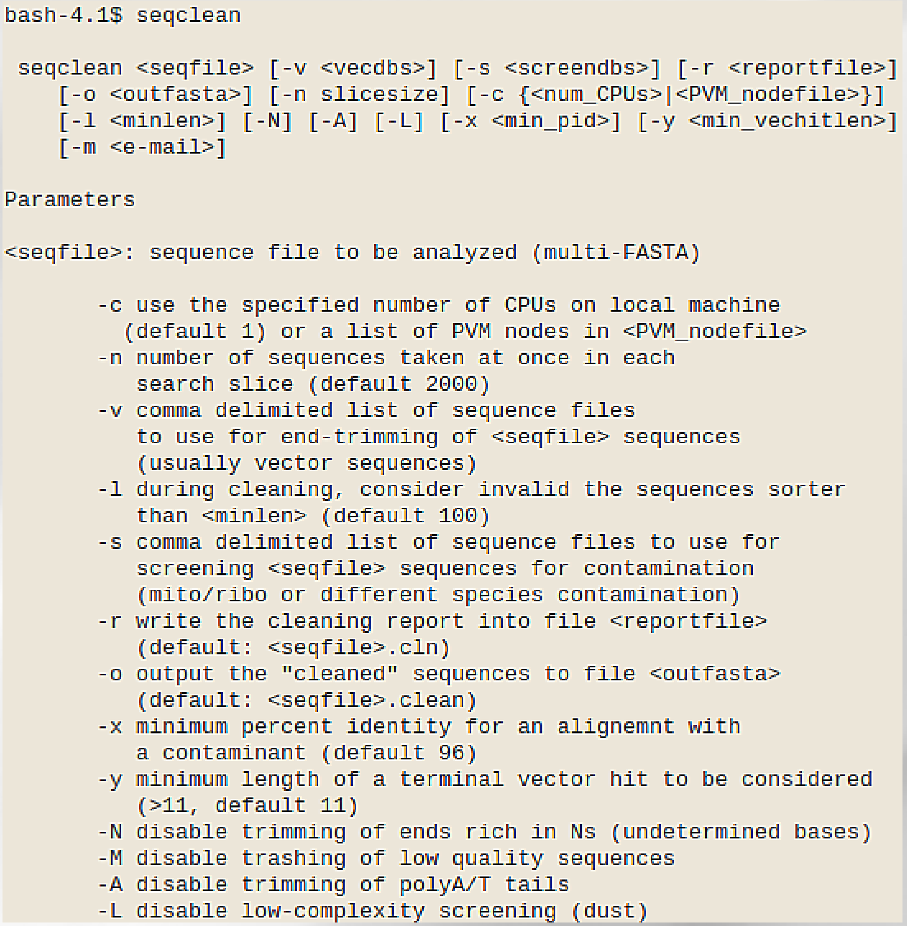
Remove low complexity areas

# Seqclean

**Seqclean**: a script for automated trimming and validation of ESTs or other DNA sequences by screening for various contaminants, low quality and low-complexity sequences.

http://compbio.dfci.harvard.edu/tgi/software

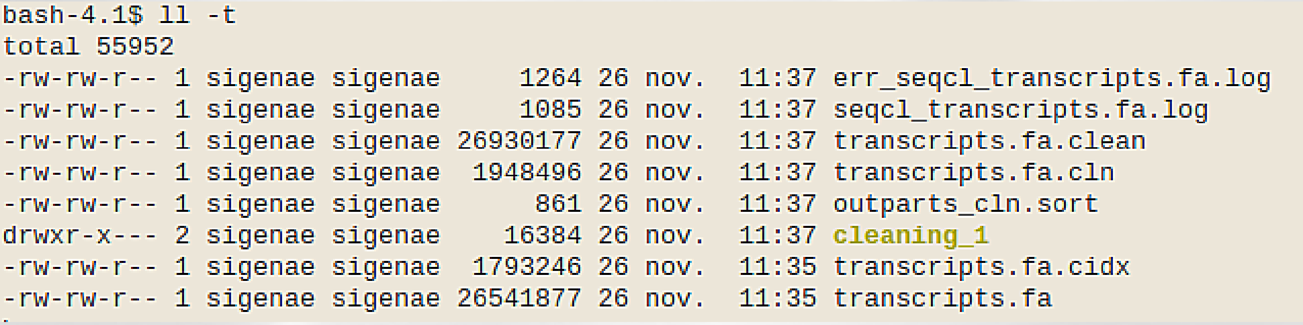
# Seqclean: command line



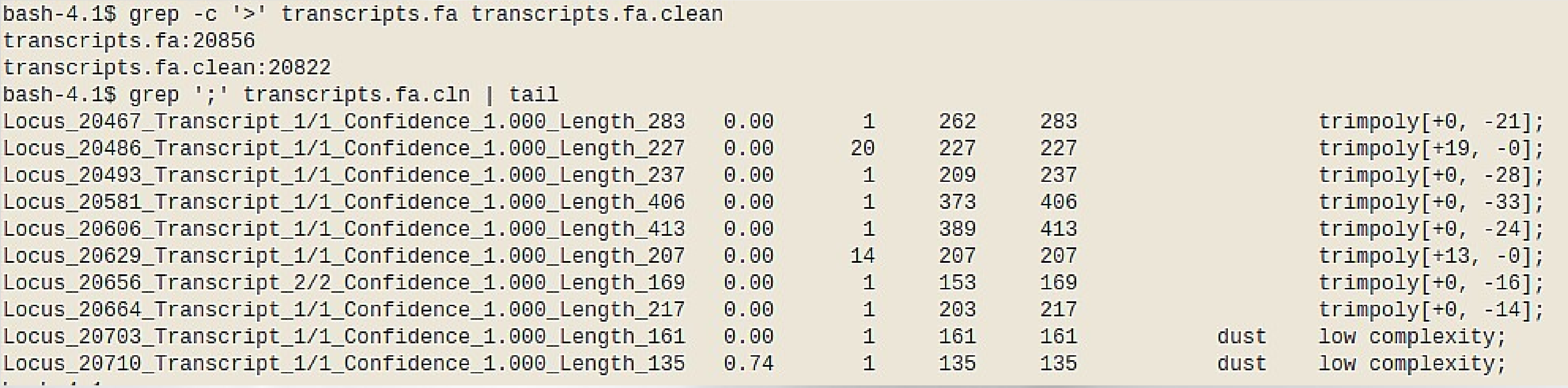
```
bash-4.1$ seqclean

 seqclean <seqfile> [-v <vecdbs>] [-s <screendbs>] [-r <reportfile>]
    [-o <outfasta>] [-n slicesize] [-c {<num_CPUs>|<PVM_nodefile>}]
    [-l <minlen>] [-N] [-A] [-L] [-x <min_pid>] [-y <min_vechitlen>]
    [-m <e-mail>]

Parameters

<seqfile>: sequence file to be analyzed (multi-FASTA)

       -c use the specified number of CPUs on local machine
         (default 1) or a list of PVM nodes in <PVM_nodefile>
       -n number of sequences taken at once in each
          search slice (default 2000)
       -v comma delimited list of sequence files
          to use for end-trimming of <seqfile> sequences
          (usually vector sequences)
       -l during cleaning, consider invalid the sequences sorter
          than <minlen> (default 100)
       -s comma delimited list of sequence files to use for
          screening <seqfile> sequences for contamination
          (mito/ribo or different species contamination)
       -r write the cleaning report into file <reportfile>
          (default: <seqfile>.cln)
       -o output the "cleaned" sequences to file <outfasta>
          (default: <seqfile>.clean)
       -x minimum percent identity for an alignemnt with
          a contaminant (default 96)
       -y minimum length of a terminal vector hit to be considered
          (>11, default 11)
       -N disable trimming of ends rich in Ns (undetermined bases)
       -M disable trashing of low quality sequences
       -A disable trimming of polyA/T tails
       -L disable low-complexity screening (dust)
```

```
seqclean input.fa -o input.fa.clean
```

# Seqclean: output



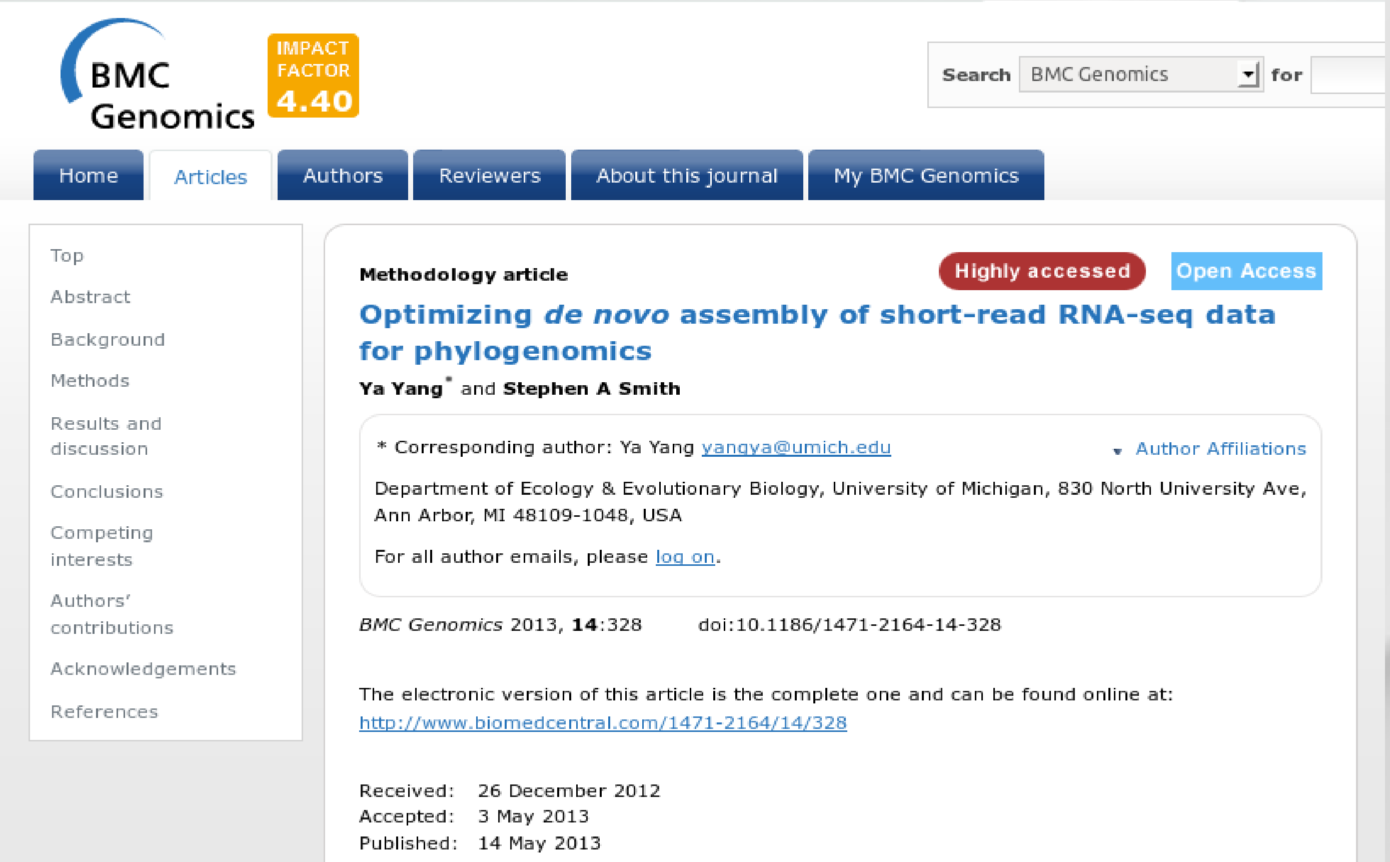
```
bash-4.1$ ll -t
total 55952
-rw-rw-r-- 1 sigenae sigenae     1264 26 nov.  11:37 err_seqcl_transcripts.fa.log
-rw-rw-r-- 1 sigenae sigenae     1085 26 nov.  11:37 seqcl_transcripts.fa.log
-rw-rw-r-- 1 sigenae sigenae 26930177 26 nov.  11:37 transcripts.fa.clean
-rw-rw-r-- 1 sigenae sigenae  1948496 26 nov.  11:37 transcripts.fa.cln
-rw-rw-r-- 1 sigenae sigenae      861 26 nov.  11:37 outparts_cln.sort
drwxr-x--- 2 sigenae sigenae    16384 26 nov.  11:37 cleaning_1
-rw-rw-r-- 1 sigenae sigenae  1793246 26 nov.  11:35 transcripts.fa.cidx
-rw-rw-r-- 1 sigenae sigenae 26541877 26 nov.  11:35 transcripts.fa
```



```
bash-4.1$ grep -c '>' transcripts.fa transcripts.fa.clean
transcripts.fa:20856
transcripts.fa.clean:20822
bash-4.1$ grep ';' transcripts.fa.cln | tail
Locus_20467_Transcript_1/1_Confidence_1.000_Length_283  0.00     1    262    283                 trimpoly[+0, -21];
Locus_20486_Transcript_1/1_Confidence_1.000_Length_227  0.00    20    227    227                 trimpoly[+19, -0];
Locus_20493_Transcript_1/1_Confidence_1.000_Length_237  0.00     1    209    237                 trimpoly[+0, -28];
Locus_20581_Transcript_1/1_Confidence_1.000_Length_406  0.00     1    373    406                 trimpoly[+0, -33];
Locus_20606_Transcript_1/1_Confidence_1.000_Length_413  0.00     1    389    413                 trimpoly[+0, -24];
Locus_20629_Transcript_1/1_Confidence_1.000_Length_207  0.00    14    207    207                 trimpoly[+13, -0];
Locus_20656_Transcript_2/2_Confidence_1.000_Length_169  0.00     1    153    169                 trimpoly[+0, -16];
Locus_20664_Transcript_1/1_Confidence_1.000_Length_217  0.00     1    203    217                 trimpoly[+0, -14];
Locus_20703_Transcript_1/1_Confidence_1.000_Length_161  0.00     1    161    161          dust   low complexity;
Locus_20710_Transcript_1/1_Confidence_1.000_Length_135  0.74     1    135    135          dust   low complexity;
```

```
seqclean transcripts.fa -o transcripts.fa.clean
```

# Chimeras



Methodology article

Optimizing *de novo* assembly of short-read RNA-seq data for phylogenomics

**Ya Yang*** and **Stephen A Smith**

* Corresponding author: Ya Yang yangya@umich.edu

Department of Ecology & Evolutionary Biology, University of Michigan, 830 North University Ave, Ann Arbor, MI 48109-1048, USA

For all author emails, please log on.

*BMC Genomics* 2013, **14**:328 doi:10.1186/1471-2164-14-328

The electronic version of this article is the complete one and can be found online at: http://www.biomedcentral.com/1471-2164/14/328

Received: 26 December 2012
Accepted: 3 May 2013
Published: 14 May 2013

Chimera typing and removal

# How important is the phenomenon?

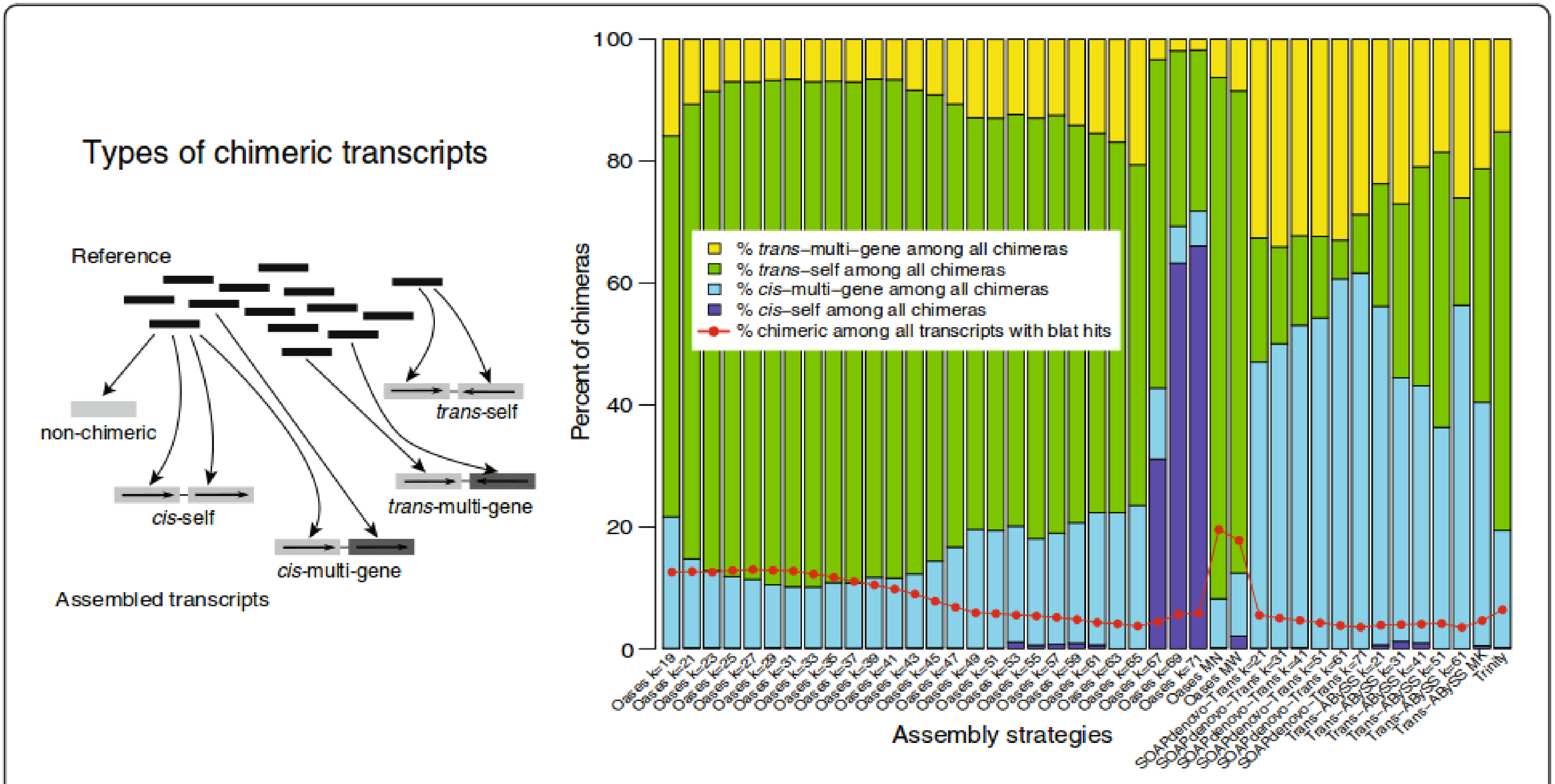


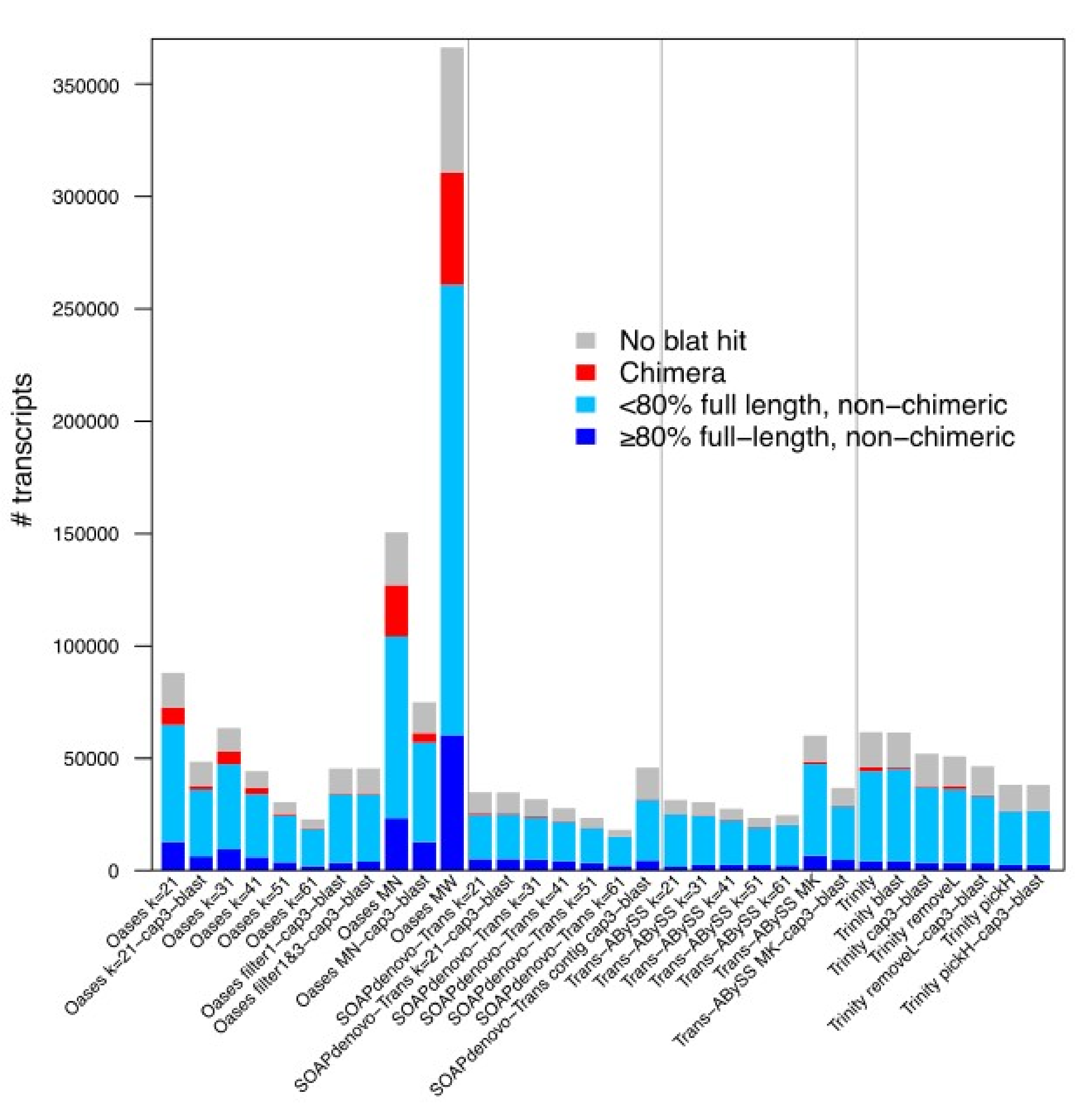
**Figure 1 Chimera compositions among assembled transcripts before post-processing.** Oases MN: Oases-M merging single *k*-mer assemblies of 21, 31, 41, 51 and 61; MW: Oases-M merging single *k*-mer assemblies of 19–71, with increment of 2; Trans-ABySS MK: Trans-ABySS merging single *k*-mer assemblies of 21, 31, 41, 51 and 61.

Majority of trans-self chimeras for small-middle k-mers

Majority of cis-self chimeras for large k-mers and oases merge

Without reference, cannot tackle multi-gene chimeras

# How important is the phenomenon?



Chimera rate is low with small k-mers, residual with middle-large ones

Chimera rate increases with oases merge procedure

# Chimera detection

Self chimera detection: each contig is aligned vs itself.

If several HSPs are produced then the contig is split in the middle of locations.

In house script having one input:

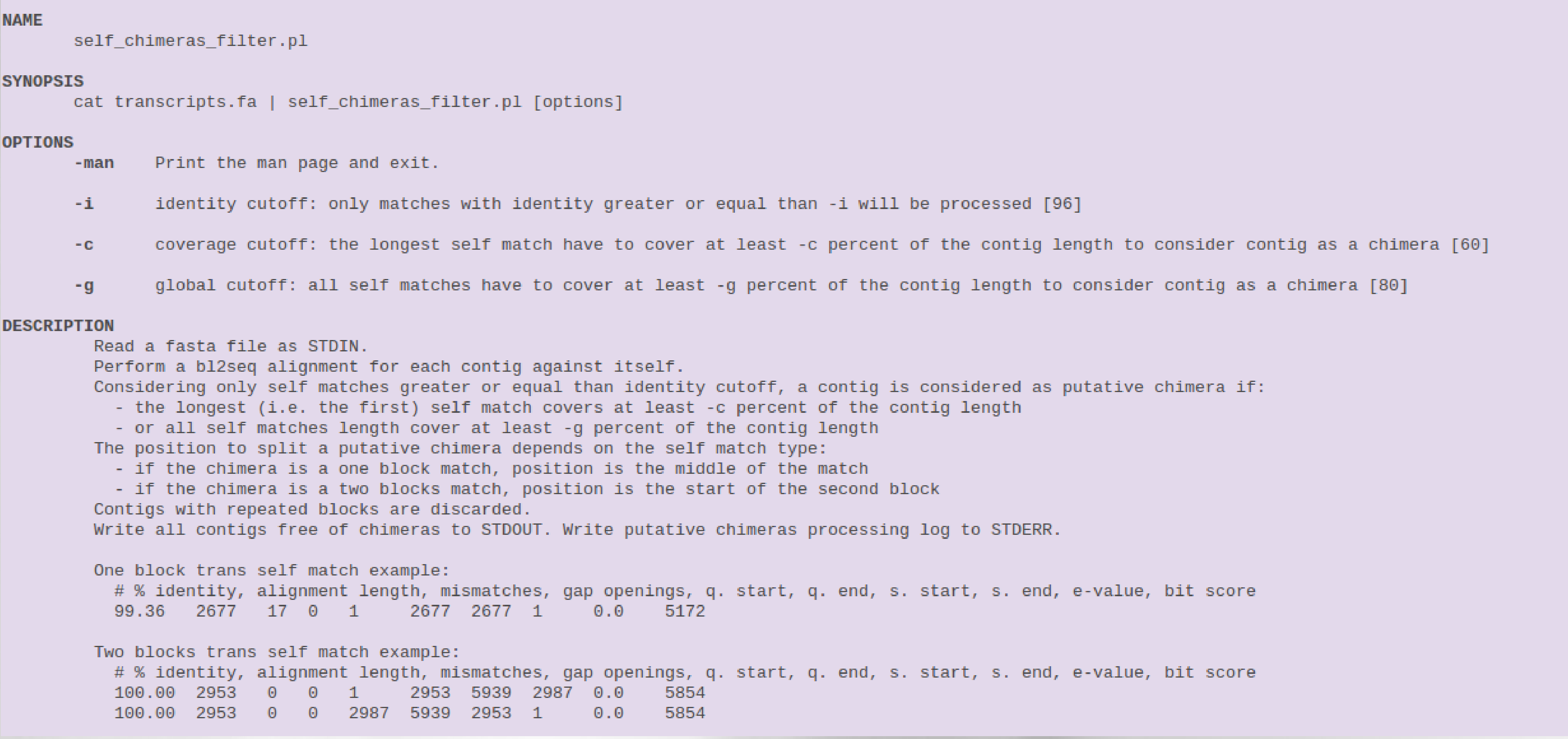
- contig fasta file

And one output:

- chimera free contig fasta file

Frequency: around 1‰

# Chimera detection script



```
NAME
       self_chimeras_filter.pl

SYNOPSIS
       cat transcripts.fa | self_chimeras_filter.pl [options]

OPTIONS
       -man    Print the man page and exit.

       -i      identity cutoff: only matches with identity greater or equal than -i will be processed [96]

       -c      coverage cutoff: the longest self match have to cover at least -c percent of the contig length to consider contig as a chimera [60]

       -g      global cutoff: all self matches have to cover at least -g percent of the contig length to consider contig as a chimera [80]

DESCRIPTION
         Read a fasta file as STDIN.
         Perform a bl2seq alignment for each contig against itself.
         Considering only self matches greater or equal than identity cutoff, a contig is considered as putative chimera if:
           - the longest (i.e. the first) self match covers at least -c percent of the contig length
           - or all self matches length cover at least -g percent of the contig length
         The position to split a putative chimera depends on the self match type:
           - if the chimera is a one block match, position is the middle of the match
           - if the chimera is a two blocks match, position is the start of the second block
         Contigs with repeated blocks are discarded.
         Write all contigs free of chimeras to STDOUT. Write putative chimeras processing log to STDERR.

         One block trans self match example:
           # % identity, alignment length, mismatches, gap openings, q. start, q. end, s. start, s. end, e-value, bit score
           99.36   2677    17   0    1     2677  2677  1     0.0    5172

         Two blocks trans self match example:
           # % identity, alignment length, mismatches, gap openings, q. start, q. end, s. start, s. end, e-value, bit score
           100.00  2953    0    0    1     2953  5939  2987  0.0    5854
           100.00  2953    0    0    2987  5939  2953  1     0.0    5854
```

```
cat transcripts.fa | self_chimeras_filter.pl > transcripts.chim_free.fa
```

# Frame-shifts

Finding frame-shifts :

- using the RMBT alignment to find INDEL
- using a proteic reference to find frame-shifts

# Insertion/deletion correction

Using the majority vote at each position of the alignment.

In house script having two inputs:

- reference contig fasta file
- mpileup output (from bam alignment file)

And one output:

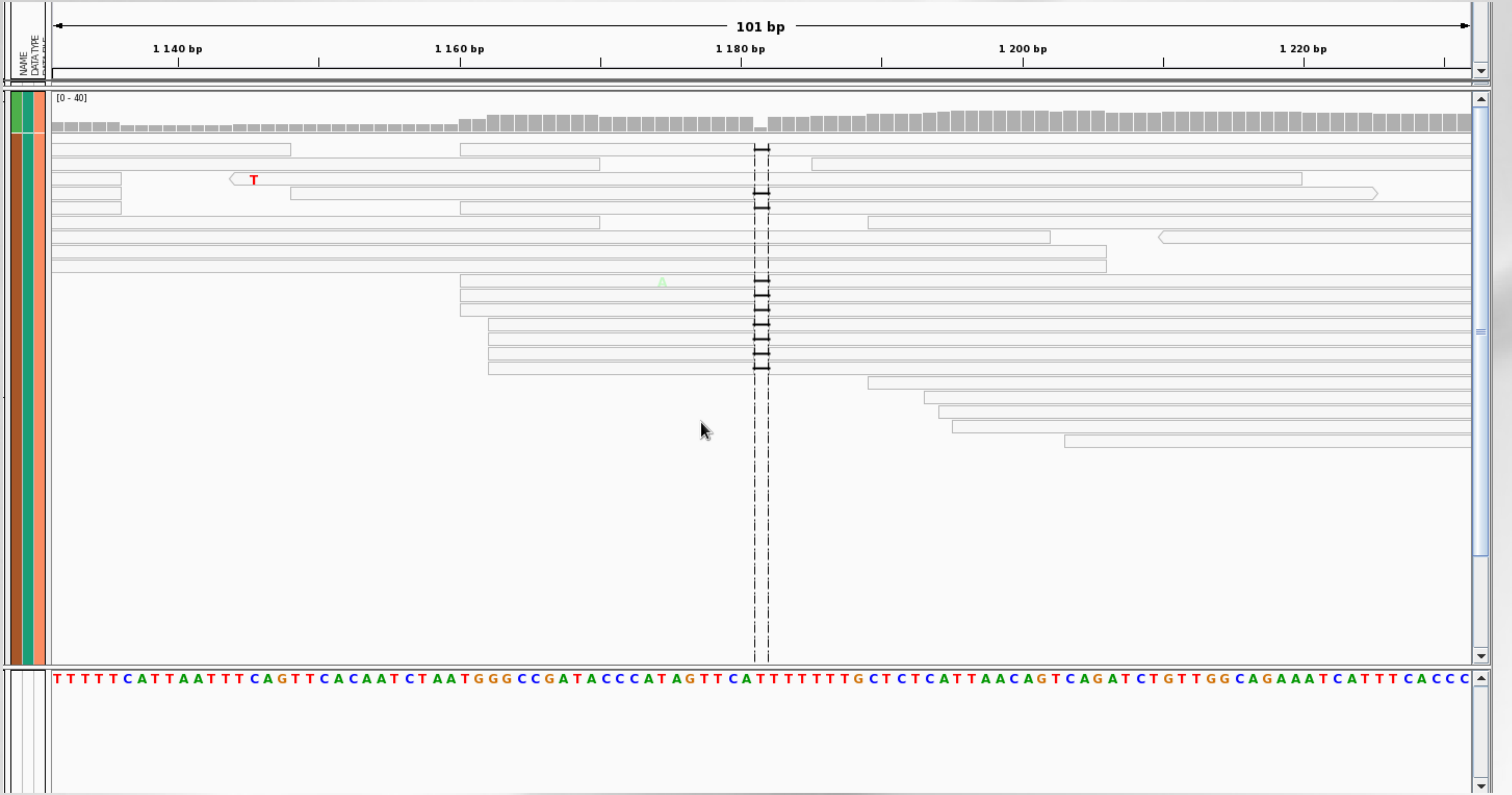
- corrected reference fasta file

Frequency:

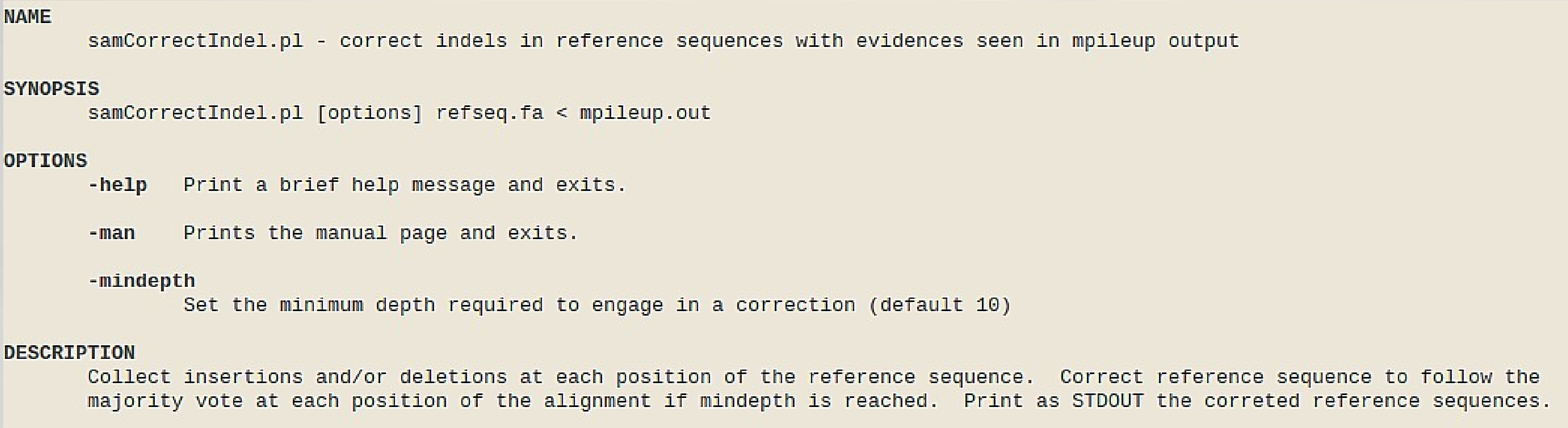
- 5% contigs
- 1-2 corrections/contigs

# Insertion/deletion correction

Locus_9_Transcript_38: remove T in position 1181 (10/14)


NAME
DATA TYPE
101 bp
1 140 bp
1 160 bp
1 180 bp
1 200 bp
1 220 bp
[0 - 40]
T
A
TTTTTCATTAATTTCAGTTCACAATCTAATGGGCCGATACCCATAGTTCATTTTTTTGCTCTCATTAACAGTCAGATCTGTTGGCAGAAATCATTTCACCC

# Indels correction script

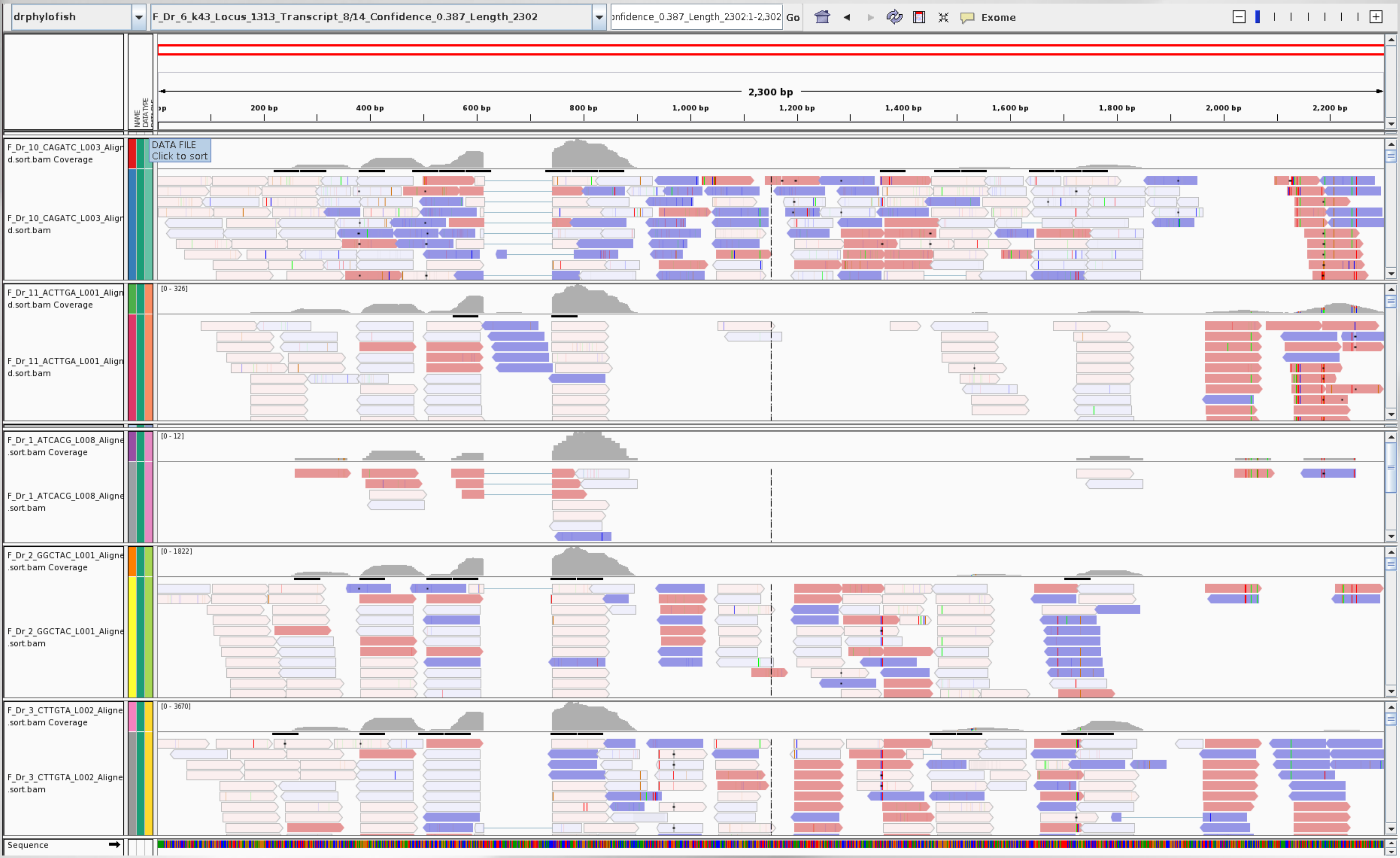


```
NAME
       samCorrectIndel.pl - correct indels in reference sequences with evidences seen in mpileup output

SYNOPSIS
       samCorrectIndel.pl [options] refseq.fa < mpileup.out

OPTIONS
       -help   Print a brief help message and exits.

       -man    Prints the manual page and exits.

       -mindepth
               Set the minimum depth required to engage in a correction (default 10)

DESCRIPTION
       Collect insertions and/or deletions at each position of the reference sequence.  Correct reference sequence to follow the
       majority vote at each position of the alignment if mindepth is reached.  Print as STDOUT the correted reference sequences.
```

```
samtools mpileup -f transcripts.fa reads_to_transcripts.bam | \
 samCorrectIndel.pl transcripts.fa > transcripts.indel_free.fa
```

# How do we detect splice forms within contigs?

# Intron retention



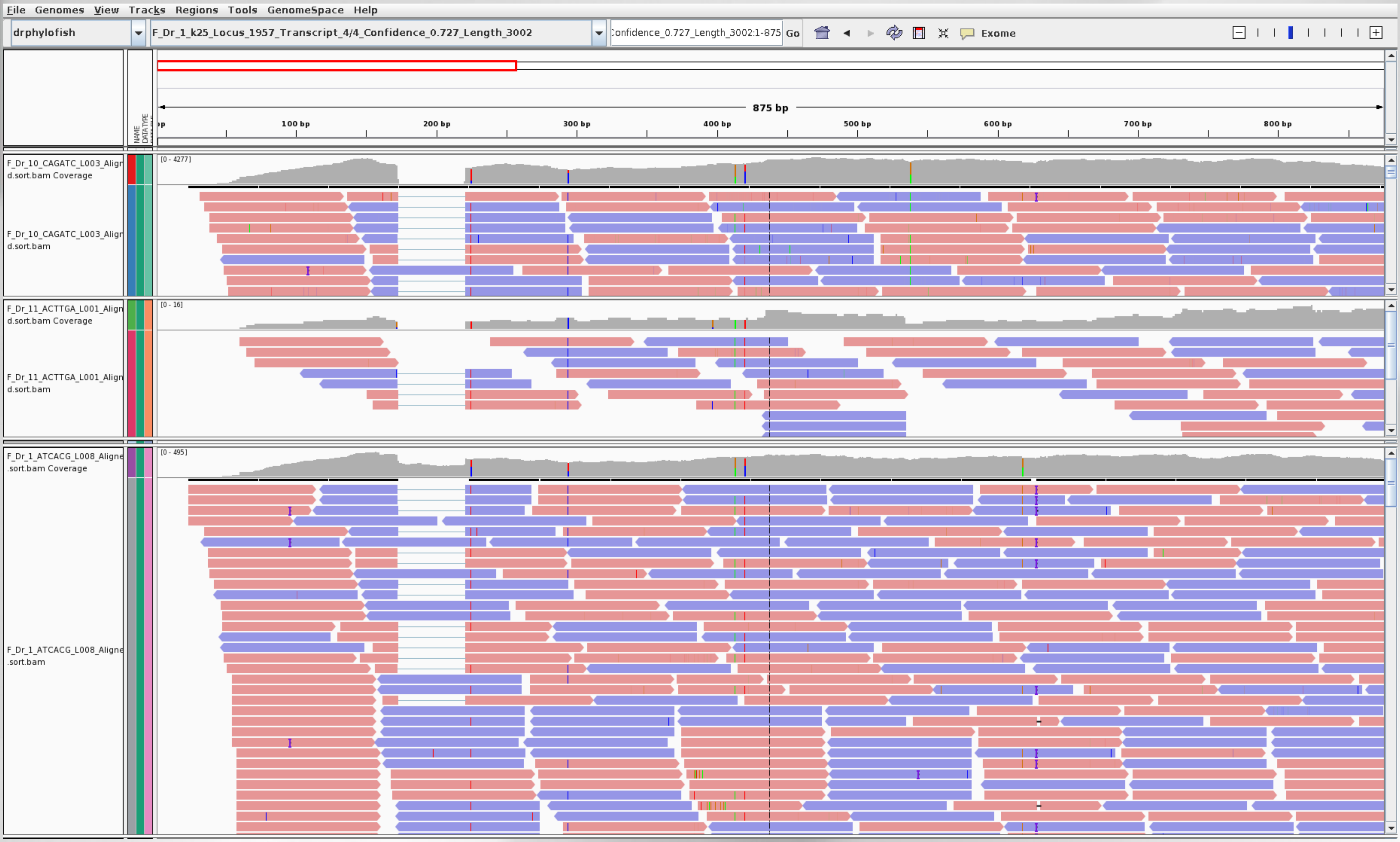
drphylofish

F_Dr_6_k43_Locus_1313_Transcript_8/14_Confidence_0.387_Length_2302

›nfidence_0.387_Length_2302:1-2,302

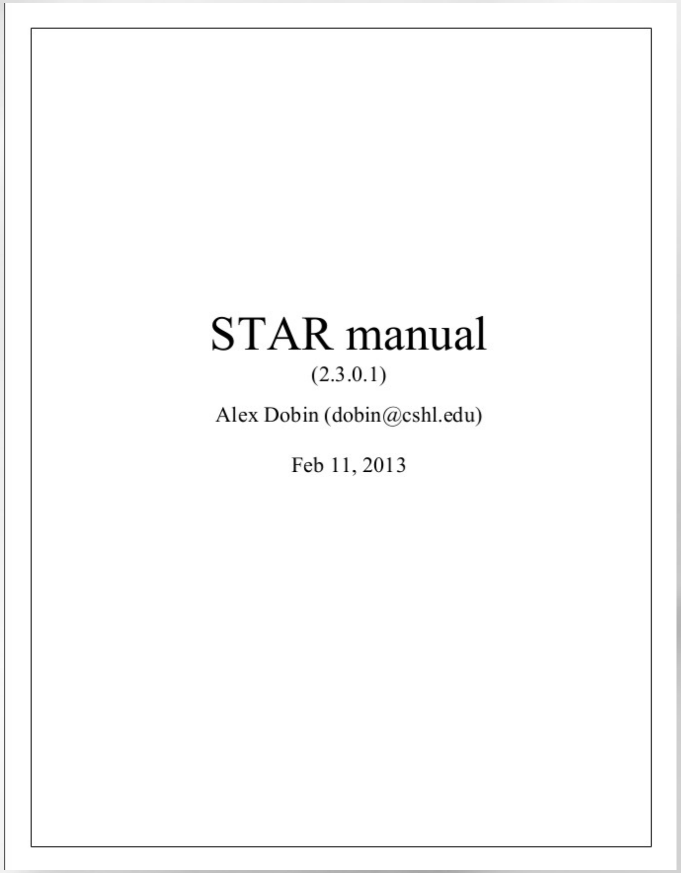
Go
Exome
2,300 bp
›p
200 bp
400 bp
600 bp
800 bp
1,000 bp
1,200 bp
1,400 bp
1,600 bp
1,800 bp
2,000 bp
2,200 bp
NAME
DATA TYPE
DATA FILE
Click to sort
F_Dr_10_CAGATC_L003_Align d.sort.bam Coverage
F_Dr_10_CAGATC_L003_Align d.sort.bam
F_Dr_11_ACTTGA_L001_Align d.sort.bam Coverage
[0 - 326]
F_Dr_11_ACTTGA_L001_Align d.sort.bam
F_Dr_1_ATCACG_L008_Aligne .sort.bam Coverage
[0 - 12]
F_Dr_1_ATCACG_L008_Aligne .sort.bam
F_Dr_2_GGCTAC_L001_Aligne .sort.bam Coverage
[0 - 1822]
F_Dr_2_GGCTAC_L001_Aligne .sort.bam
F_Dr_3_CTTGTA_L002_Aligne .sort.bam Coverage
[0 - 3670]
F_Dr_3_CTTGTA_L002_Aligne .sort.bam
Sequence

# Splice variant


File
Genomes
View
Tracks
Regions
Tools
GenomeSpace
Help
drphylofish
F_Dr_1_k25_Locus_1957_Transcript_4/4_Confidence_0.727_Length_3002
:onfidence_0.727_Length_3002:1-875
Go
Exome
875 bp
100 bp
200 bp
300 bp
400 bp
500 bp
600 bp
700 bp
800 bp
NAME
DATA TYPE
F_Dr_10_CAGATC_L003_Aligned.sort.bam Coverage
[0 - 4277]
F_Dr_10_CAGATC_L003_Aligned.sort.bam
F_Dr_11_ACTTGA_L001_Aligned.sort.bam Coverage
[0 - 16]
F_Dr_11_ACTTGA_L001_Aligned.sort.bam
F_Dr_1_ATCACG_L008_Aligned.sort.bam Coverage
[0 - 495]
F_Dr_1_ATCACG_L008_Aligned.sort.bam

# STAR

```
STAR --runMode genomeGenerate --genomeDir STAR --genomeFastaFiles transcripts.fa
STAR --genomeDir STAR --readFilesIn R1.fastq.gz R2.fastq.gz --readFilesCommand zcat
```



STAR manual

(2.3.0.1)

Alex Dobin (dobin@cshl.edu)

Feb 11, 2013

# Exercises

# Exercise n°4

# How biologically relevant are our contigs in the end?

# Phylogenomics

Genes are transmitted during the evolution

Some genes are present in all organisms

↳ small subset which can be used in any case

Most genes are conserved in organisms having a close common ancestor. The closer:

- the large is the set
- the more the comparison with our assembly will be meaningful

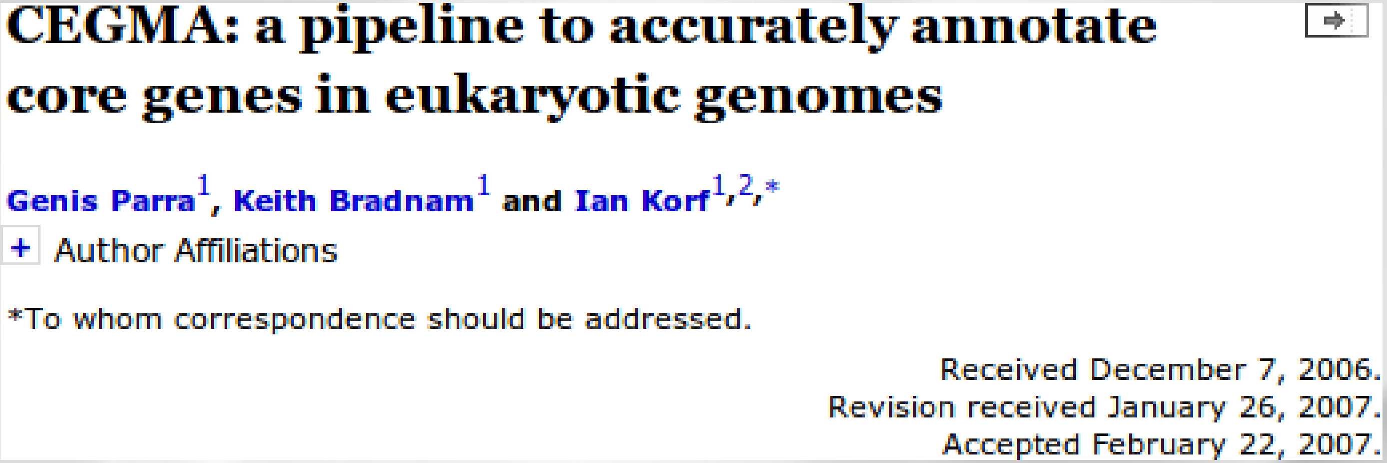
# From contigs to unigenes

When analyzing protein coding genes biologists often require one representative ORF for a protein.

- splitting contigs with multiple non overlapping ORF
- using a reference (anchor)

# CEGMA

- Core Eukaryotic Genes Mapping Approach



**CEGMA: a pipeline to accurately annotate core genes in eukaryotic genomes**

Genis Parra[1], Keith Bradnam[1] and Ian Korf[1,2,*]

+ Author Affiliations

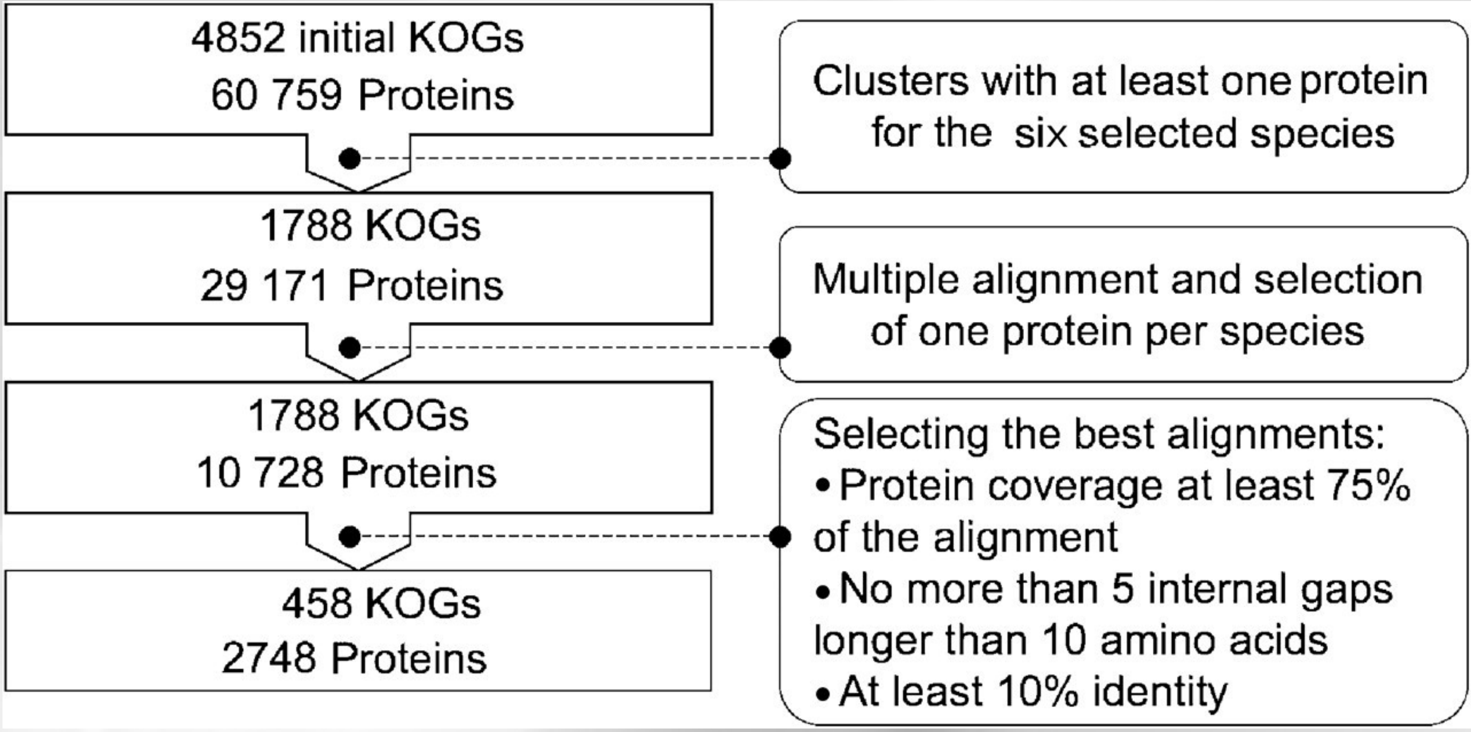
*To whom correspondence should be addressed.

Received December 7, 2006.
Revision received January 26, 2007.
Accepted February 22, 2007.

- Mapping a set of conserved protein families that occur in a wide range of eukaryotes onto assembly to assess completeness

# CEGMA

- A set of eukaryotic core proteins (KOG = euKaryotic Orthologous Groups) from 6 species: H. sapiens, D. melanogaster, C. elegans, A. thaliana, S. cerevisiae, S.pombe



4852 initial KOGs
60 759 Proteins

Clusters with at least one protein for the six selected species

1788 KOGs
29 171 Proteins

Multiple alignment and selection of one protein per species

1788 KOGs
10 728 Proteins

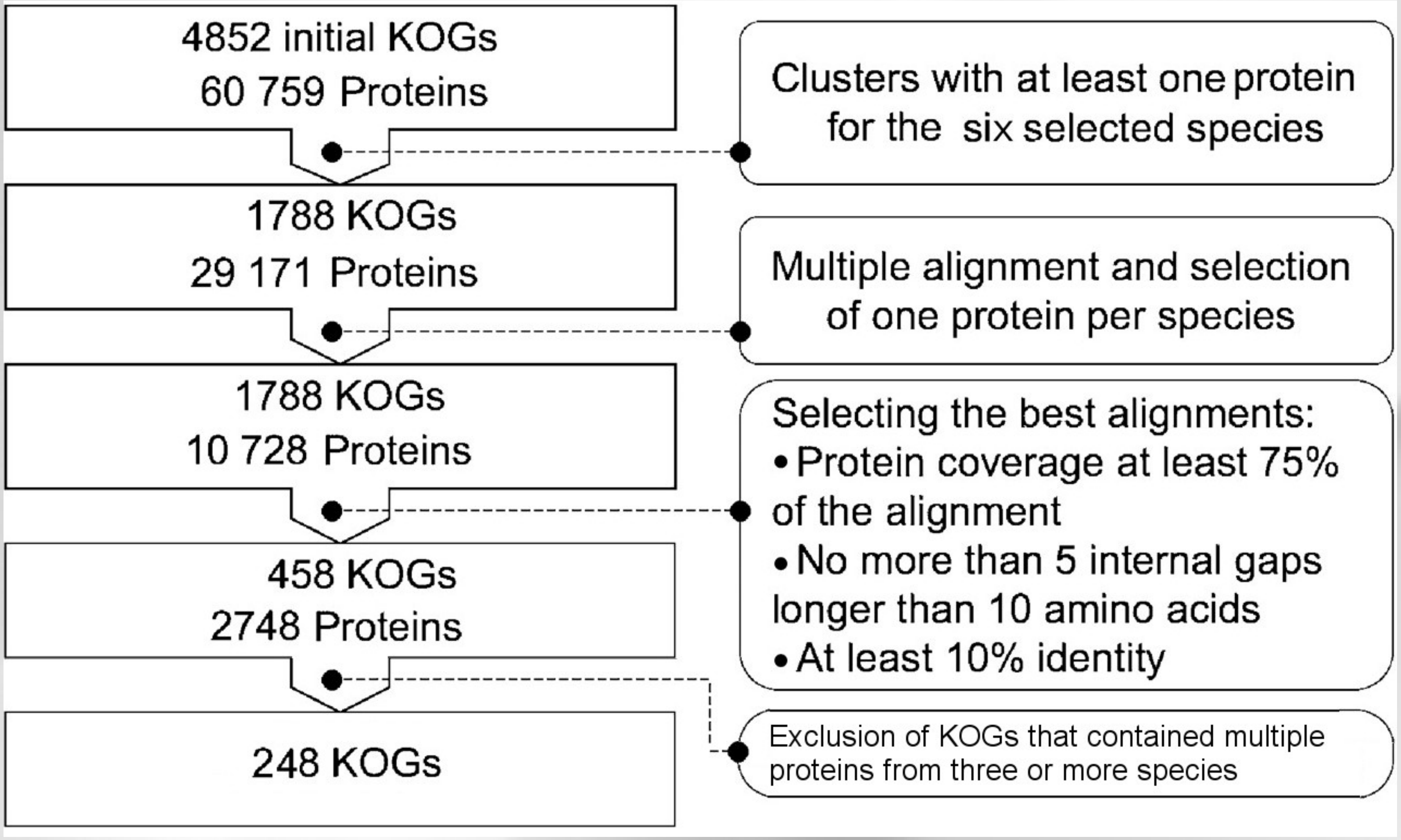
Selecting the best alignments:
- Protein coverage at least 75% of the alignment
- No more than 5 internal gaps longer than 10 amino acids
- At least 10% identity

458 KOGs
2748 Proteins

- Set of proteins finally contains 458 groups (2748 proteins)

# CEGMA

- A set of eukaryotic core proteins with less paralogs for draft genome and transcriptome



4852 initial KOGs
60 759 Proteins

Clusters with at least one protein for the six selected species

1788 KOGs
29 171 Proteins

Multiple alignment and selection of one protein per species

1788 KOGs
10 728 Proteins

Selecting the best alignments:
- Protein coverage at least 75% of the alignment
- No more than 5 internal gaps longer than 10 amino acids
- At least 10% identity

458 KOGs
2748 Proteins

Exclusion of KOGs that contained multiple proteins from three or more species

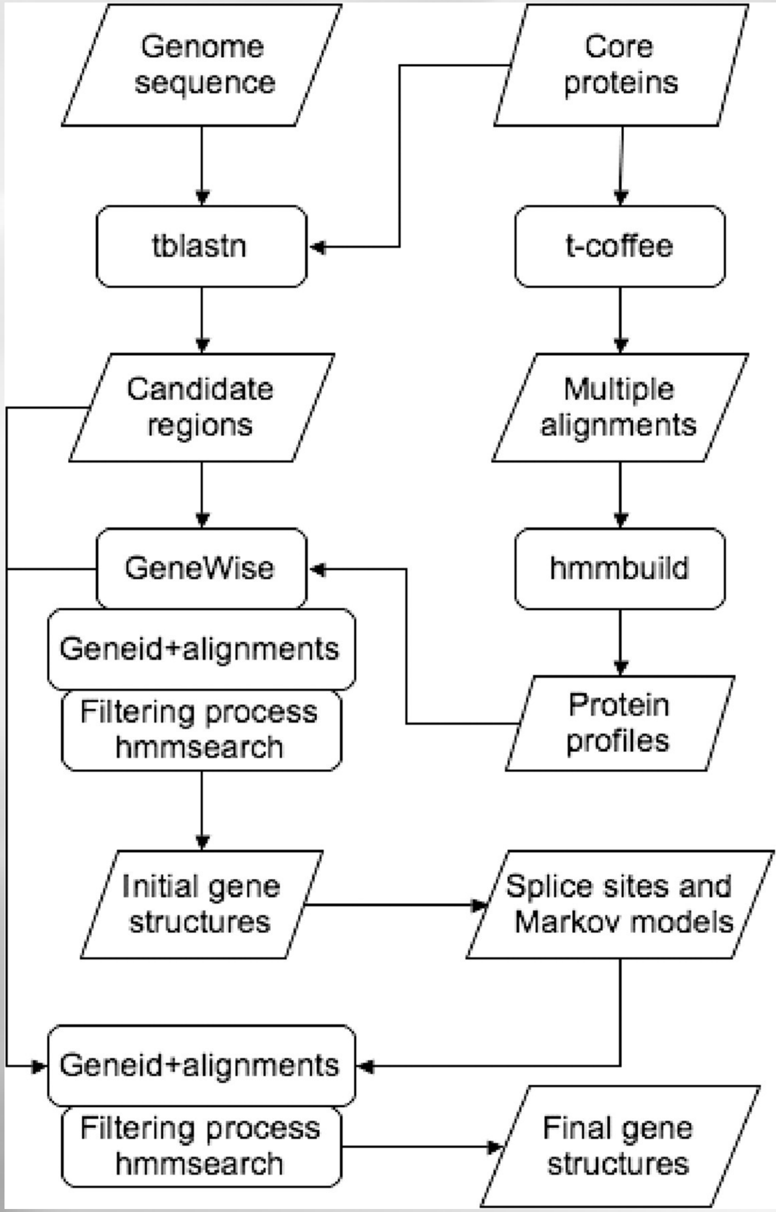
248 KOGs

⤷ set of 248 CEGs (Core Eukaryotic Genes)

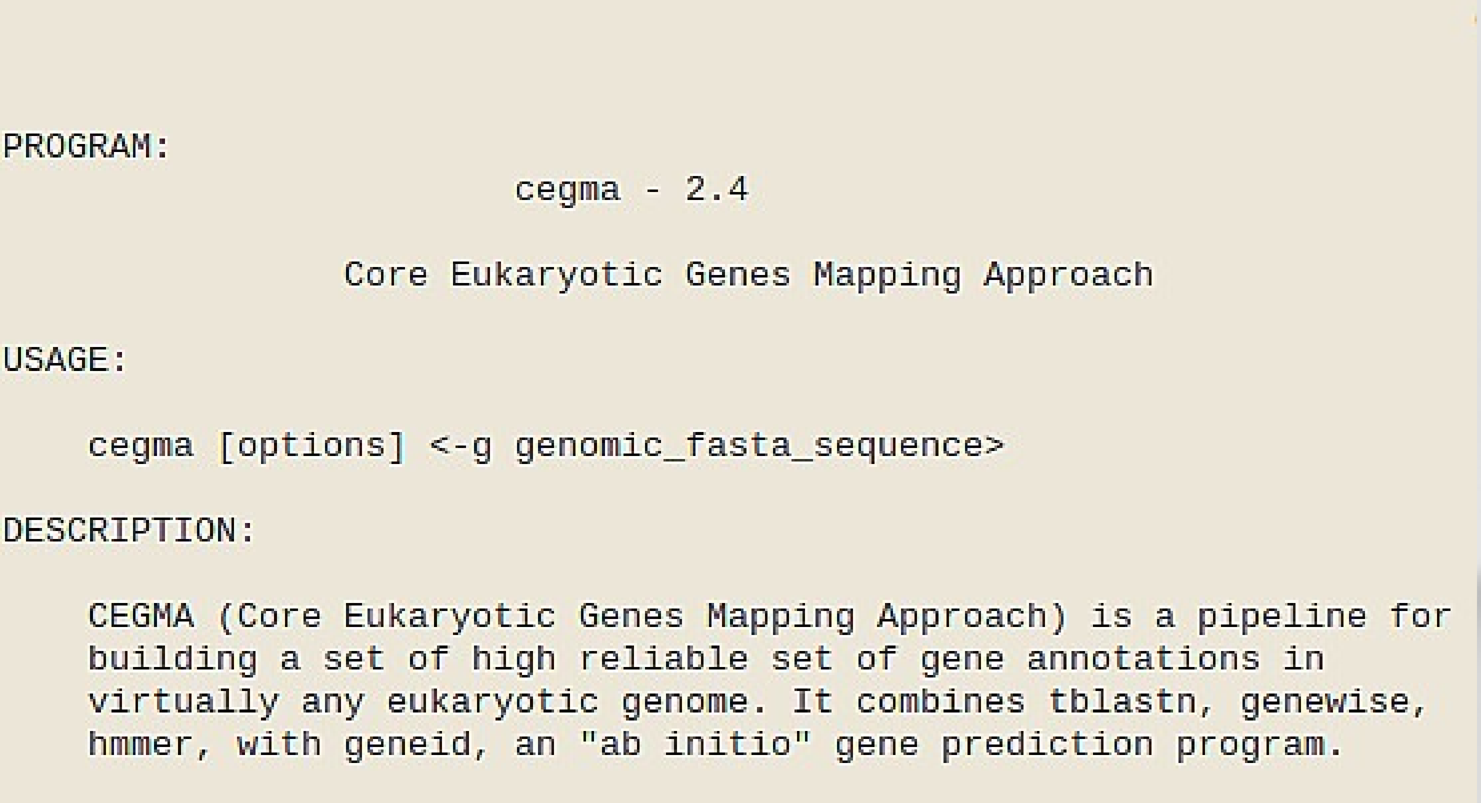
## CEGMA

Mapping on assembly

- protein profiles are built from set of core protein
- profiles are aligned on candidate regions from assembly
- the final structure of the gene is refined
- count of profiles which are found



# CEGMA: command line



```
PROGRAM:
                        cegma - 2.4

                Core Eukaryotic Genes Mapping Approach

USAGE:

    cegma [options] <-g genomic_fasta_sequence>

DESCRIPTION:

    CEGMA (Core Eukaryotic Genes Mapping Approach) is a pipeline for
    building a set of high reliable set of gene annotations in
    virtually any eukaryotic genome. It combines tblastn, genewise,
    hmmer, with geneid, an "ab initio" gene prediction program.
```

```
cegma -g assembly.fa
```

# CEGMA: output

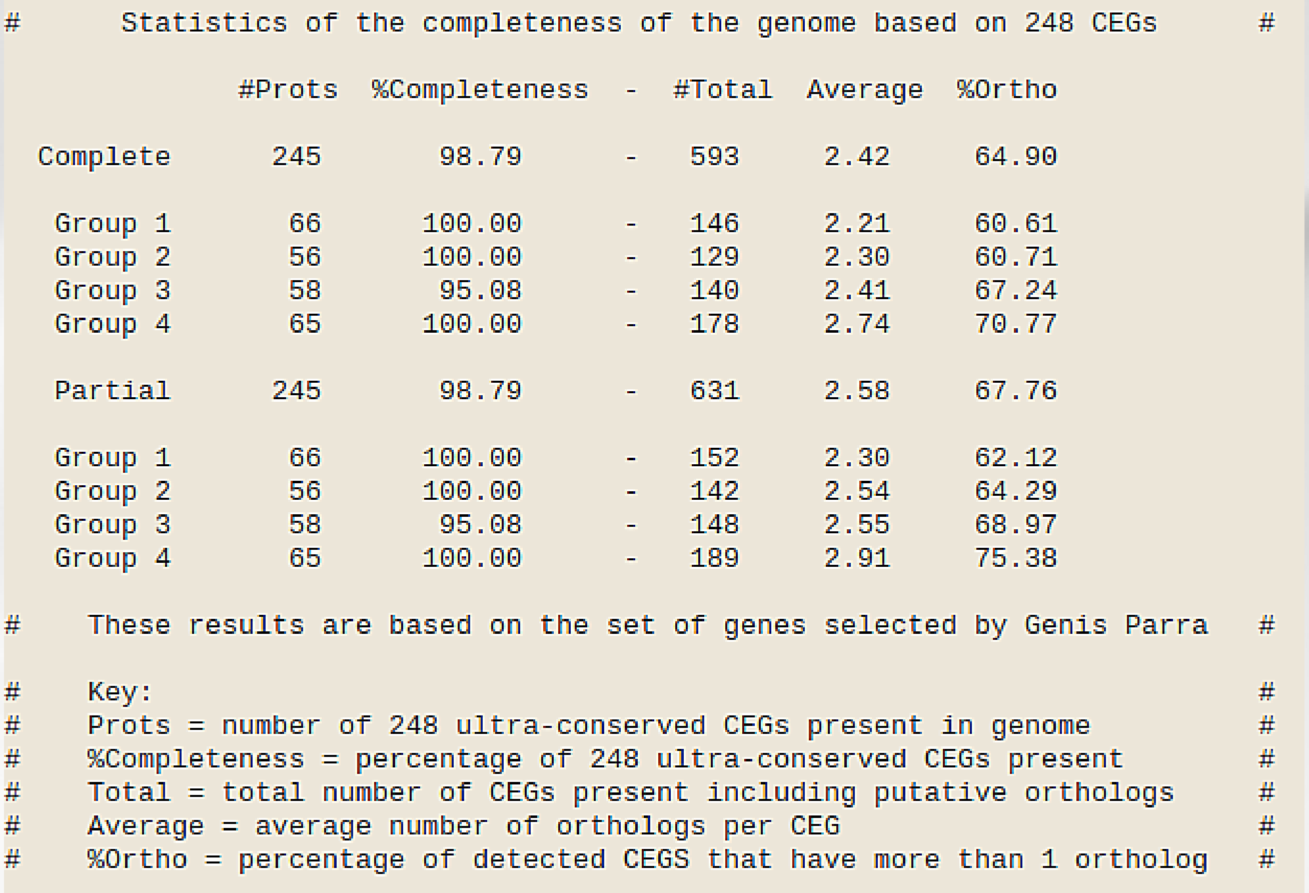
CEGMA produces 7 output files for each run.

- output.cegma.dna - contains DNA sequence of each CEGMA prediction with flanking DNA (defaults to ± 2000 bp)
- output.cegma.errors - contains any error messages
- output.cegma.fa - contains protein sequences of the predicted CEGs. One protein for each of the 248 core genes found
- output.cegma.gff - contains exon details of all of the CEGMA predicted genes
- output.cegma.id - contains the KOG IDs for the selected proteins
- output.cegma.local.gff - contains the GFF information of the CEGs using local coordiantes (relative to the dna file)
- output.completeness_report - contains a summary of which of the subset of the 248 CEGs are present

# CEGMA: output

Output example (output.completeness_report)

- Complete (70% of the protein length
- Partial (not matching “complete” criteria but exceed a pre-computed alignment score)



```
#      Statistics of the completeness of the genome based on 248 CEGs      #

             #Prots  %Completeness  -  #Total  Average  %Ortho

  Complete      245       98.79      -    593     2.42     64.90

   Group 1       66      100.00      -    146     2.21     60.61
   Group 2       56      100.00      -    129     2.30     60.71
   Group 3       58       95.08      -    140     2.41     67.24
   Group 4       65      100.00      -    178     2.74     70.77

   Partial      245       98.79      -    631     2.58     67.76

   Group 1       66      100.00      -    152     2.30     62.12
   Group 2       56      100.00      -    142     2.54     64.29
   Group 3       58       95.08      -    148     2.55     68.97
   Group 4       65      100.00      -    189     2.91     75.38

#    These results are based on the set of genes selected by Genis Parra   #

#    Key:                                                                  #
#    Prots = number of 248 ultra-conserved CEGs present in genome          #
#    %Completeness = percentage of 248 ultra-conserved CEGs present        #
#    Total = total number of CEGs present including putative orthologs     #
#    Average = average number of orthologs per CEG                         #
#    %Ortho = percentage of detected CEGS that have more than 1 ortholog   #
```

# ORF detection

EMBOSS getorf: find and extract open reading frames (ORFs)

ORF may be defined as a region between two STOP codons, or between a START and a STOP codon

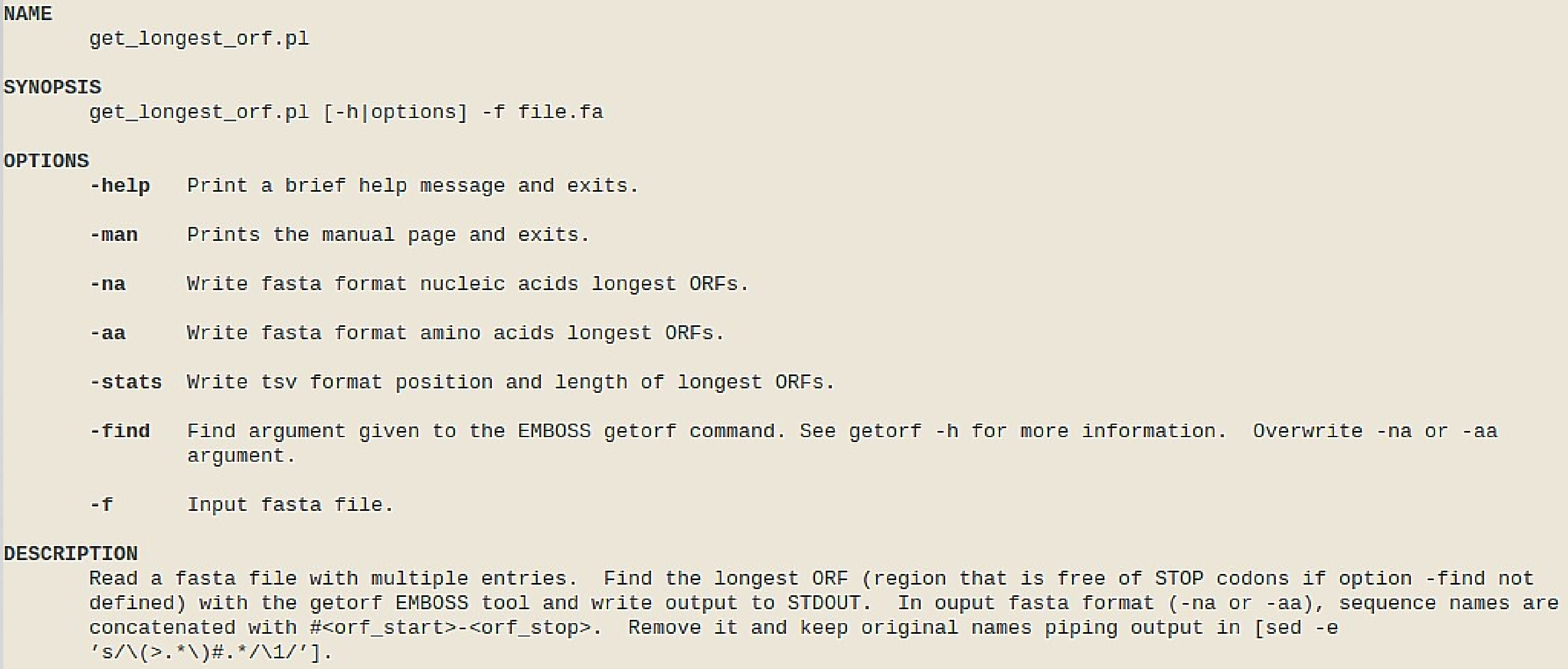
In house script to extract the longest ORF of each contig, having one input:

- contig fasta file

And one output:

- translated ORFs fasta file

# ORF detection



```
NAME
       get_longest_orf.pl

SYNOPSIS
       get_longest_orf.pl [-h|options] -f file.fa

OPTIONS
       -help   Print a brief help message and exits.

       -man    Prints the manual page and exits.

       -na     Write fasta format nucleic acids longest ORFs.

       -aa     Write fasta format amino acids longest ORFs.

       -stats  Write tsv format position and length of longest ORFs.

       -find   Find argument given to the EMBOSS getorf command. See getorf -h for more information.  Overwrite -na or -aa
               argument.

       -f      Input fasta file.

DESCRIPTION
       Read a fasta file with multiple entries.  Find the longest ORF (region that is free of STOP codons if option -find not
       defined) with the getorf EMBOSS tool and write output to STDOUT.  In ouput fasta format (-na or -aa), sequence names are
       concatenated with #<orf_start>-<orf_stop>.  Remove it and keep original names piping output in [sed -e
       ’s/\(>.*\)#.*/\1/’].
```

```
get_longest_orf.pl -f transcripts.fa -aa > transcripts.longest_orf.faa
```

# Contigs/ORFs annotation

Alignment against a reference:

- transcriptome
- proteome

Alignment using:

- blat (speed)
- exonerate (frame-shift)

May able to determine if our set of contigs:

- is exhaustive
- is mainly full length

# Exercise n°5

Plateforme Bioinformatique Midi-Pyrénées

# Example of an assembly pipeline

# PhyloFish Project

PHYLOgenomic analysis of gene duplications in teleost FISHes

- 20 fish species
- 13 tissues/species
- MGX platform in Montpellier
- HiSeq 2000 - PE - 100 pb
- Assembled using Velvet/Oases
- Build an assembly pipeline using Zebrafish data as test data and apply to all other species

# Assembly pipeline I

pre-oases

- illumina filter (discard low quality reads)
- extract the longest sub-sequence without N from each read

velvet-oases

- 9 independent assemblies (k-mers: 25, 31, 37, 43, 49, 55, 61, 65, 69)

merge

- select a unique transcript per Oases locus (bioinfo team of the Brown University)
- concatenate the 9 transcript files
- filter anti-sens chimeras (~~oases -merge~~)

# Assembly pipeline II

cd-hit-est

- remove duplicate transcripts build by close k-mers

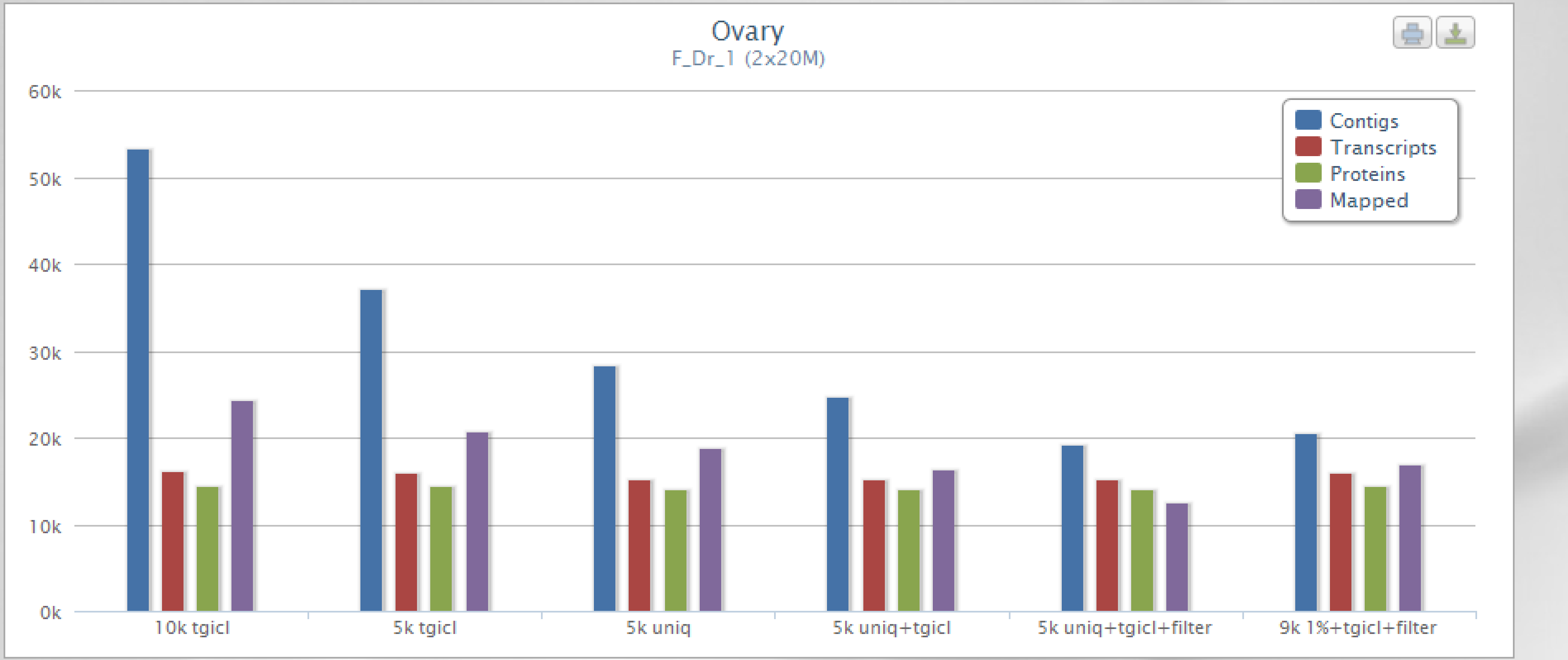
TGICL-CAP3

- assemble similar transcripts sharing large fragments (partial assemblies)

coverage and size filtering

- map reads back to transcripts
- find the longest ORF of each transcript
- coverage filter: at least 2/1M mapped reads
- size filter: ORF covers at least 200 pb

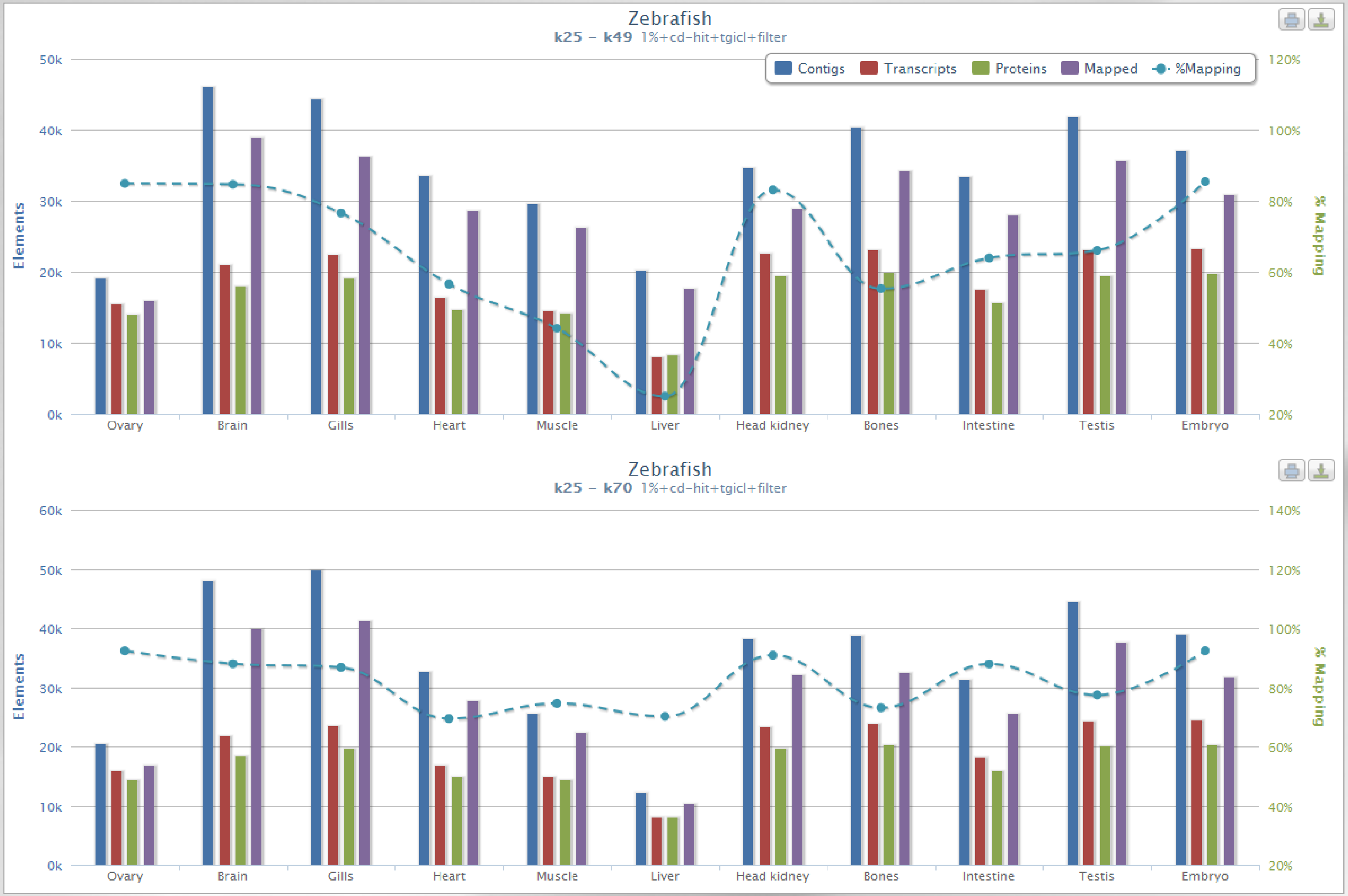
# Assembly pipeline tuning



10k ➪ 10 k-mers from 21 to 39 ; 5k ➪ 5 k-mers from 25 to 49 ; 9k ➪ 9 k-mers from 25 to 69

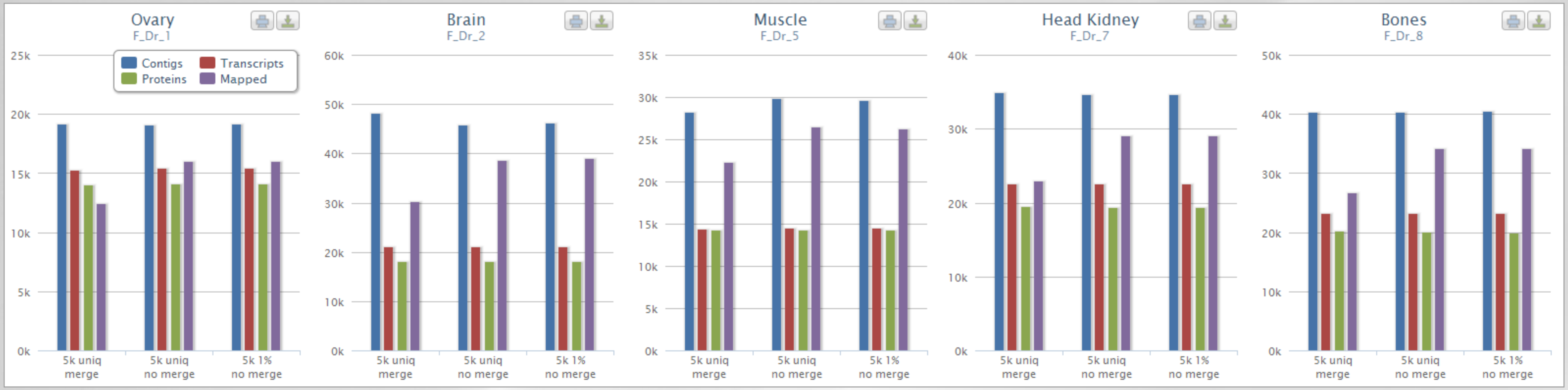
The number of transcripts falls whereas the number of rebuilt transcripts or proteins is quite stable

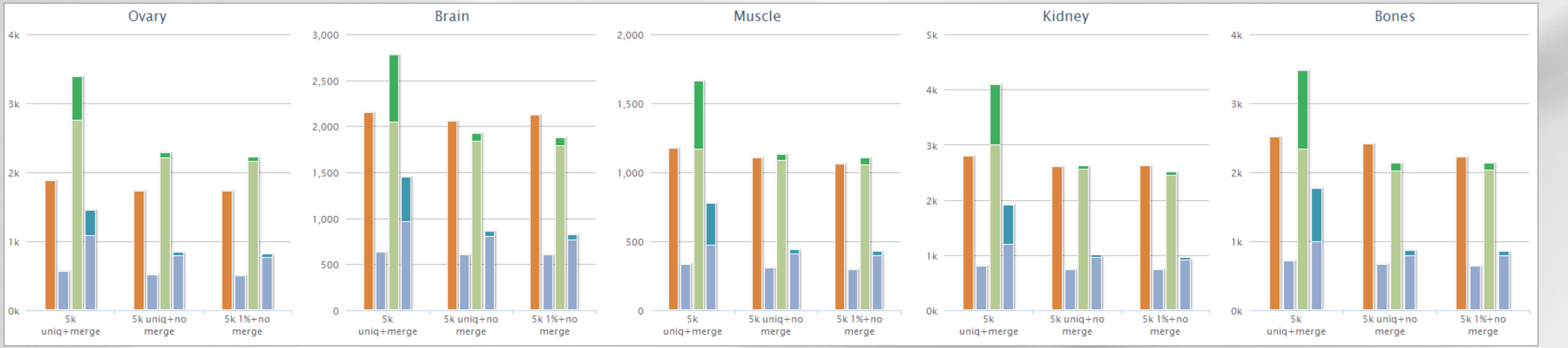
# Assembly pipeline tuning



Enlarge from 5 k-mers to 9 k-mers increases slightly the total of produced transcripts but increases significantly the mapping rate

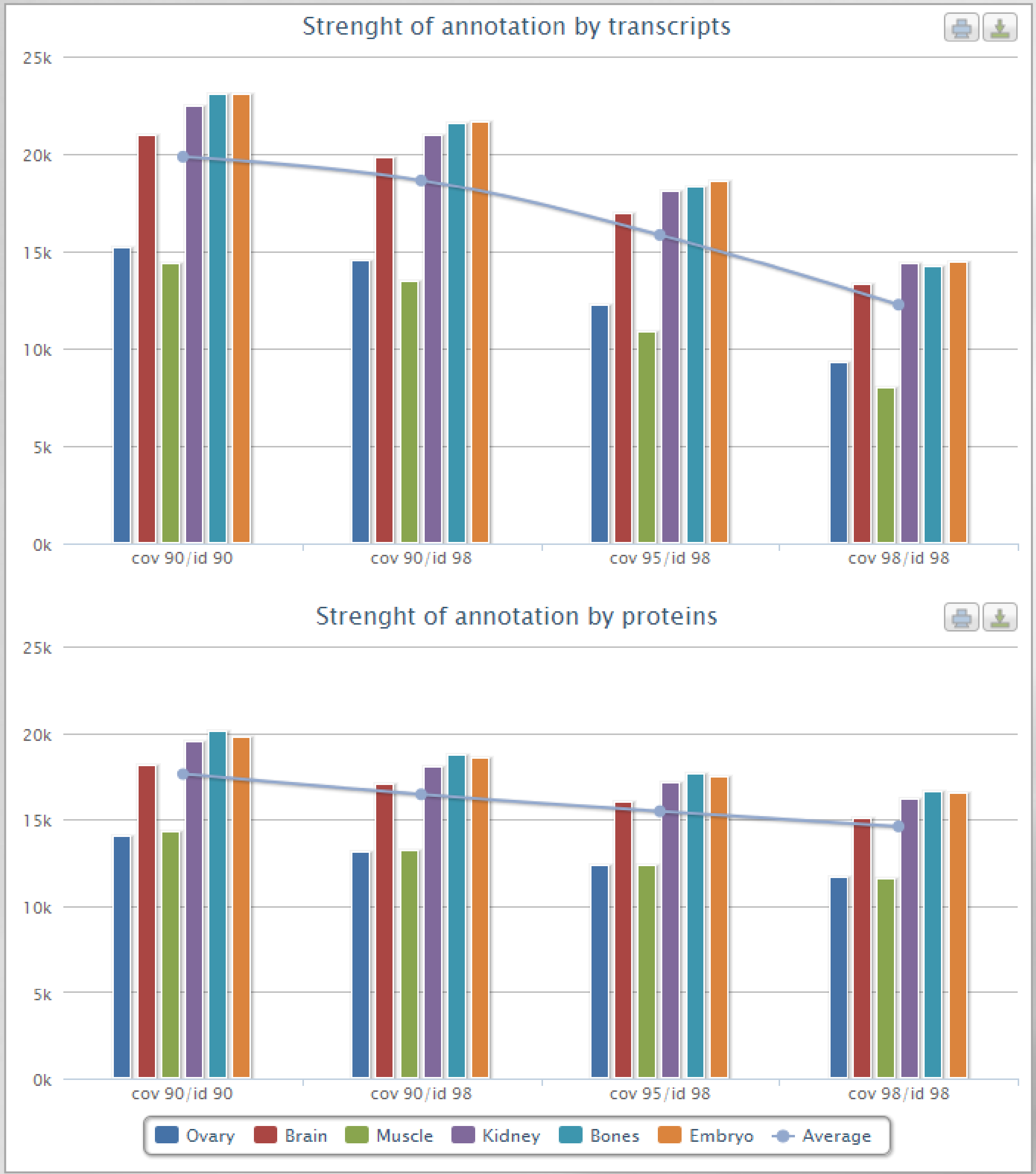
# Assembly pipeline tuning





Remove the **oases -merge** and keep 1% transcripts/locus has a minor effect on the total of produced transcripts, rebuilt transc., rebuilt prot. but allow to sensibly reduce the total of anti-sens chimeras

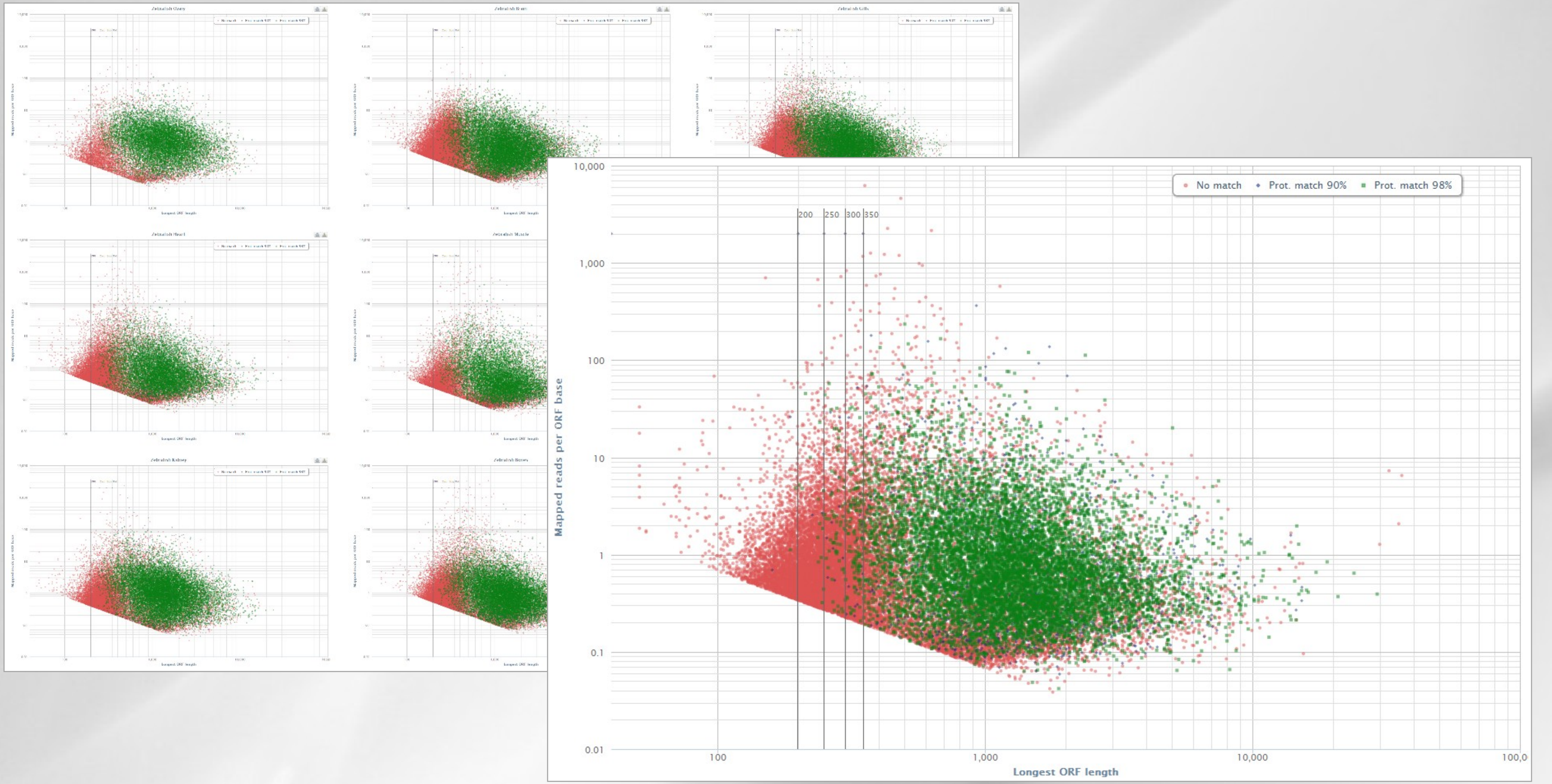
# Assembly pipeline tuning



Increase the identity threshold has a minor effect. Not true for the coverage threshold.

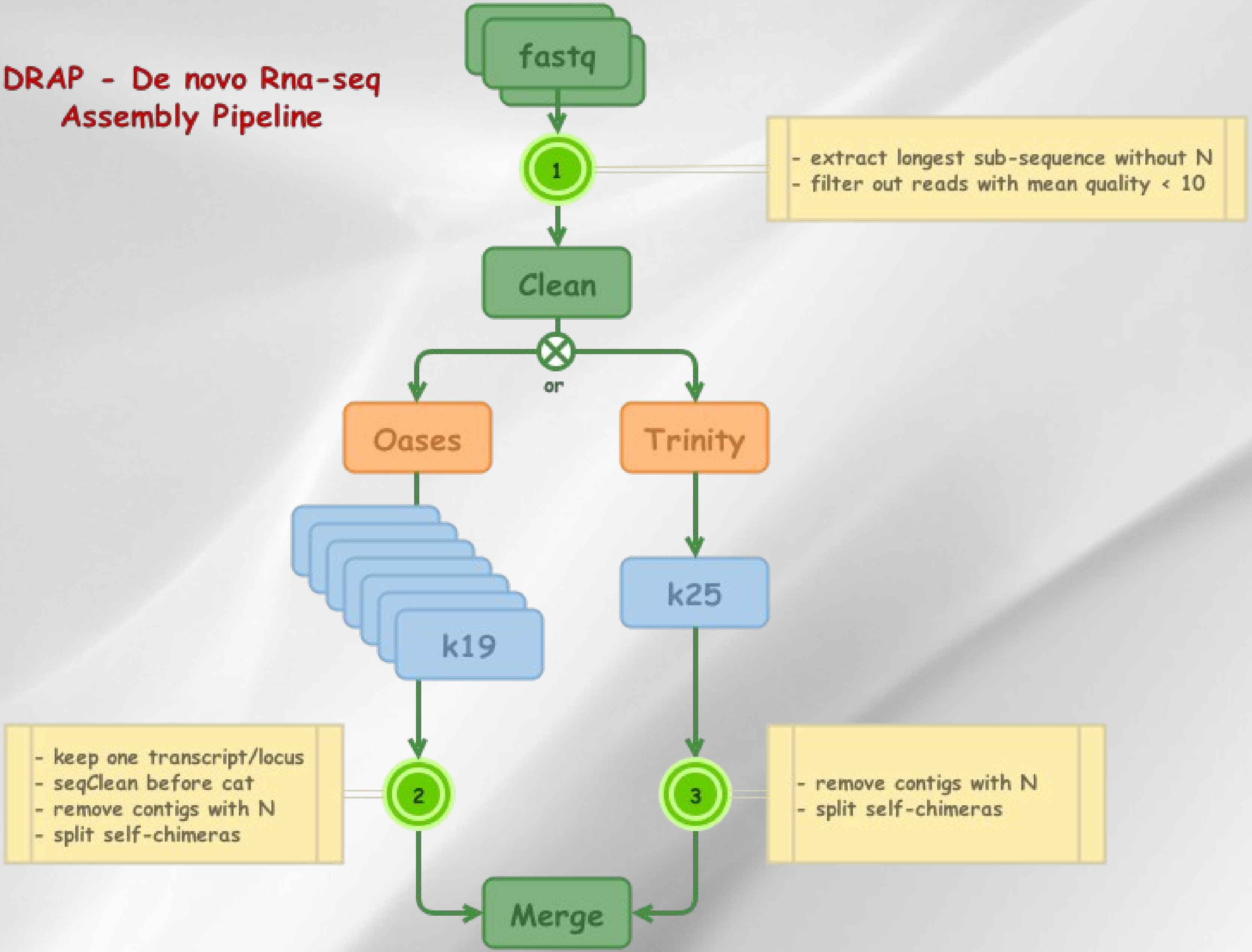
This could means that rebuilt transcripts are pretty well rebuilt but might be incomplete.

# Assembly pipeline tuning



Coverage and ORF size filters (+contamination removal) were determined by analysis of plots of transcript features

# Our assembly pipeline



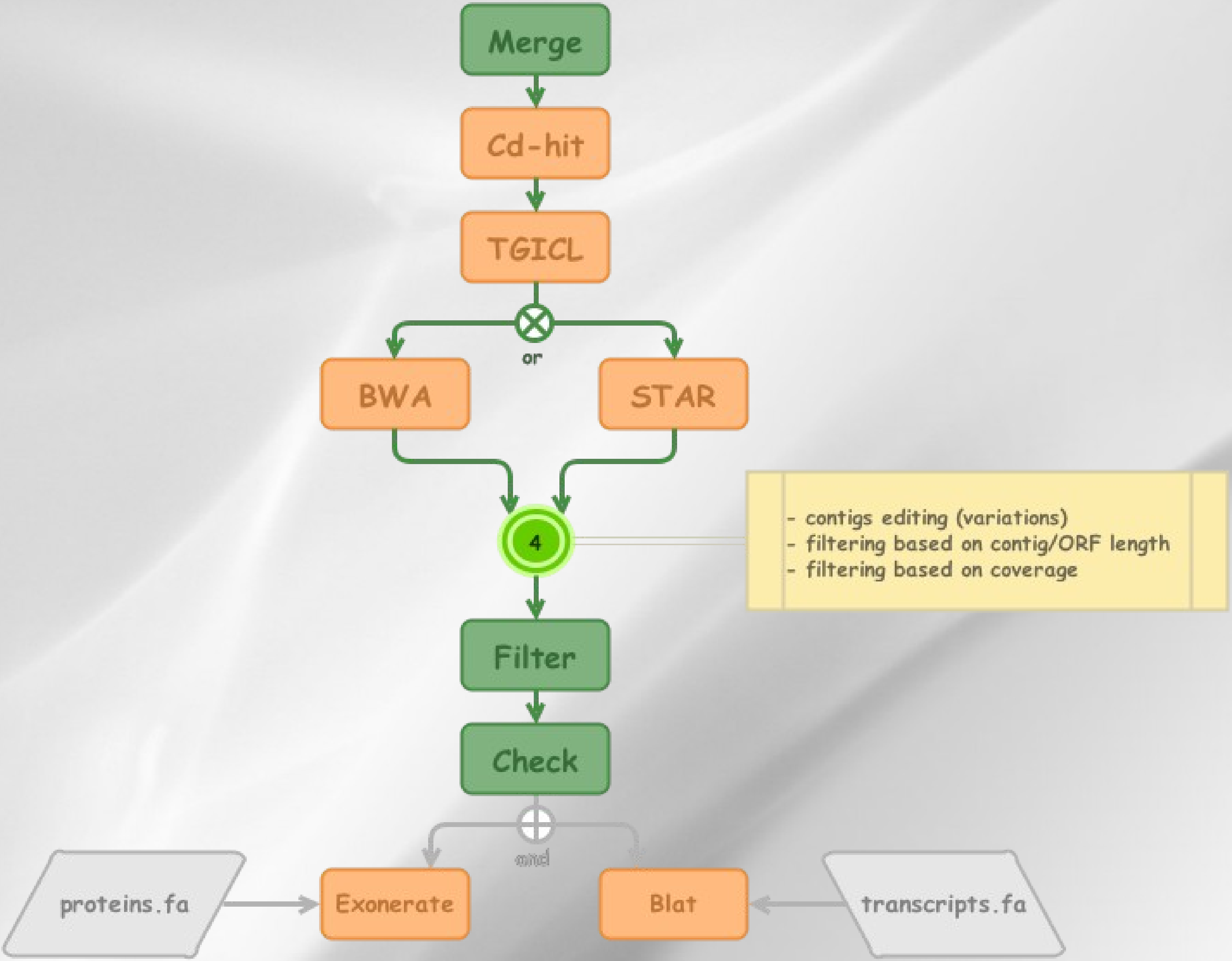
DRAP - De novo Rna-seq Assembly Pipeline

- extract longest sub-sequence without N
- filter out reads with mean quality < 10

- keep one transcript/locus
- seqClean before cat
- remove contigs with N
- split self-chimeras

- remove contigs with N
- split self-chimeras

# Our assembly pipeline



- contigs editing (variations)
- filtering based on contig/ORF length
- filtering based on coverage

# Exercise n°6

# Do not forget that we have several samples

# Let meta-assemble!

# Meta-assembly on the experiment level

Produce a unique transcriptome from several samples assembled separately

Samples could be:

- from different organisms
- from different tissues
- from different experimental conditions

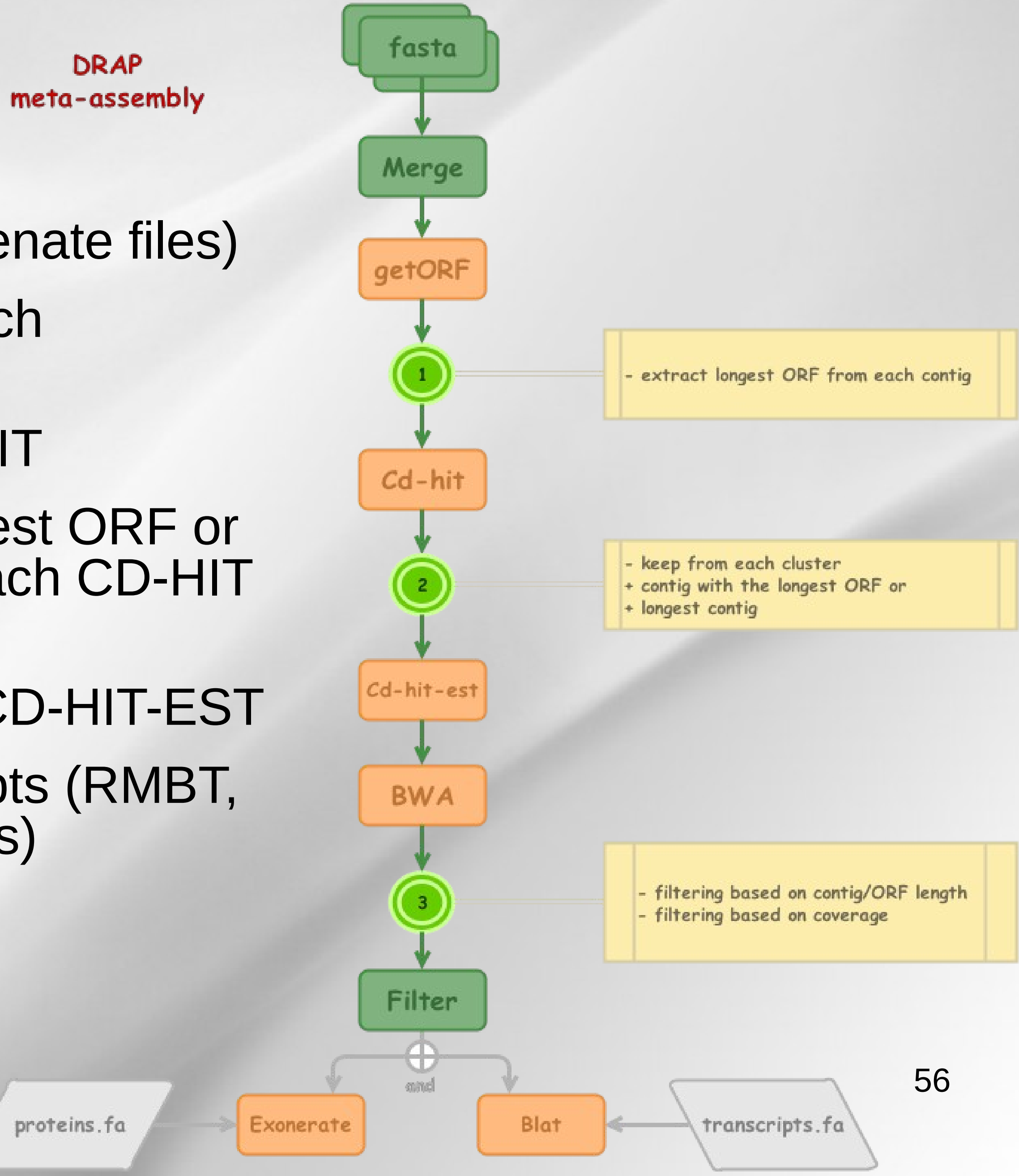
⤷ clusterize transcripts from same genes rebuilt in each sample

⤷ keep only one representative transcript per cluster

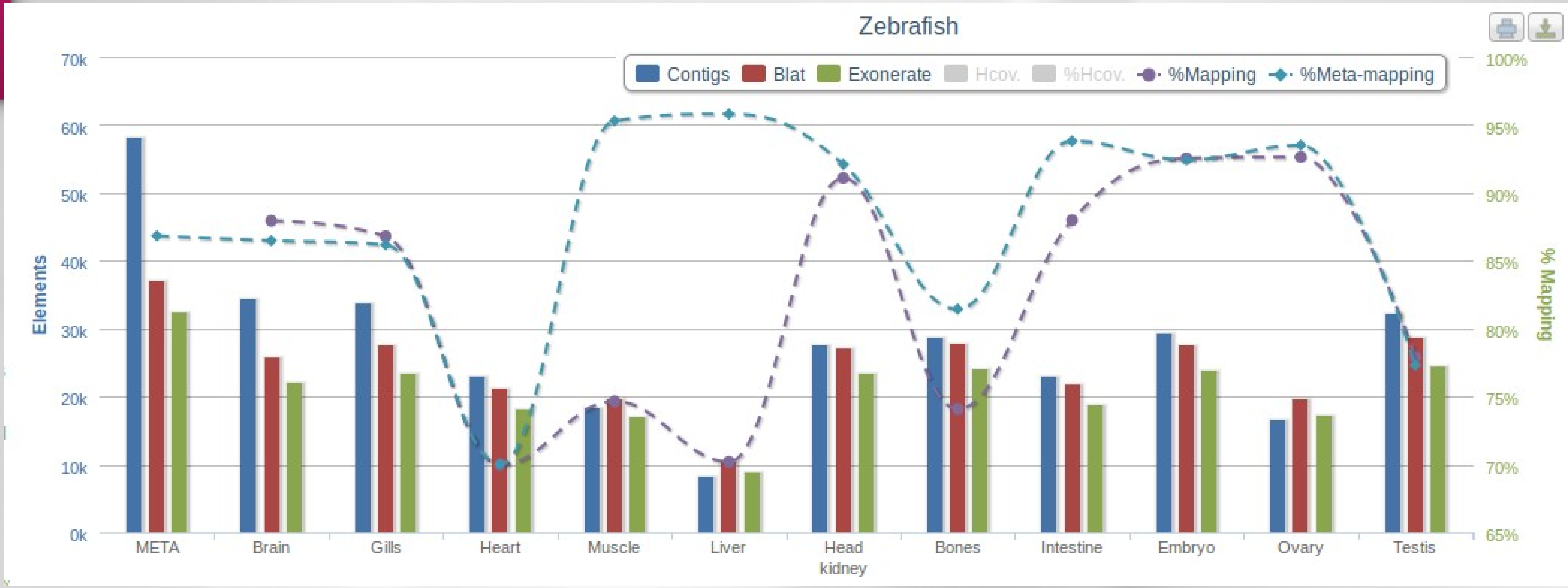
# Meta-assembly procedure

Six steps:

- merge assemblies (concatenate files)
- get the longest ORF for each transcript
- clusterize ORFs with CD-HIT
- get transcript with the longest ORF or the longest transcript for each CD-HIT cluster
- clusterize transcripts with CD-HIT-EST
- filter low coverage transcripts (RMBT, at least 1/1M mapped reads)



# Meta-assembly benefits



For reads from all tissues, the mapping rate on meta-assembly is at least equivalent to those on tissue specific assembly. Sometimes it is much higher.

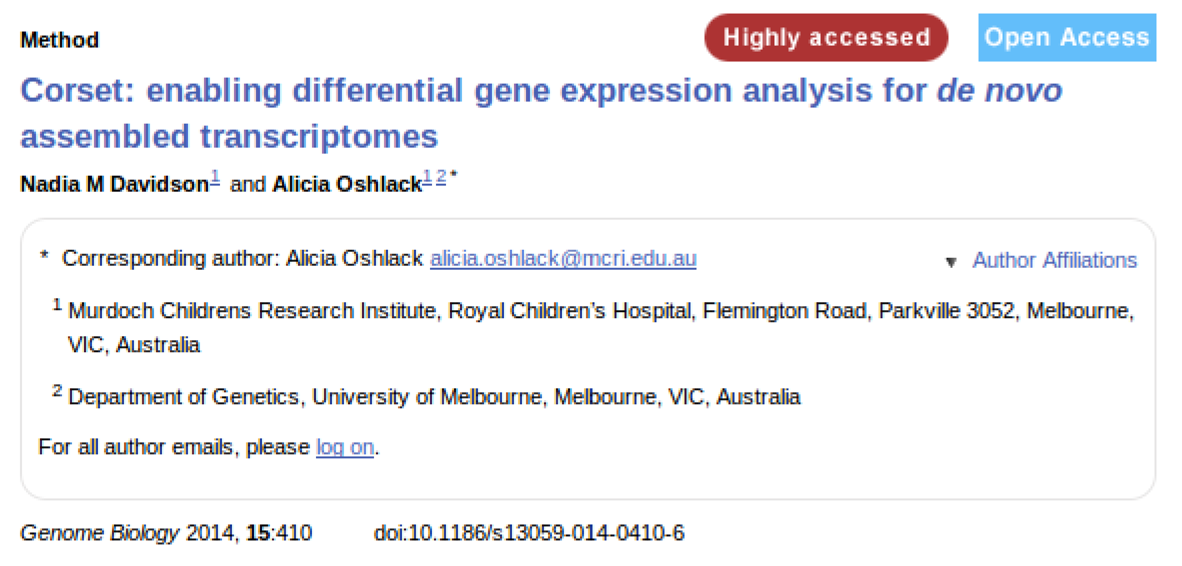
# Exercise n°7

Plateforme Bioinformatique Midi-Pyrénées

# Once the assembly is finished

# Contigs to Unigenes

Corset: hierarchically clustering of the transcripts based on the proportion of shared reads. Need to produce bam files with all locations for each reads (bowtie2 --all or STAR).



Method

**Corset: enabling differential gene expression analysis for *de novo* assembled transcriptomes**

**Nadia M Davidson**[1] and **Alicia Oshlack**[1 2 *]

[*] Corresponding author: Alicia Oshlack alicia.oshlack@mcri.edu.au

▾ Author Affiliations

[1] Murdoch Childrens Research Institute, Royal Children's Hospital, Flemington Road, Parkville 3052, Melbourne, VIC, Australia

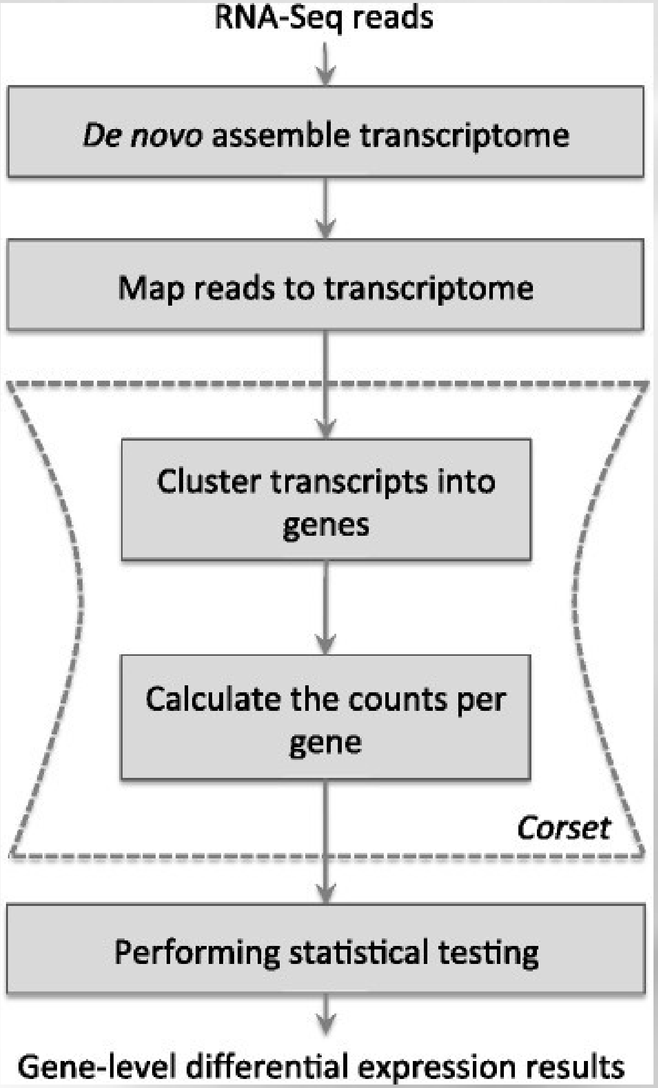
[2] Department of Genetics, University of Melbourne, Melbourne, VIC, Australia

For all author emails, please log on.

*Genome Biology* 2014, **15**:410 doi:10.1186/s13059-014-0410-6

Corset: enabling differential gene expression analysis for de novo assembled transcriptomes.
Davidson NM, Oshlack A.
Genome Biol. 2014 Jul 26;15(7):410

# Corset


RNA-Seq reads
*De novo* assemble transcriptome
Map reads to transcriptome
Cluster transcripts into genes
Calculate the counts per gene
*Corset*
Performing statistical testing
Gene-level differential expression results

```
Usage: corset [options] -n <names> <input bam files>
```

# Further investigations

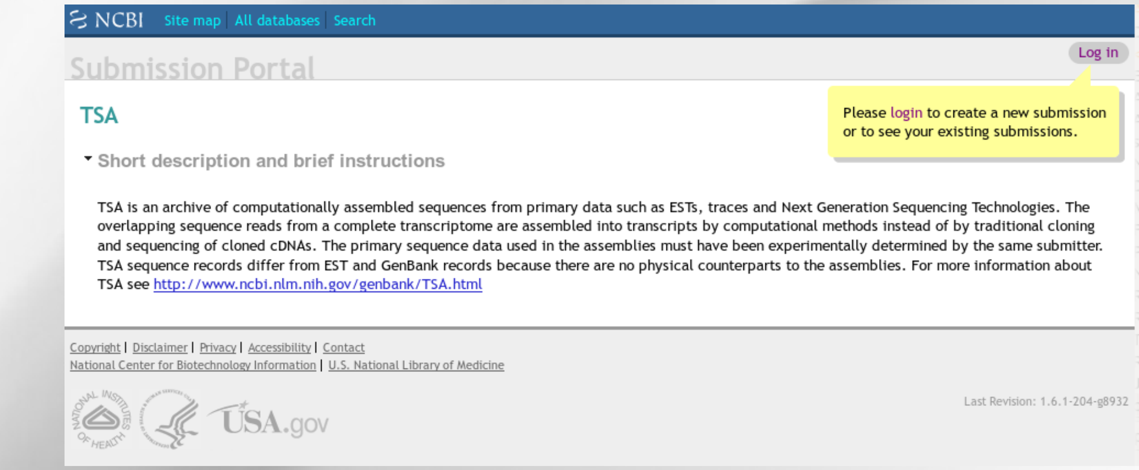
- Annotation
  - Functional
  - Structural
- Variation search
- Publication

# Publishing your transcriptome assembly

## Strategies :

- Put the contig file in the paper supplementary data.
- Provide an access to download the contig file.
- Provide a web-site with the contigs, annotation, etc...
- Publish your contigs in the corresponding public archive TSA.

https://submit.ncbi.nlm.nih.gov/subs/tsa/



Submission Portal

TSA

Please login to create a new submission or to see your existing submissions.

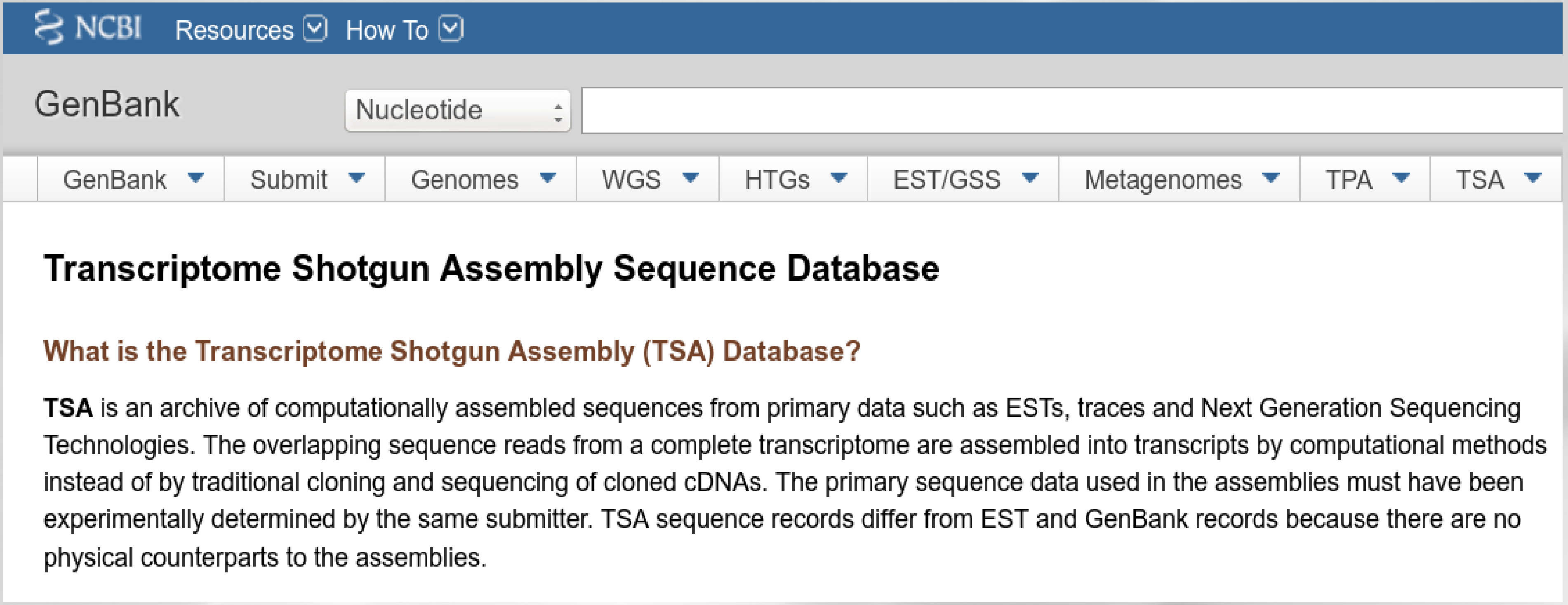
Short description and brief instructions

TSA is an archive of computationally assembled sequences from primary data such as ESTs, traces and Next Generation Sequencing Technologies. The overlapping sequence reads from a complete transcriptome are assembled into transcripts by computational methods instead of by traditional cloning and sequencing of cloned cDNAs. The primary sequence data used in the assemblies must have been experimentally determined by the same submitter. TSA sequence records differ from EST and GenBank records because there are no physical counterparts to the assemblies. For more information about TSA see http://www.ncbi.nlm.nih.gov/genbank/TSA.html

Copyright | Disclaimer | Privacy | Accessibility | Contact
National Center for Biotechnology Information | U.S. National Library of Medicine

Last Revision: 1.6.1-204-g8932

# Transcriptome Shotgun Assembly Sequence Database



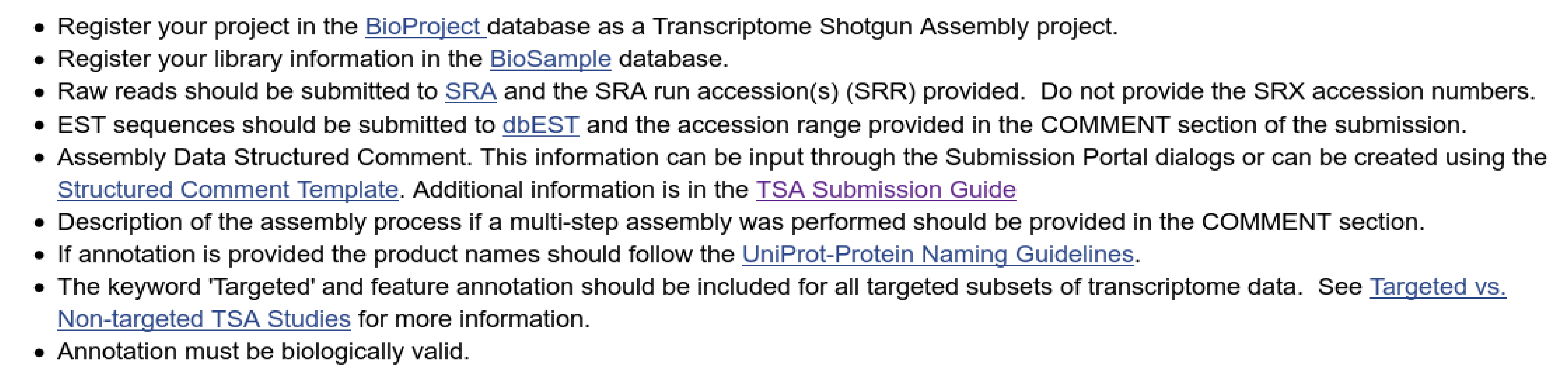
## Transcriptome Shotgun Assembly Sequence Database

### What is the Transcriptome Shotgun Assembly (TSA) Database?

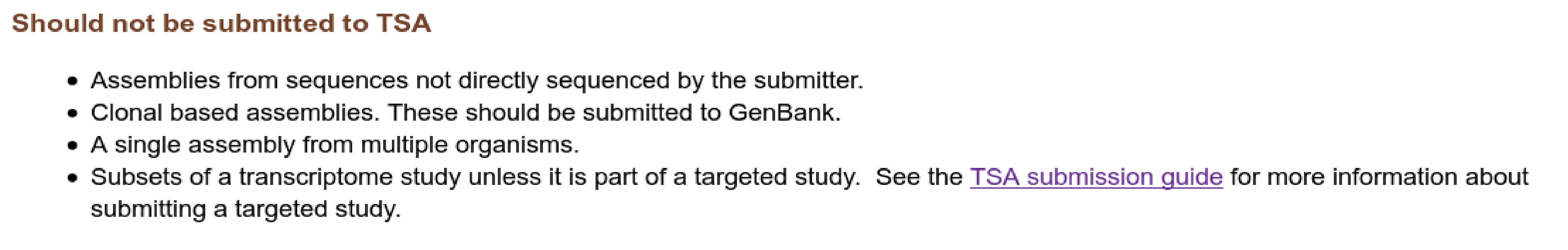
**TSA** is an archive of computationally assembled sequences from primary data such as ESTs, traces and Next Generation Sequencing Technologies. The overlapping sequence reads from a complete transcriptome are assembled into transcripts by computational methods instead of by traditional cloning and sequencing of cloned cDNAs. The primary sequence data used in the assemblies must have been experimentally determined by the same submitter. TSA sequence records differ from EST and GenBank records because there are no physical counterparts to the assemblies.

http://www.ncbi.nlm.nih.gov/genbank/tsa/

# Requirements



- Register your project in the BioProject database as a Transcriptome Shotgun Assembly project.
- Register your library information in the BioSample database.
- Raw reads should be submitted to SRA and the SRA run accession(s) (SRR) provided. Do not provide the SRX accession numbers.
- EST sequences should be submitted to dbEST and the accession range provided in the COMMENT section of the submission.
- Assembly Data Structured Comment. This information can be input through the Submission Portal dialogs or can be created using the Structured Comment Template. Additional information is in the TSA Submission Guide
- Description of the assembly process if a multi-step assembly was performed should be provided in the COMMENT section.
- If annotation is provided the product names should follow the UniProt-Protein Naming Guidelines.
- The keyword 'Targeted' and feature annotation should be included for all targeted subsets of transcriptome data. See Targeted vs. Non-targeted TSA Studies for more information.
- Annotation must be biologically valid.



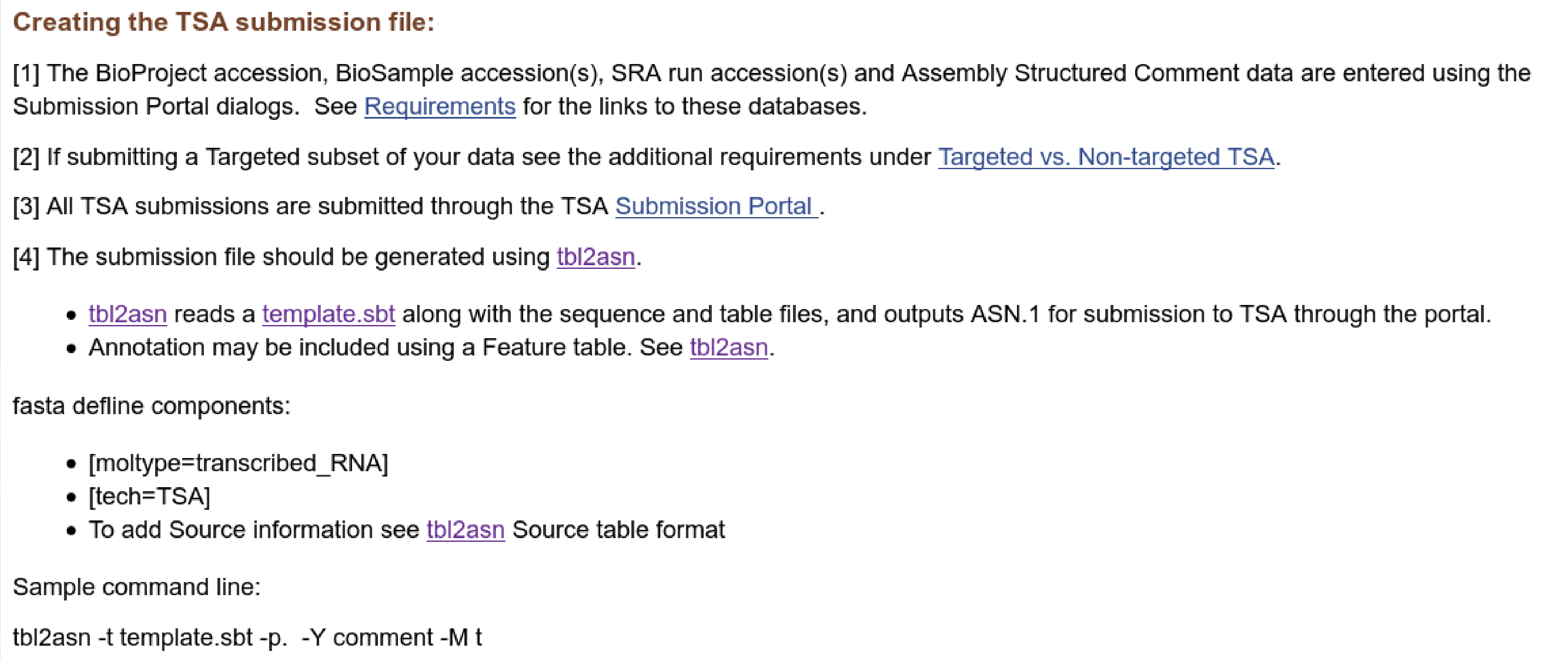
**Should not be submitted to TSA**

- Assemblies from sequences not directly sequenced by the submitter.
- Clonal based assemblies. These should be submitted to GenBank.
- A single assembly from multiple organisms.
- Subsets of a transcriptome study unless it is part of a targeted study. See the TSA submission guide for more information about submitting a targeted study.

# Submission standards

- Submitted sequences must be assembled from data experimentally determined by the submitter.
- **Screened for vector contamination and any vector/linker sequence removed. This includes the removal of NextGen sequencing primers.**
- Sequences should be **greater than 200 bp** in length.
- **Ambiguous bases should not be more than total 10% length or more than 14n's in a row.**
- Sequence gaps of known length may be present and annotated with the assembly_gap feature if there is sufficient evidence for the linkage between the sequences.  See the TSA Submission Guide for more information about adding assembly_gap features.
- Gaps cannot be of unknown length.
- If the submission is a single-step, unannotated assembly and the output is a BAM file(s) these should be submitted as a TSA project to SRA.

# Submission file



### Creating the TSA submission file:

[1] The BioProject accession, BioSample accession(s), SRA run accession(s) and Assembly Structured Comment data are entered using the Submission Portal dialogs. See Requirements for the links to these databases.

[2] If submitting a Targeted subset of your data see the additional requirements under Targeted vs. Non-targeted TSA.

[3] All TSA submissions are submitted through the TSA Submission Portal .

[4] The submission file should be generated using tbl2asn.

- tbl2asn reads a template.sbt along with the sequence and table files, and outputs ASN.1 for submission to TSA through the portal.
- Annotation may be included using a Feature table. See tbl2asn.

fasta defline components:

- [moltype=transcribed_RNA]
- [tech=TSA]
- To add Source information see tbl2asn Source table format

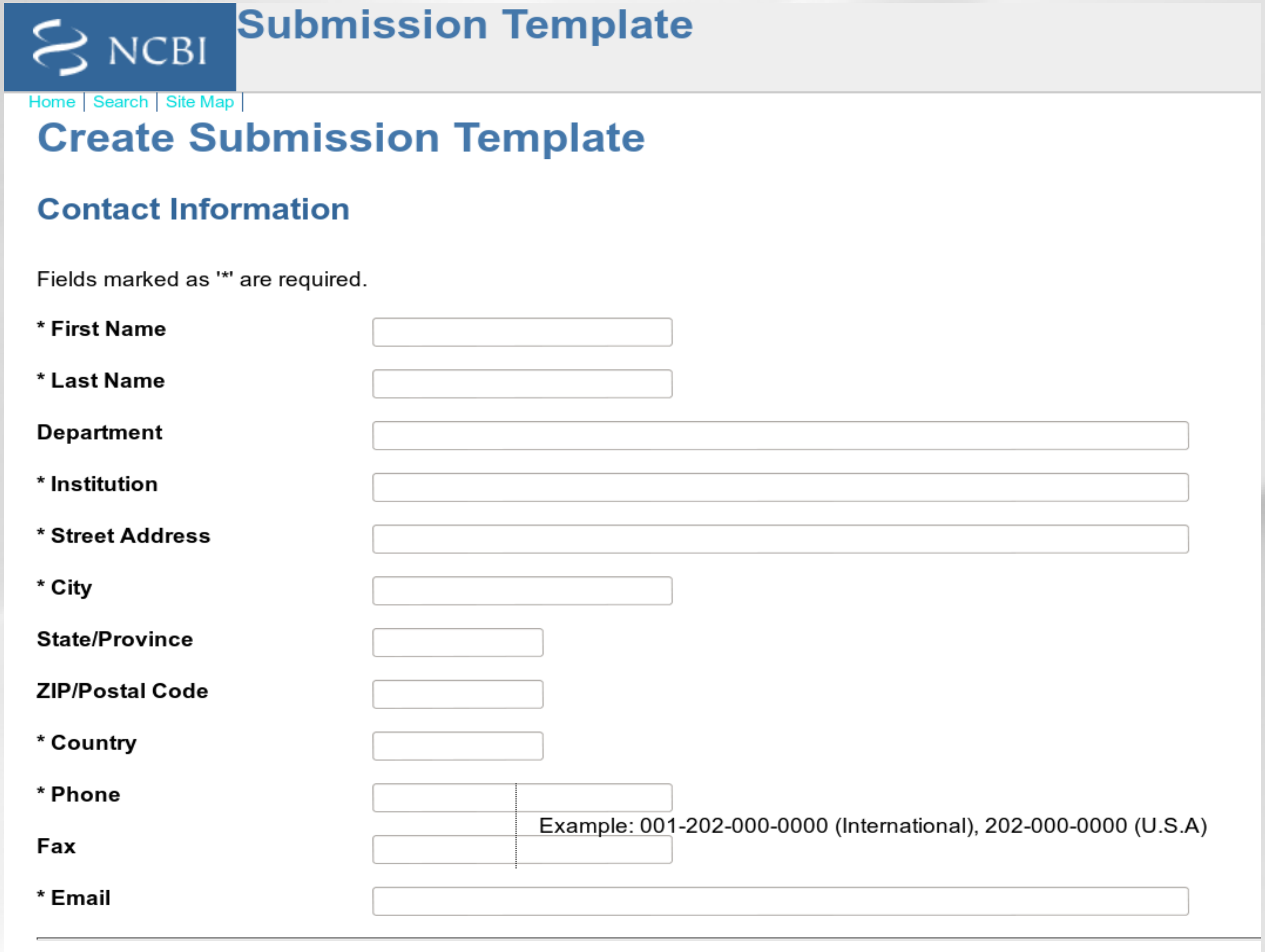
Sample command line:

```
tbl2asn -t template.sbt -p.  -Y comment -M t
```

http://www.ncbi.nlm.nih.gov/genbank/tsaguide

# Producing your template file

http://www.ncbi.nlm.nih.gov/WebSub/template.cgi



## Create Submission Template

### Contact Information

Fields marked as '*' are required.

* First Name
* Last Name
Department
* Institution
* Street Address
* City
State/Province
ZIP/Postal Code
* Country
* Phone
Fax
* Email

Example: 001-202-000-0000 (International), 202-000-0000 (U.S.A)

# Best practices

Run your contigs through the TSA publication process before using them in the analysis step in order to filter out the ones you will not be able to publish.

For multi-species experiments (host / pathogene,...) separate the contigs after annotation and publish the different contig sets individually.

# Questions?

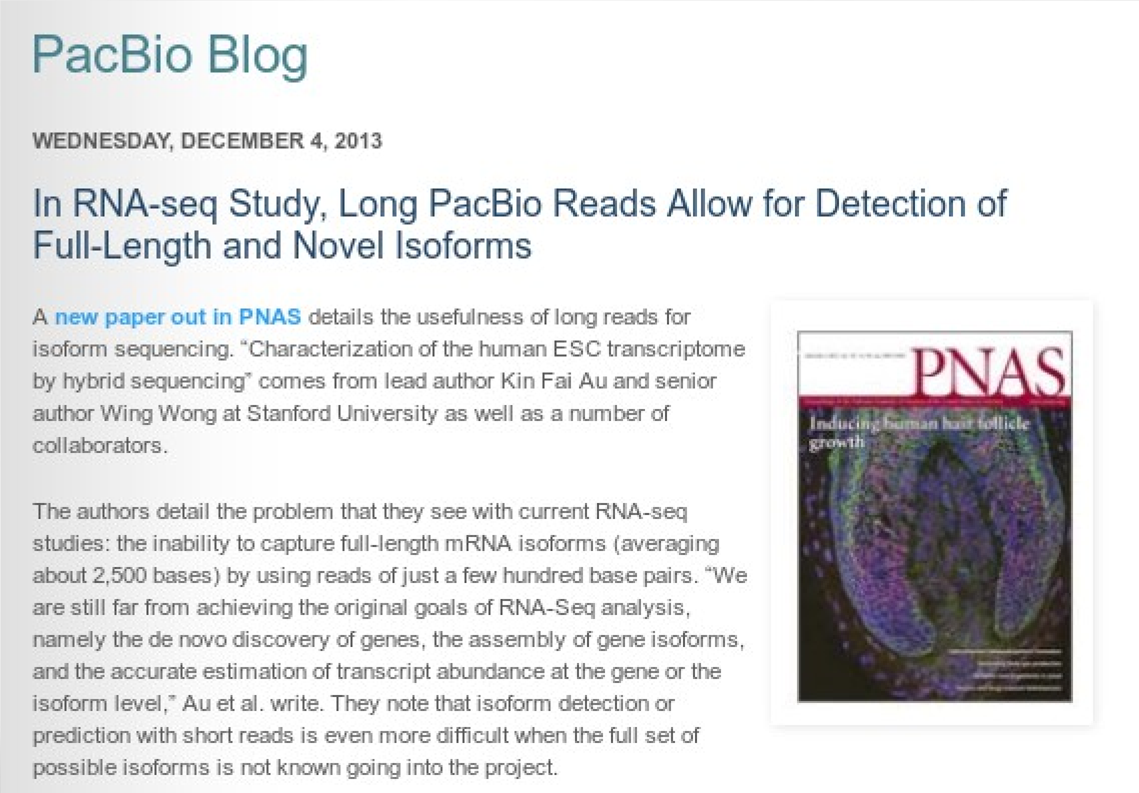
# Conclusions

- For a good assembly, better :
  - have many biological replicates (even low coverage),
  - have several tissues and conditions to have a broader view of the transcriptome,
  - clean input data
  - use different contig cleaning steps corresponding to error patterns (refining)
  - check your re-mapping rate
  - get rid of lowly covered contigs
  - check your contigs versus a closely related protein set

# Perspectives

Third generation sequencers :

- New PacBio chemistry P6-C4
  - Average read length 14kb
  - 1 Gb per cell



PacBio Blog

WEDNESDAY, DECEMBER 4, 2013

## In RNA-seq Study, Long PacBio Reads Allow for Detection of Full-Length and Novel Isoforms

A new paper out in PNAS details the usefulness of long reads for isoform sequencing. "Characterization of the human ESC transcriptome by hybrid sequencing" comes from lead author Kin Fai Au and senior author Wing Wong at Stanford University as well as a number of collaborators.

The authors detail the problem that they see with current RNA-seq studies: the inability to capture full-length mRNA isoforms (averaging about 2,500 bases) by using reads of just a few hundred base pairs. "We are still far from achieving the original goals of RNA-Seq analysis, namely the de novo discovery of genes, the assembly of gene isoforms, and the accurate estimation of transcript abundance at the gene or the isoform level," Au et al. write. They note that isoform detection or prediction with short reads is even more difficult when the full set of possible isoforms is not known going into the project.