

How to do a genome annotation?



Introduction: Formats

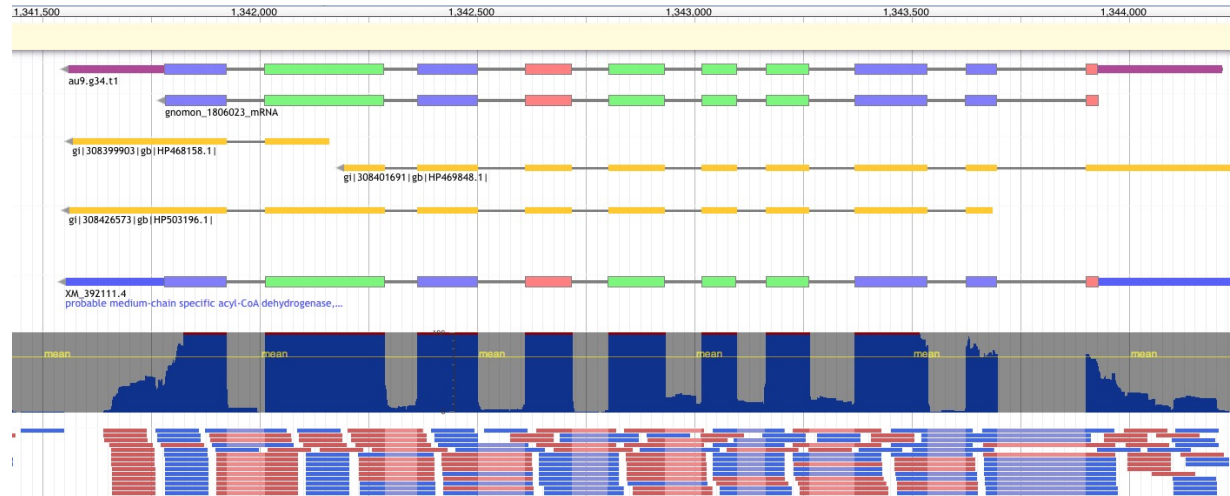
From a genome...

FASTA

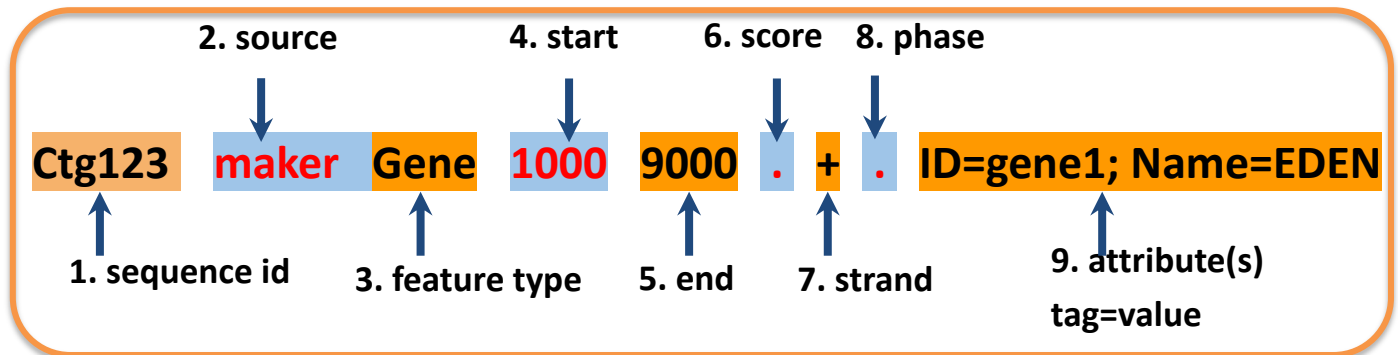
...to an annotated gene

GTF/GFF

```
>scaffold_26
AGTCACACACCCCTTCAGCTTACACCCCTGACTGCAGCCCTTACTCAAAACA
TTCCAGCCAGGAAGATGCTCCGACACAGCTTCTGGATGCCGCTCCTCGAC
GTCGAACGGCCCGCGCCGGGAAAATCGGCAGCGTCGGTGACCGCGGAGAT
CCGAAGCCGCTCGGGGACCTGCGAGACAACGGGAGGCGGTCAACGAGAC
GCCGAGGGCTGGGAGTTATTCACACACGGGCCCGTAAGTTTTCTACCCA
AAAACCCATAGAAAAGAGATGAACCACTAAGTTTGATAACTCTTCTACTT
AACCCTGACCCCTACGTGCCGGGCGAGGCGAGCTCTGACCCTAAGCGGCAC
ACGAACAAGGTGGTGCGCCCAATATAAACAAAGATGATGCAAGGGCTTGA
AATAAATCTCCGGAAGATTAATTCTCGAGCCCGACACGCTTTGAGGCAGC
GGAACCTACAGAACCCCGCAGTCACGTGAGAAGAGTCTAATACTCTCCA
AAGAGAAGTCCAAGGGAATGGAACGTGAAAAGAAGGTGCTTATCAAAAGC
GAGAAGGAAGATGGATGAGAACATCTTGTACTTCTTCTGGTCTCAAAA
AGCAAAAATGTAAGATGCCAGACTAAGCCCGACTCTGAGAAGTACGCGA
GCAGAGACCCCGCTGCCGATGTGGCCGAGAAGCATGCCGATAAAGCACC
GAGACATAACAAGCCCTGTGACACACAAGACGATGGACACAACTACAT
AACACAGACACAACTAAATGACACAGAGAGAAGTTGAAACTTCTGGGGA
AGTAAACATTTCTGAAACATCTACCAACAATCCGTCATATATATTTCCA
TTCCAGGGGACTCTGGTTTGATATATGCGTGTAAACAGTAATCCCGCT
GTAGCAATCACCACATATGCATAATTCATTAATCTTTGGAGTTGCTGAGT
ATCATCTTATCAGTCTTATTTTTTCTTGGCTCTGTTTCGGGCTTTTT
TTTTTCTTCTGATAAGATTTTCCAGGAATGTGAAGACCCCTGCATCCT
TCCAAACTGACCAACCAACTACAGACATTCATAGCATTACATTACAC
AACCTAGGCAAGTTTTTCTAACATTAAAGAACATGAAAAGCCAACAT
CACAATATATTCATAACAATTATGGAACATGCGAAAAGCCAATACACAG
TACATTATAACAATACCTCCCTTTTCTTTCTTTAGAGATCATATGGCT
TGACCGCCGCTCTCTGCCCGCCACCGCTGAGTACTGCGGTGCCGGAGTC
ACGGAGCCAGTCCCCCGCGGCCACCGCTCTCTGCCCGCCGCGCACGGA
GATCGGCTGCGCACTCCCGAGCTCGGCCGTGCCATCGCGCCCGCCGCG
GGGTCCCCCGCNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```



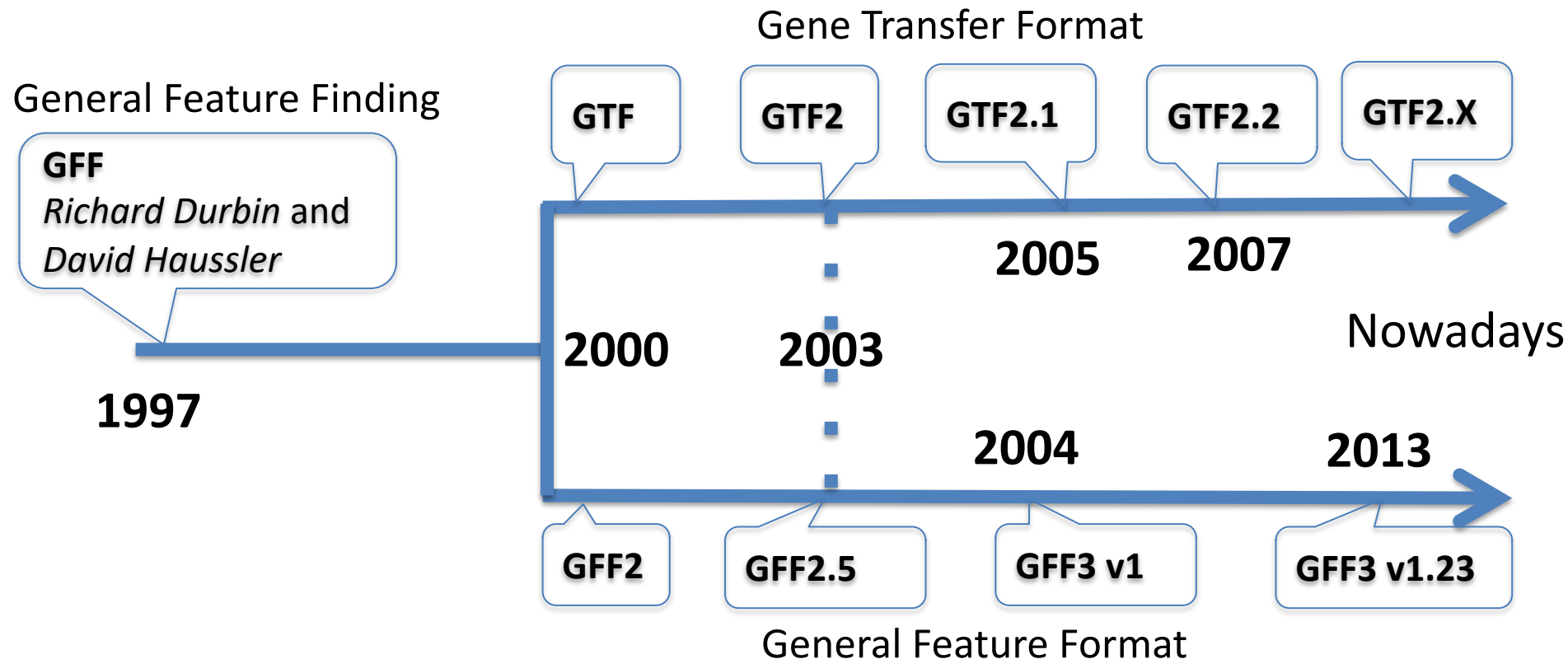
- 9 columns
- 1 feature = 1 line





□ GFF / GTF formats

<https://github.com/NBISweden/GAAS/blob/master/annotation/knowledge/gxf.md>



Introduction: Formats: GTF2.X

- 9 columns
- 1 feature = 1 line

Header

#!genome-build GRCz11 #!genome-date 2017-05								
Ctg123	.	Gene	1000	9000	.	+	.	gene_id gene1; name EDEN;
ctg123	.	Transcript	1050	9000	.	+	.	gene_id gene1; transcript_id=t1; name EDEN;
ctg123	.	Transcript	1050	9000	.	+	.	gene_id gene1; transcript_id=t2; name EDEN;
ctg123	.	exon	1300	1500	.	+	.	gene_id gene1; transcript_id=t1; name EDEN;
ctg123	.	exon	1050	1500	.	+	.	gene_id gene1; transcript_id=t1; name EDEN;
tg123	.	exon	1050	1500	.	+	.	gene_id gene1; transcript_id=t2; name EDEN;
ctg123	.	exon	3000	3902	.	+	.	gene_id gene1; transcript_id=t1; name EDEN;
ctg123	.	exon	5000	5500	.	+	.	gene_id gene1; transcript_id=t1; name EDEN;
ctg123	.	exon	5000	5500	.	+	.	gene_id gene1; transcript_id=t2; name EDEN;
ctg123	.	exon	7000	9000	.	+	.	gene_id gene1; transcript_id=t1; name EDEN;
ctg123	.	exon	7000	9000	.	+	.	gene_id gene1; transcript_id=t2; name EDEN;
ctg123	.	CDS	1201	1500	.	+	0	gene_id gene1; transcript_id=t1; name EDEN;
ctg123	.	CDS	3000	3902	.	+	0	gene_id gene1; transcript_id=t1; name EDEN;
ctg123	.	CDS	5000	5500	.	+	0	gene_id gene1; transcript_id=t1; name EDEN;
ctg123	.	CDS	7000	7600	.	+	0	gene_id gene1; transcript_id=t1; name EDEN;
Ctg123	.	CDS	1201	1500	.	+	0	gene_id gene1; transcript_id=t2; name EDEN;
ctg123	.	CDS	5000	5500	.	+	0	gene_id gene1; transcript_id=t2; name EDEN;
Ctg123	.	CDS	7000	7600	.	+	0	gene_id gene1; transcript_id=t2; name EDEN;

1) sequence id 2) source 3) feature type (9 possibilities) 4) start 5) end 6) score 7) strand 8) phase 9) attributes *tag value;*

! Features grouped by a **common attribute** (gene_id / transcript_id)

Introduction: Formats: GFF3

```
##gff-version 3.2.1
##sequence-region ctg123 1 1497228
```

← Header

- 9 columns
- 1 feature = 1 line

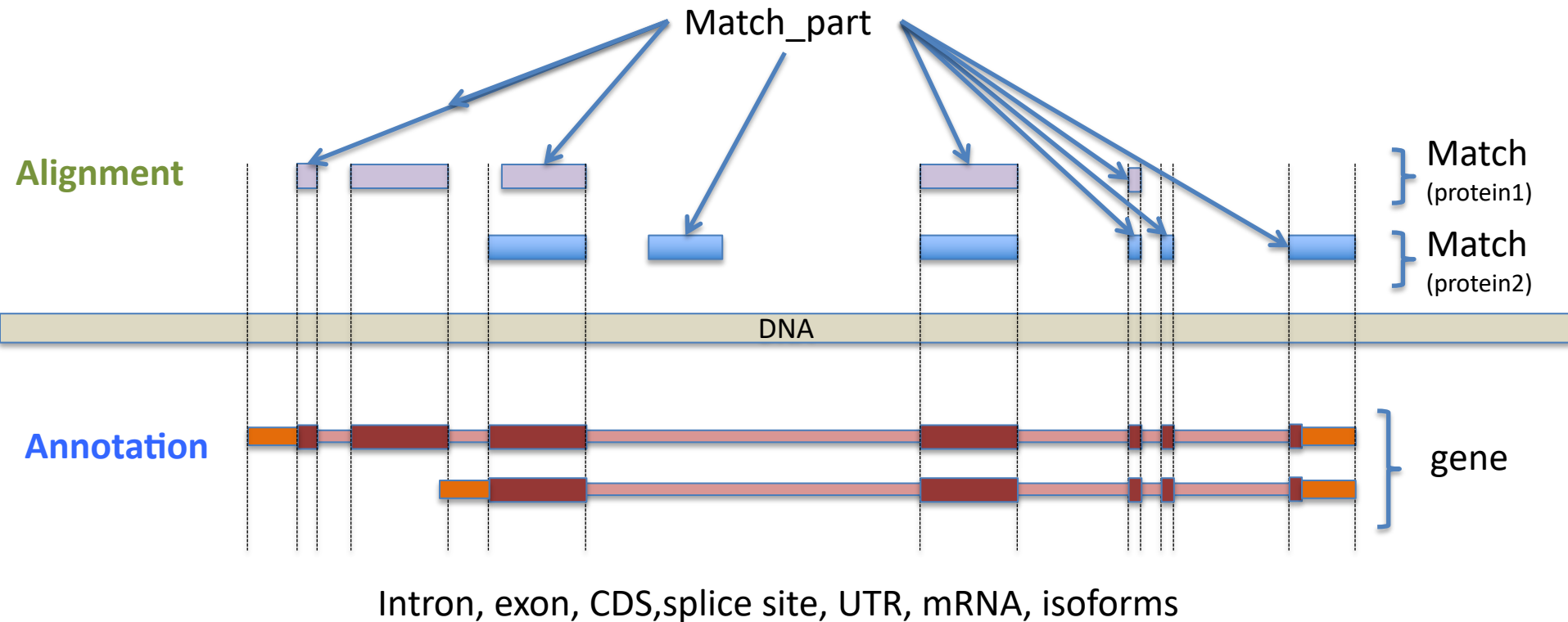
Ctg123	.	Gene	1000	9000	.	+	.	ID=gene1;Name=EDEN
ctg123	.	mRNA	1050	9000	.	+	.	ID=mRNA1;Parent=gene1;Name=EDEN.1
ctg123	.	mRNA	1050	9000	.	+	.	ID=mRNA2;Parent=gene1;Name=EDEN.2
ctg123	.	exon	1300	1500	.	+	.	ID=exon1;Parent=mRNA3
ctg123	.	exon	1050	1500	.	+	.	ID=exon2;Parent=mRNA1,mRNA2
ctg123	.	exon	3000	3902	.	+	.	ID=exon3;Parent=mRNA1
ctg123	.	exon	5000	5500	.	+	.	ID=exon4;Parent=mRNA1,mRNA2
ctg123	.	exon	7000	9000	.	+	.	ID=exon5;Parent=mRNA1,mRNA2
ctg123	.	CDS	1201	1500	.	+	0	ID=cds1;Parent=mRNA1;Name=eden1
ctg123	.	CDS	3000	3902	.	+	0	ID=cds1;Parent=mRNA1;Name=eden1
ctg123	.	CDS	5000	5500	.	+	0	ID=cds1;Parent=mRNA1;Name=eden1
ctg123	.	CDS	7000	7600	.	+	0	ID=cds1;Parent=mRNA1;Name=eden1
Ctg123	.	CDS	1201	1500	.	+	0	ID=cds2;Parent=mRNA2;Name=eden2
ctg123	.	CDS	5000	5500	.	+	0	ID=cds2;Parent=mRNA2;Name=eden2
Ctg123	.	CDS	7000	7600	.	+	0	ID=cds2;Parent=mRNA2;Name=eden2

1) sequence id
 2) source
 3) feature type
 (SO term = 2278 possibilities)
 4) start
 5) end
 6) score
 7) strand
 8) phase
 9) attributes
tag=value

! Features are grouped by **parent** relationship

Introduction: Formats: GFF3

/!\ different type of gff: **annotation** / **alignment** / other



The main steps in genome annotation

1

QC assembly



2

Structural
annotation



EuGene-EP

3

Manual
curation



4

Functional
annotation



5

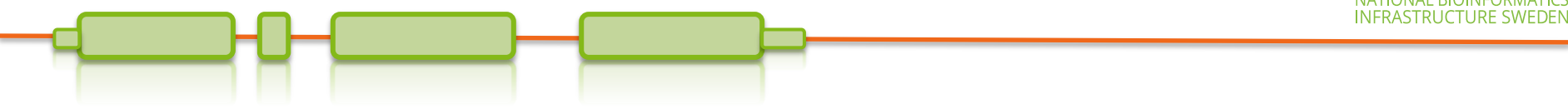
Downstream
analysis

Submission





Before all annotations

- 
- Get the best assembly! The quality of the assembly will heavily influence the quality of the annotation
 - ☐ SNP-errors can change start/stop-codons
 - ☐ Indels can cause frame-shifts
 - ☐ High fragmentation could break loci
 - ☐ missing loci cannot be annotated
- => Annotation tools have difficulties to deal with those problems
- Freeze the assembly!
 - => Updating assembly ~ annotation from scratch

Introduction: Before annotation

Always check :

- Fragmentation (N50, number of sequences, how many small contigs)
- Sanity of the fasta file (Ns, IUPAC, lowercase nucleotides)
- Completeness / duplication / fragmentation
- Presence of Organelles
- Other (GC content, how distant from other species)





BUSCO used on assembly and annotation

Example of output:

```
# BUSCO version is: 3.0.2
# The lineage dataset is: fungi_odb9 (Creation date: 2016-02-13,
number of species: 85, number of BUSCOs: 290)
#
# Summarized benchmarking in BUSCO annotation for file genome.fa
# BUSCO was run in mode: genome
```

C:98.6% [S:97.9%,D:0.7%] ,F:0.0%,M:1.4%,n:290

```
286 Complete BUSCOs (C)
284 Complete and single-copy BUSCOs (S)
2   Complete and duplicated BUSCOs (D)
0   Fragmented BUSCOs (F)
4   Missing BUSCOs (M)
290 Total BUSCO groups searched
```

The different approaches

- Similarity-based methods :

These use similarity to annotated sequences like proteins, cDNAs, or ESTs

- *Ab initio* prediction :

Likelihood based methods

- Hybrid approaches :

Ab initio tools with the ability to integrate external evidence/hints

- Comparative (homology) based gene finders :

These align genomic sequences from different species and use the alignments to guide the gene predictions

- Chooser, combiner approaches :

These combine gene predictions of other gene finders

- Pipelines :

These combine multiple approaches

Types data used vs methods

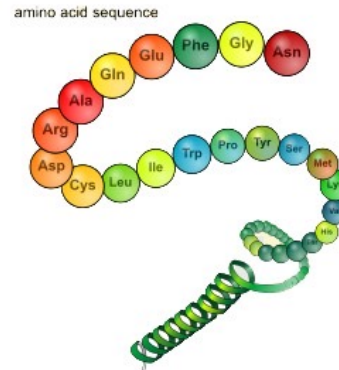
Annotation approach

∅

This space intentionally left blank.

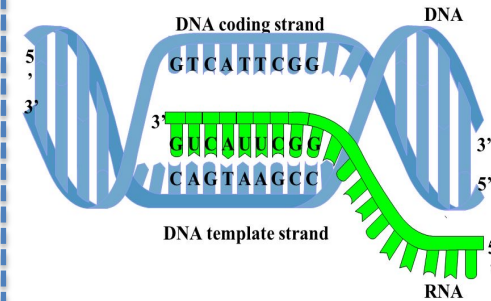
Proteins

Known amino acid sequences from other organisms



Transcripts

Assembled from RNA-seq or downloaded ESTs



Similarity		X	X
Pure ab initio	X		
Hybrid	X	X	X
Comparative	X	X	X
Chooser/combiner	X	X	X
Pipeline	X	X	X

Strengths :

- Fast and easy
- Annotate unknown genes
- Sensitivity ok
- Need no external evidence

Limits :

- No UTR
- No alternatively spliced transcripts (augustus does)
- Bad specificity (Over prediction of exons or/and genes)
- **Training** needed (Need external evidence)

Common errors in annotation:

- Split single gene into multiple predictions
- Fused with neighboring genes
- Less accurate than homology based method:
 - Exon boundaries
 - Splicing sites

Exercises

https://nbisweden.github.io/workshop-genome_annotation_elixir/labs/augustus

The different approaches

- Similarity-based methods :

These use similarity to annotated sequences like proteins, cDNAs, or ESTs

- *Ab initio* prediction :

Likelihood based methods

- Hybrid approaches :

Ab initio tools with the ability to integrate external evidence/hints

- Comparative (homology) based gene finders :

These align genomic sequences from different species and use the alignments to guide the gene predictions

- Chooser, combiner approaches :

These combine gene predictions of other gene finders

- **Pipelines :**

These combine multiple approaches

Types data used vs methods

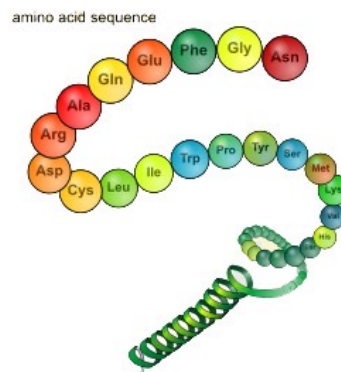
Annotation approach

∅

This space intentionally left blank.

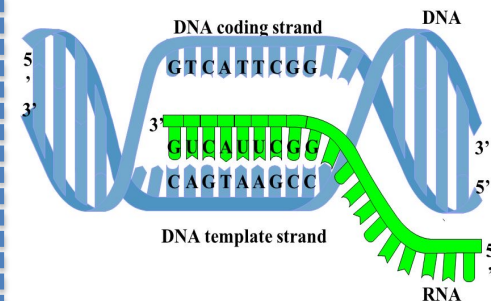
Proteins

Known amino acid sequences from other organisms



Transcripts

Assembled from RNA-seq or downloaded ESTs

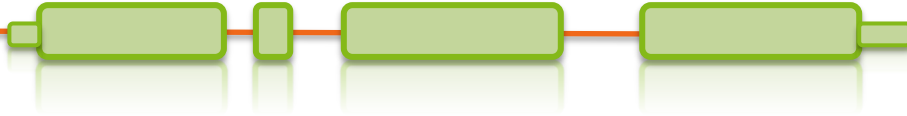


Similarity		X	X
Pure ab initio	X		
Hybrid	X	X	X
Comparative	X	X	X
Chooser/combiner	X	X	X
Pipeline	X	X	X

Get the data

- **Genome**
 - Fasta or gff format
- **Repeats**
 - Fasta or gff format
- **Proteins**
 - Fasta format
 - Uniprot/swissprot
 - Close related species
- **RNAseq**
 - Fasta or gff format
 - Same individual best
 - SRA (Sequence Read Archive)

Repeats

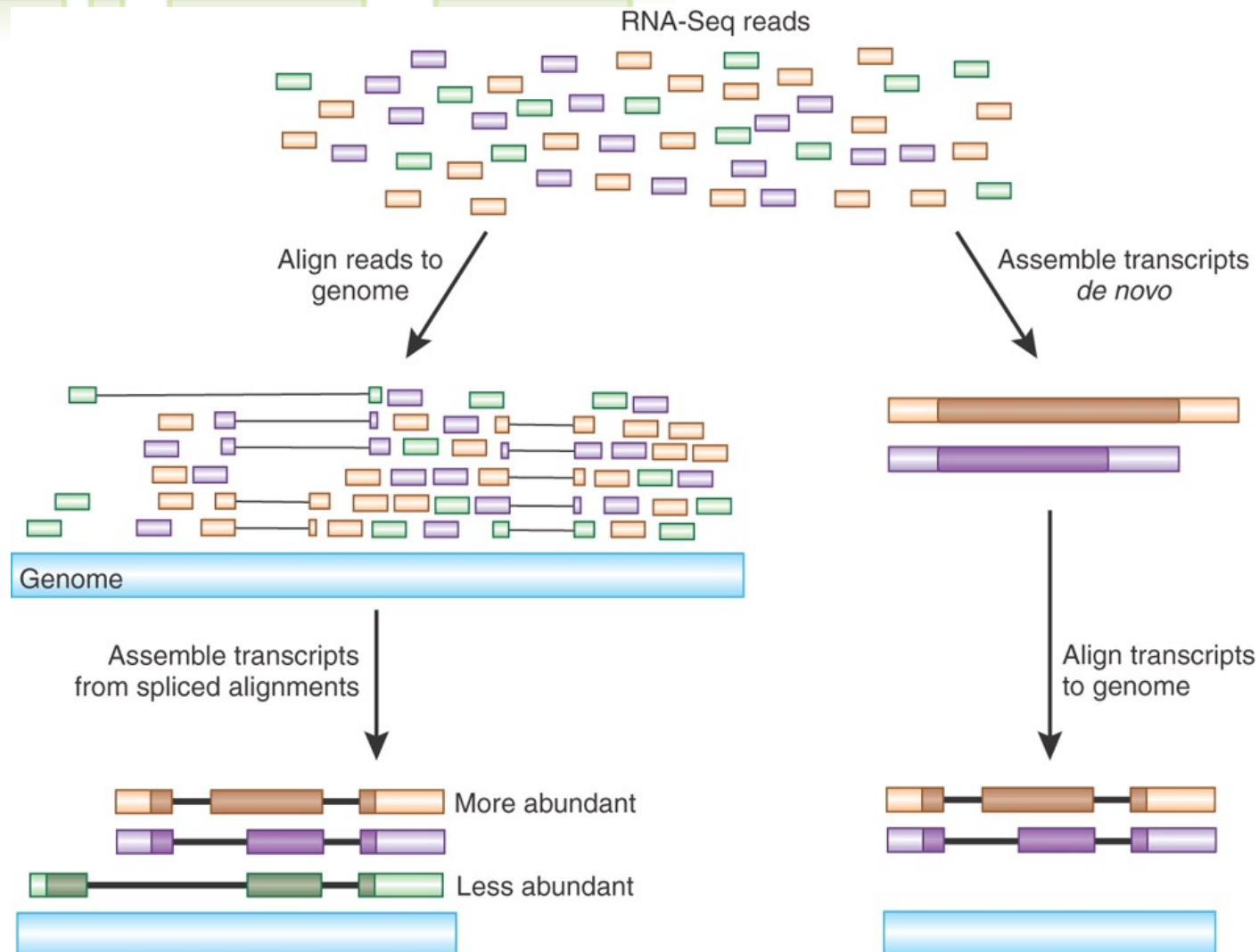


First of all: Repeat Masking

- Repeatmodeler to find new repeats
 - <http://www.repeatmasker.org/RepeatModeler/>
 - Repeatmasker to mask known repeats
 - <http://www.repeatmasker.org>
- + Save time
- + Increase quality of the gene coding annotation

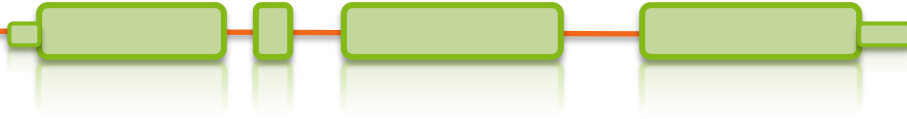
- **Proteins :**
 - Related to pre-existing data
 - Proteins from model organisms often used => bias?
 - Proteins can be incomplete
 - Protein can be wrong (PE)
 - No UTR
- **RNAseq :**
 - Hard to catch low expressed / peculiar expressed (stage of life, condition, etc...) / isoforms
 - short-reads:
 - Transcriptome assembly errors
 - Long-reads:
 - error rate / frameshift / indels

Assembly of transcripts



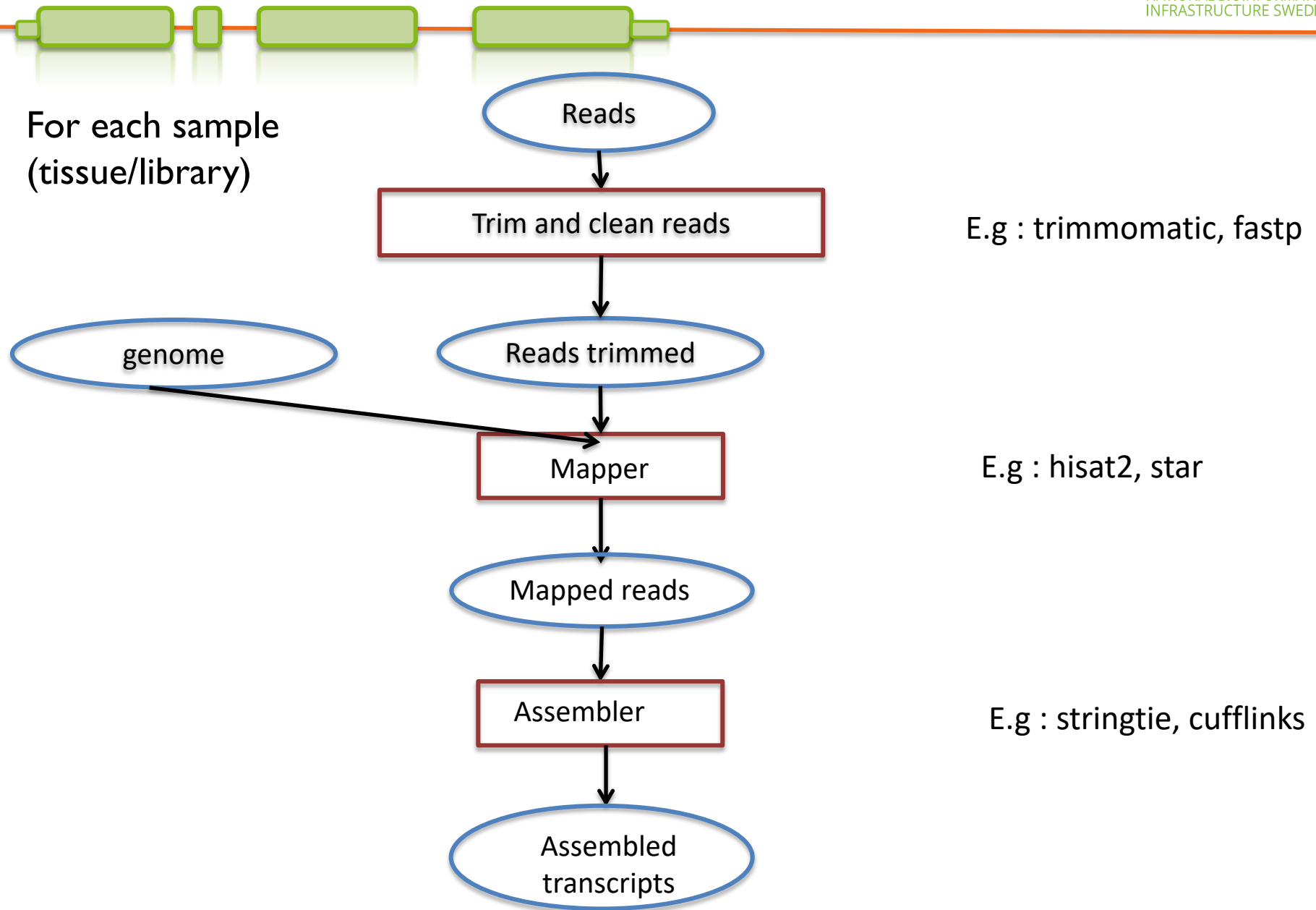
- Most used programs (latest release date):
 - Trinity (Sept 2018)
 - SOAPdenovo-Trans (July 2013)
 - Trans-ABYSS (Feb 2018)
 - Velvet+Oases (March 2015)
- Originally SOAPdenovo, ABYSS and Velvet for de novo genome assembly
- “SOAPdenovo-Trans incorporates the error-removal model from Trinity and the robust heuristic graph traversal method from Oases.”

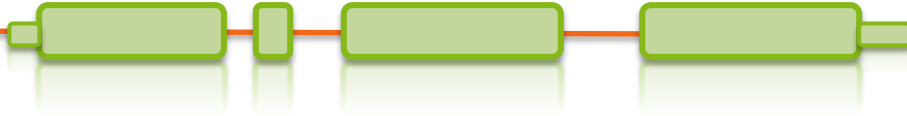




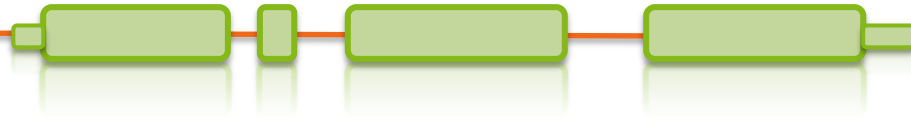
- No reference needed
- Many programs available
- Lots of potential transcripts. Filter!

Genome guided transcriptome assembly





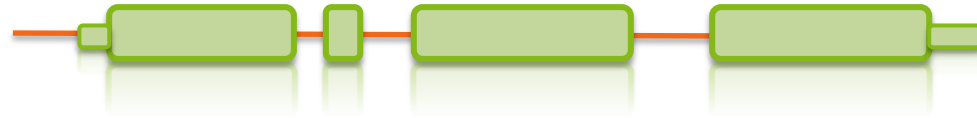
- Need a very good reference (genome most of the time)
- Can use existing annotation (GTF/GFF file) (in option for stringtie)
- Can detect novel transcripts



RNAseq

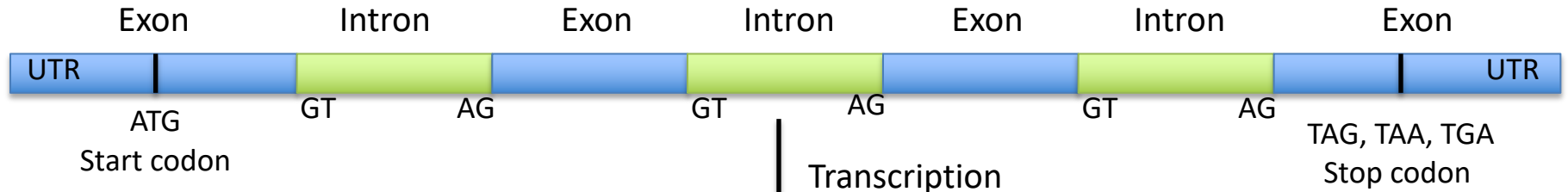
How does it look
when it does not look good?

RNA-seq noise

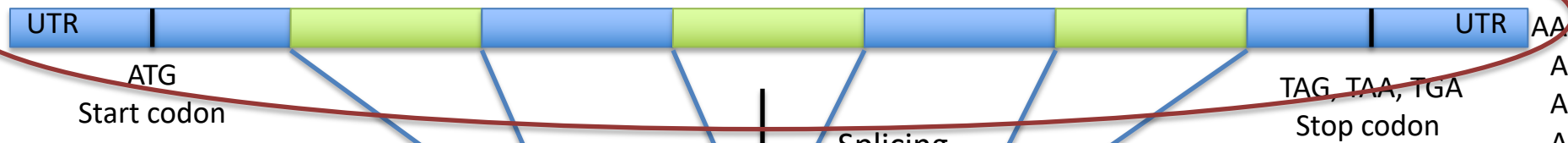


Types of data used: RNA-seq

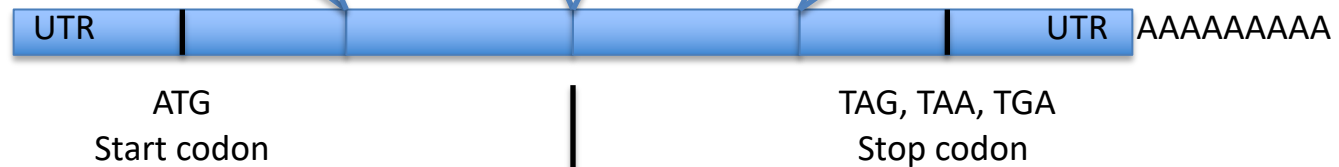
DNA



Pre-mRNA

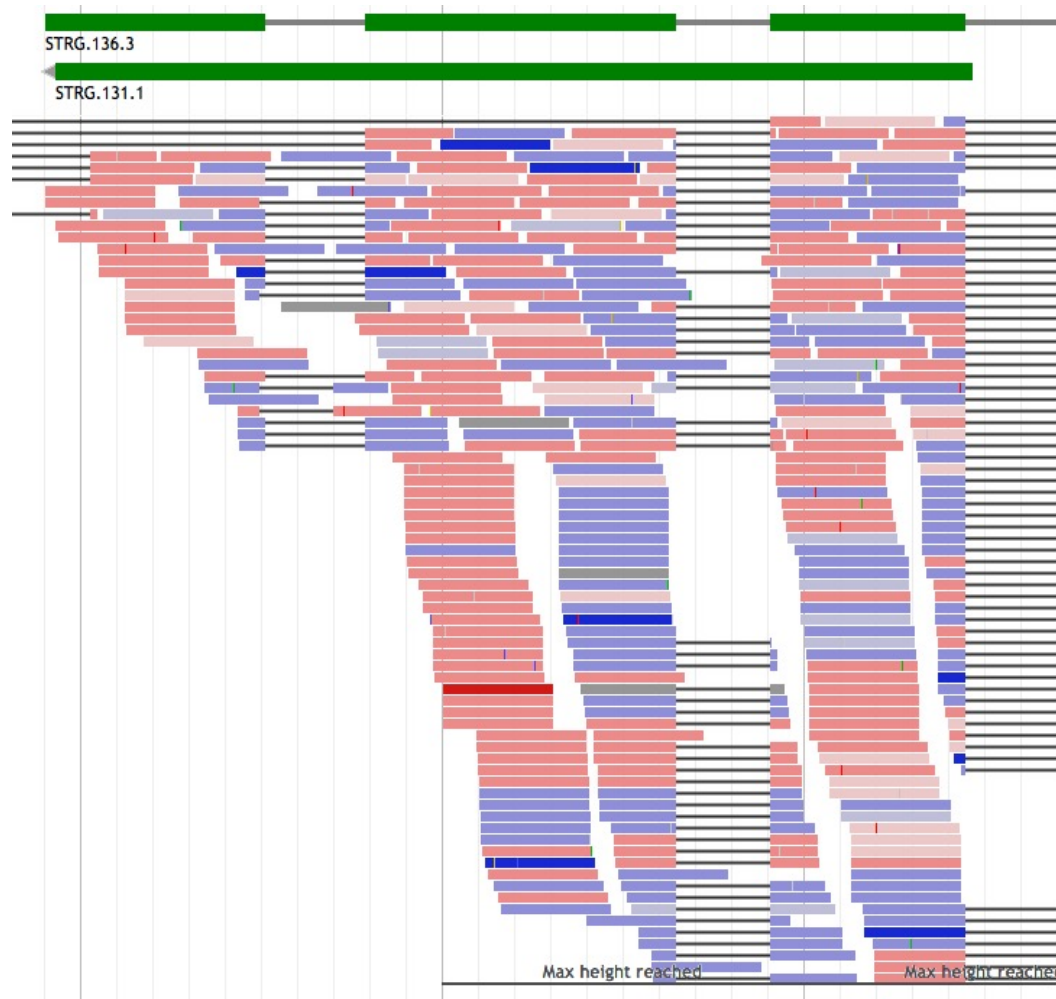


mRNA

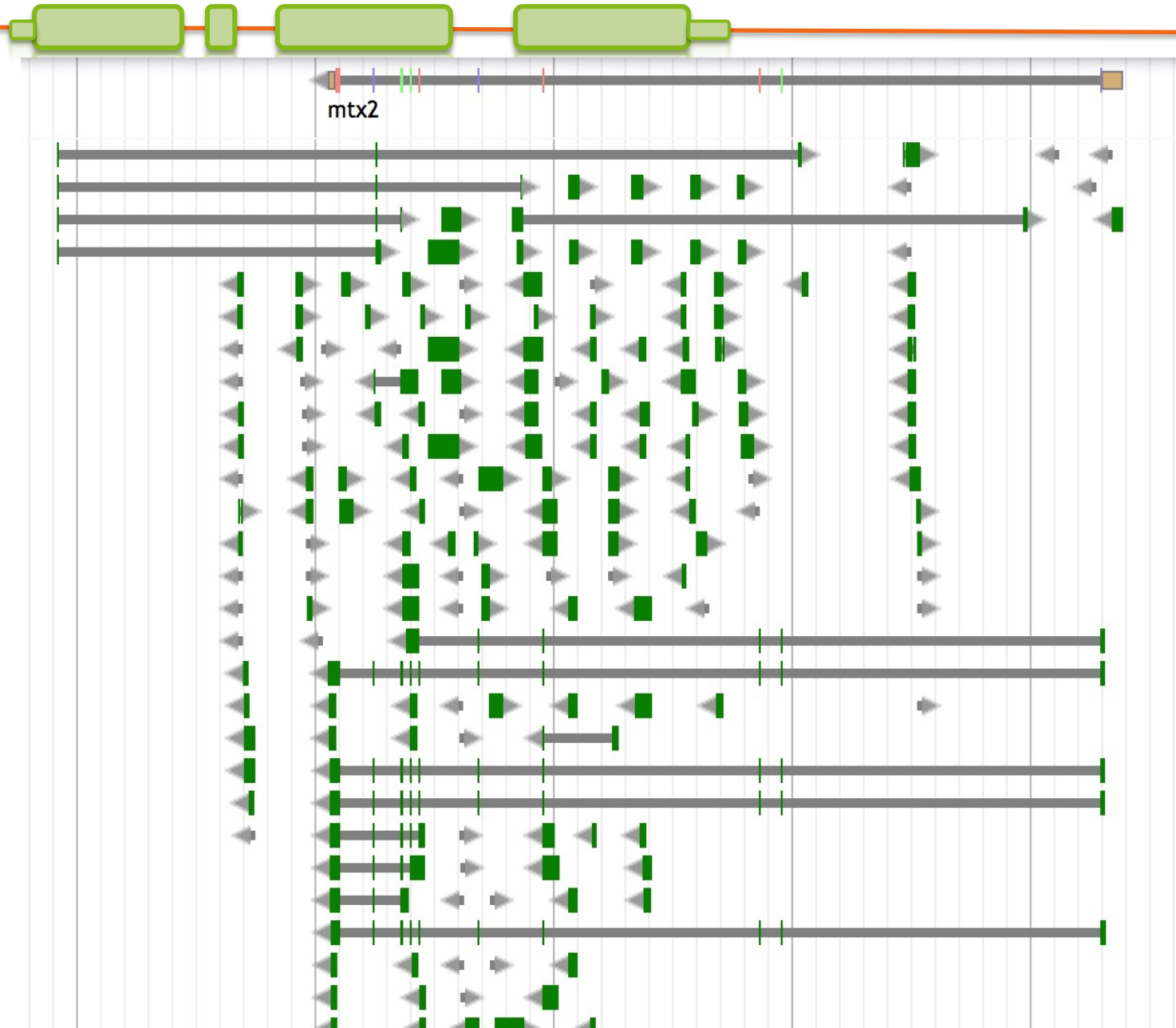


AA
A
A
A
A
A

RNA-seq – pre-mRNA noise



Trinity noise



- RNAseq data should always be included in an annotation project
- From the same organism as the genomic data => unbiased
- Can be used before annotation or after to improve an annotation already existing
- Sample different tissues or life stages if possible
- Avoid gonads and brain; muscle is good
- /!\ Can be very noisy (tissue/species dependent), can include pre-mRNA
- Combining method is best if possible

MAKER lecture

https://nbisweden.github.io/workshop-genome_annotation_elixir/lectures/Structural_annotation_MAKER_Norway2021.pptx

Exercises

https://nbisweden.github.io/workshop-genome_annotation_elixir/labs/maker_evidence

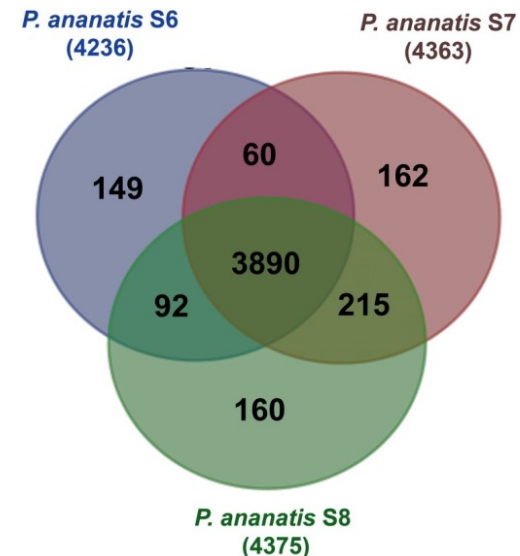
https://nbisweden.github.io/workshop-genome_annotation_elixir/labs/augustus_training

https://nbisweden.github.io/workshop-genome_annotation_elixir/labs/maker_abinitio_evidence_driven



After structural annotation Assessing an annotation

- Simple statistics (number genes / number exon per gene)
- **BUSCO** (and compare against assembly result)
- Protein/transcript evidence (AED score in MAKER)
- Comparative genomics (OrthoMCL)
- Domain / Function attached
- Visualization



Selection of most common visualization or/and Manual curation tools

Name	Standalone	Web tool	Manual curation	year	comment
Artemis	X		X	2000	Can save annotation in EMBL format
IGV	X			2011	Popular
Savant	X			2010	Sequence Annotation, Visualization and ANalysis Tool. enable Plug-ins
Tablet	X		X	2013	
IGB	X			2008	enable Plug-ins. Can load local and remote data (dropbox, UCSC genome, etc)
Jbrowse		X		2010	GMOD (successor of Gbrowse)
Web Apollo		X	X	2013	Active community (gmod). Based on Jbrowse. Real-time collaboration
UCSC		X		2000	A large amount of locally stored data must be uploaded to servers across the internet
Ensembl genome browsers		X		2002	A large amount of locally stored data must be uploaded to servers across the internet

Exercises

[https://nbisweden.github.io/workshop-
genome_annotation_elixir/labs/annotation_assessment](https://nbisweden.github.io/workshop-genome_annotation_elixir/labs/annotation_assessment)

Closing remarks

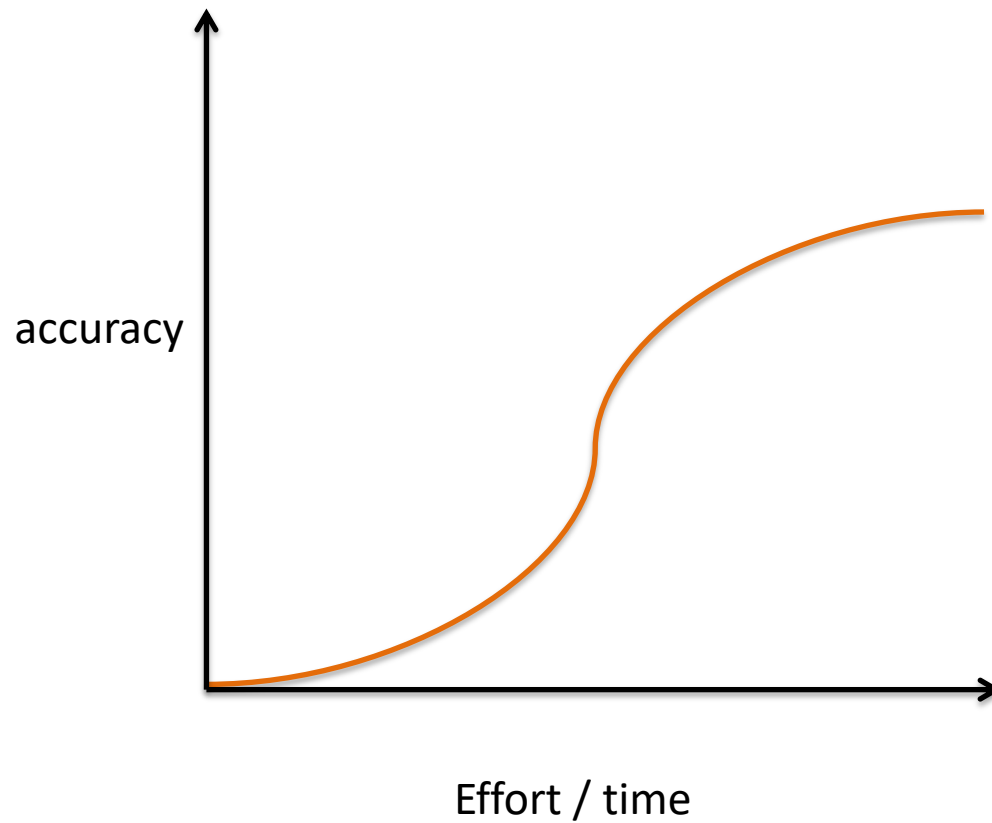
- >100 annotation tools – as many methods
(https://github.com/NBISweden/GAAS/blob/master/annotation/knowledge/annotation_tools_genome.md)
- 6 main class of approaches (Similarity-based, *ab initio*, hybrid, comparative, combiner, pipeline)

How to choose Method:

- Scientific question behind (need of a conservative annotation vs exhaustive)
- Species dependent (plant / Fungi / eukaryotes)
- phylogenetic relationship of the investigated genome to other annotated genomes (Terra incognita, close, already annotated).
- Data available (hmm profile, RNAseq, etc...)
- Depending on computing resources (*ab initio* ~ hours < **VS** > pipeline ~ weeks)

- Several *ab-initio* tools together give better result than one alone (they complement each other)
- Pipelines give good results
MAKER2 the most flexible, adjustable
- Most methods only build gene models, no **functional inference**
- No annotation method is perfect, they make mistakes !!
- Annotation requires **manual curation**
- As for assembly, an annotation is never finished, it can always be improved
=> e.g. Human (to know when to stop)
- Submit your annotation in public archive

Effort versus accuracy





THE END

<https://github.com/NBISweden/GAAS>

<https://github.com/NBISweden/AGAT>

<https://github.com/NBISweden/pipelines-nextflow>