

GENOME ANNOTATION WORKSHOP

MIRIAN T. N. TSUCHIYA
DATA SCIENCE POSTDOCTORAL FELLOW
DATA SCIENCE LAB - OCIO



WORKSHOP INFO

**INTRODUCTIONS, INFO, CODE OF
CONDUCT**

WORKSHOP INFO

- Code of Conduct:

[https://docs.carpentries.org/topic_folders/policies
/code-of-conduct.html](https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html)

- Use welcoming and inclusive language
- Be respectful of different viewpoints and experiences
- Gracefully accept constructive criticism
- Focus on what is best for the community. Show respect and courtesy towards other community members



WHAT IS GENOME ANNOTATION?

- Genome annotation is the process of identifying different elements in a genome assembly:
 - Structural
 - Repetitive elements
 - Genes (with introns and exons)
 - Functional
 - What does each gene do?

GENE PREDICTION

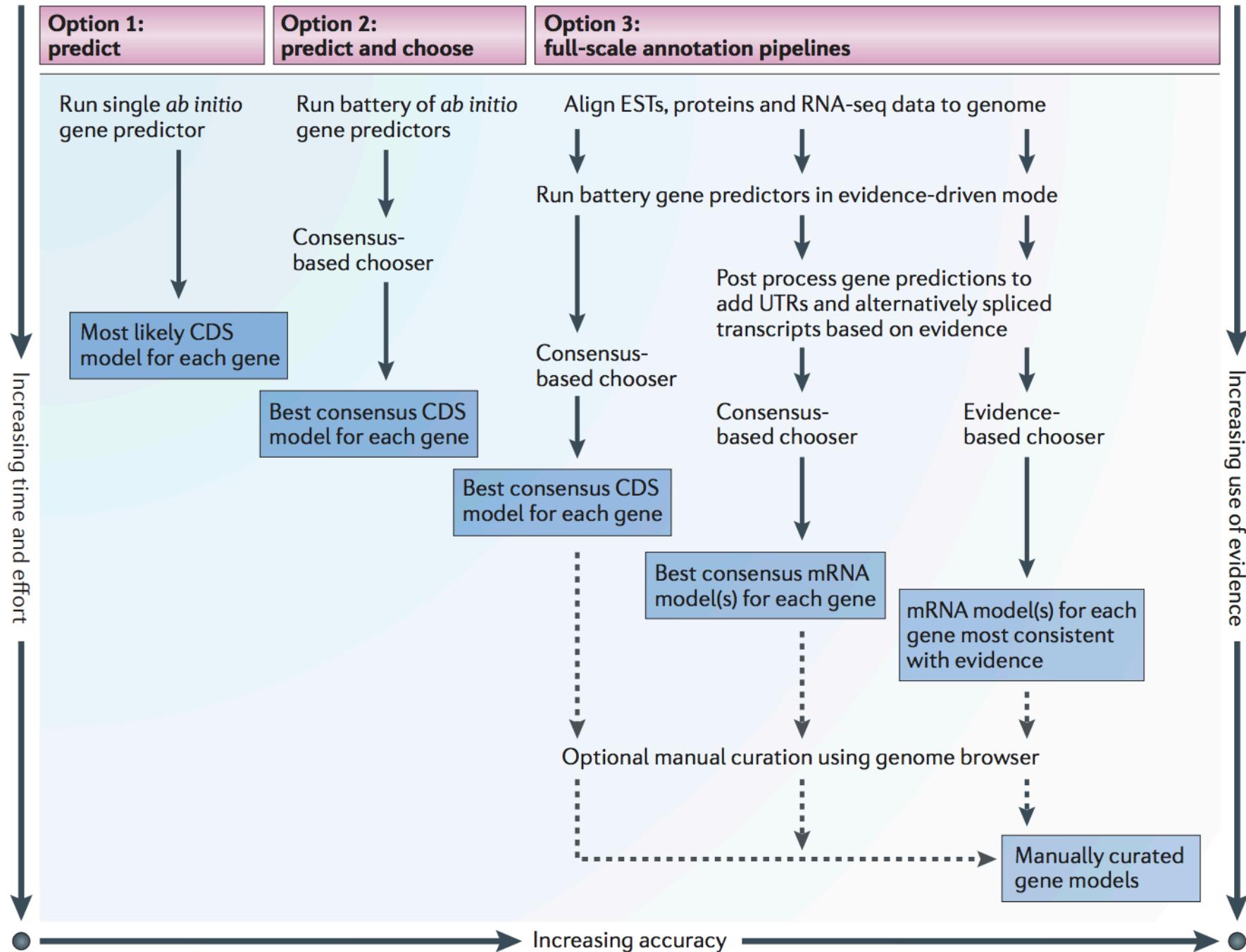
ab initio

- use only the query sequence

evidence-driven

- use external evidence

Combining both approaches is the best option



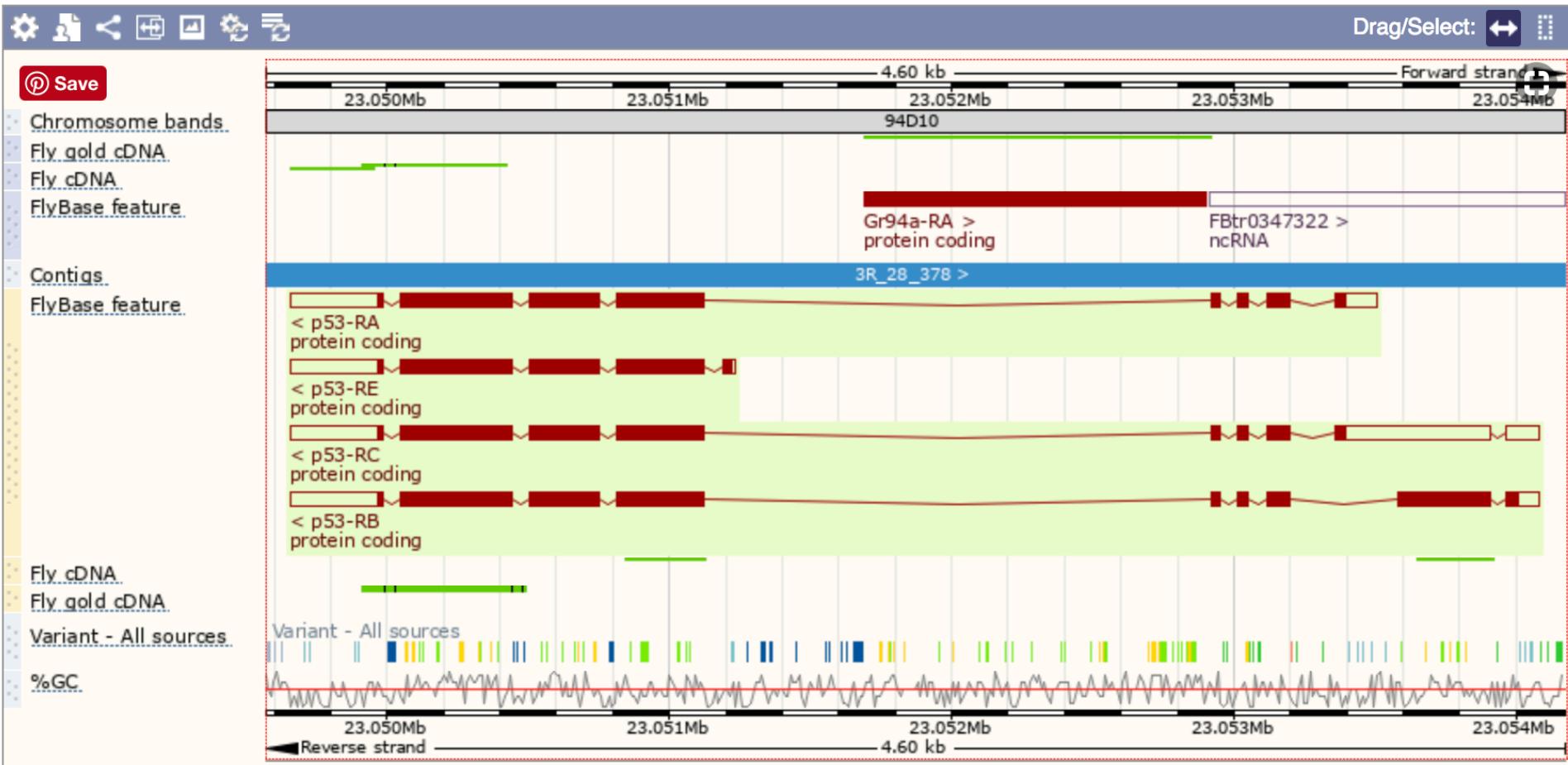
Chromosome 3R: 23,049,569-23,054,170



ATGTTGCAAAATATGACAACCTTTAATAAATTCACTTTGGCTTTGTACAGAGTATGTTGTTGGTAGACAAAAGTTACGTAGTCACA
TAGAACTGCAGTTAACACAGTACTATCAAAAATATAAAATGTTTCTTTCGGCCATCGAACATGCCAAAAGGGGAATATCACCGTTAAA
CGCCTTACGAACAAGGACTAACAAACTTAAGATACTAATAGCTATGAGTACAGATGCCAACACATTCAAAGGTATCTGATACC
TGGGAGATTGTCGACCAGATCAGAAGTCATGGCAGCTCGTAGGCACGTTCTGAAAGGAAAGTCGTAAGTAAAGCTTAATCTCTTGCTCCT
CCGATTCACTGCTTACTCTAAGGCTCAGCAATTGTTGGCATGGCAGCTAGATTCTCTGGTTGGATTGCGCAGGACTTCAGCCGCCGC
CTCCTTAATCATGCCCTCGATGCTCTGCAGCAGCCATTCTTATTGGGGCACGTAATAGCCAGACGGTAATGCCATCCGGTGTCCCGCGACA
CGTTCCACTCTGCGCGGAGTCGTCGAGCTCGCTATCATTGCTCTCGTCTCGTCTTATAGCAATGCACCGACGCCACCTGGAC
GGCTCATCTTCTCGGCGCTTCCGGCACGGACTTGCCTTGTCTTATTGAGCTGGCGTCTGGATGCGATCCCCTGGGACAGT
ACATATTAAACATGTATAACATGCTGCCCCACGATATGCCGCTGCAAGAAAGAAAACACGTTAAGCTTGTCTAGCCATCTAGAGTTTGC
TGTACCTTACCATGCTTCTCCAGGCAGAAGACTAAGGAAGTTCTTCGCCGATACACGAGTTTGGCAGACGAACCTGAAGGCCAGG
GTCTGGCGCGTGGAGCCACTGCGGGTTACAGACGGCTCATGTTAGGGGACTACAACGGAAAACGCTCGGAAATTCCCTGCCGAG
CATTCCACAATATACACTGTTGGATTCTCGCTGCGCAGCAAGCTCGCGCATTTGCGTTATTGGCCGTCATAAAGGTTGTACAAT
GATAGTTAAATAATCCTGTTGATCTGTATTGTTATCTTCACTTACAAGGCTAACGCTAACGGTATTGACAGCGGACCAACGGAGACTC
ACATCATTGGAGAAGCAAAGGAACACACGCAAATTAAAGTGGTGGATGGCATTAGACTTGAACGTCCACGTTGAAGGCCCTGTT
CATCCGGATGTAGAGCTTGTTCAGCGGAATCGAGTACATCCAAGAGACTTGGCGGCTCATCCAGAACCATGCTGAAGCAATAACCAACCGA
TGTGATTCTCTAGCTTGGGCGAGCGTGTGCGCTGGATCTGAATGCTCTGCAGCATATTGCGCAGCACGGATTGCTGTAAAGAAAAAA
CAAGATATTATTGAATGTCACTTAACCGCAATCATATAAGGGTTCACACTTGCCTCATAGAGCTGATAGAGCTCATGTTCCAGCTTGC
TCGAGTTACAAATGGACTGGCGATTGTTGTTGATTCTATTCTATTGCAAGTGTGAAATTCTGCTAAACGATGACGACGGAGGGAG
TGCACCGTGCTAACCGGCTAGATAAGAGTGTGTTCTGCTCTCCAACTGAAAGTGAACCTCGAAGCGACTTGCACGTCATGTCGCTTAT
GAAATTGCAGGCAGCGGCTGAGTCACGCAAGGAATGCCGCTTCCCTCACGCTTCTTATGCTCTTGTGGTAGTAGCACAACGC
ATGCTAGTTGGTTCTGGACGGCGGTGAAATACATGCGTTCGCGATGTGTCGGCTACTTGACAGGTCTCTAGATGG
CATCAGTTGGTAGTTGGAAAACAATTAAAGCAGATATTAAATGCTGTGGAAAGTGCCTCACAGTTGTTATTAAATTGTCGAGAGAAAATGGACT
TCACCAAGCGACTACGCGCATCGCGTATGGTAAATTCTGACGATCATACTGATAGGTTATGACCGTCTCGGACTCTGGCAATCGA
TATCGGGCGGGCGCTGTGAAAGATTCCGCTTCTAAAGGCAAATCGGCTTGTCTGCTGTGGCAATTGCAATTGCTTGGTTACGG
CGCGCAAATCTACAAGGAGTACCAAGGAGGTAGATCAACCTGAAGGACGCCACACTCTGTACAGCTATATGAACATTACGGTGGCTGTTA
TTAACTATGTCGCAAATGATAATCACTGACCATGTGGCAAGGTGTTGAGCAAAGTGCCTTGTGATACCCCTAAAGAATTCCGCTGG
ACAGCAGGTGCGTGTACATATCCATCGTTGGCTCTGGTCAAGACCGTGGCTTCCCTTAACAAATTGAAGTGGCTTCAACTGCAACAGA
GGCGCAGCATCCCGAGATGAGCTGATGGACCTTGTACCGCTGCTGAACAGACGGCTGGAAGCGCAGCTGCAGGAGGTGAATCTGCTGCA
CAATGGTGGTGGTGAAGGAGATTCTGTACGCTCTGAACAGACGGCTGGAAGCGCAGCTGCAGGAGGTGAATCTGCTGCAAGAGGAAGGACCA
GCTAAAGTTGACTAAATACTACCGCATGCGATTTGCGCTTGGCGATGAACTCGACAGCTGGCGTATCGCTATAGGTTGATATA
TGTGCATTGGAAAAGTATCTGACCCCAATGTCCTTGTCCATGATTCTGCGCTCATATGCCACCTGCTCGGAATAACGGTGGTTCTACAG
TCTGTACTATGCCATAGCGGACACCTTAATCATGGGCAAGCCGTACGATGGTCTGGATGCTGATCAATCTGGTTTCTCCATCTCGCT
GGCGGAGATCACATTGCTCACGCATTGTGCAACCACCTATTGGTGGCCACCCGAAGATCGGCAGTCATTCTCAGGAGATGAATCTCCAGC
ATGCGGACAGCCGCTACCGTCAAGGCACTCCACGGTTACTCTGCTGGTCAAGGTGACCAAGTACCAAATTAAACCTTGGCTGTACGAG
CTGGACATGCGACTGATCAGCAATGTCTTCTCGGCGGTGGCCAGCTTCTGCTGATCCTCGTGCAGGCCGATCTGCCCAGCGCTTCAAGAT
GCAATAGCTAATCGATGTTACCCACCTGGCTGAACAGCATCAGATTCCCGACTGCGGGAAATAATTAAAGTTAGTAAGCTATAGCTT

Chromosome 3R: 23,049,569-23,054,170

Drag/Select:



- Variant Legend
- stop gained
 - splice region variant
 - 5 prime UTR variant
 - non coding transcript exon variant
 - upstream gene variant

- missense variant
- synonymous variant
- 3 prime UTR variant
- intron variant

- Gene Legend
- Protein Coding
- Ensembl protein coding

- Non-Protein Coding
- RNA gene

There are currently 22 tracks turned off.

Ensembl Drosophila melanogaster version 94.6 (BDGP6) Chromosome 3R: 23,049,569 - 23,054,170

SOFTWARE/WRAPPERS

Augustus

SNAP

**Glimmer
HMM**

**Genemark
-ES**

FGenesh

Gnomon



MAKER
Annotate this!

Web  Apollo

OTHER ALTERNATIVES



Pros

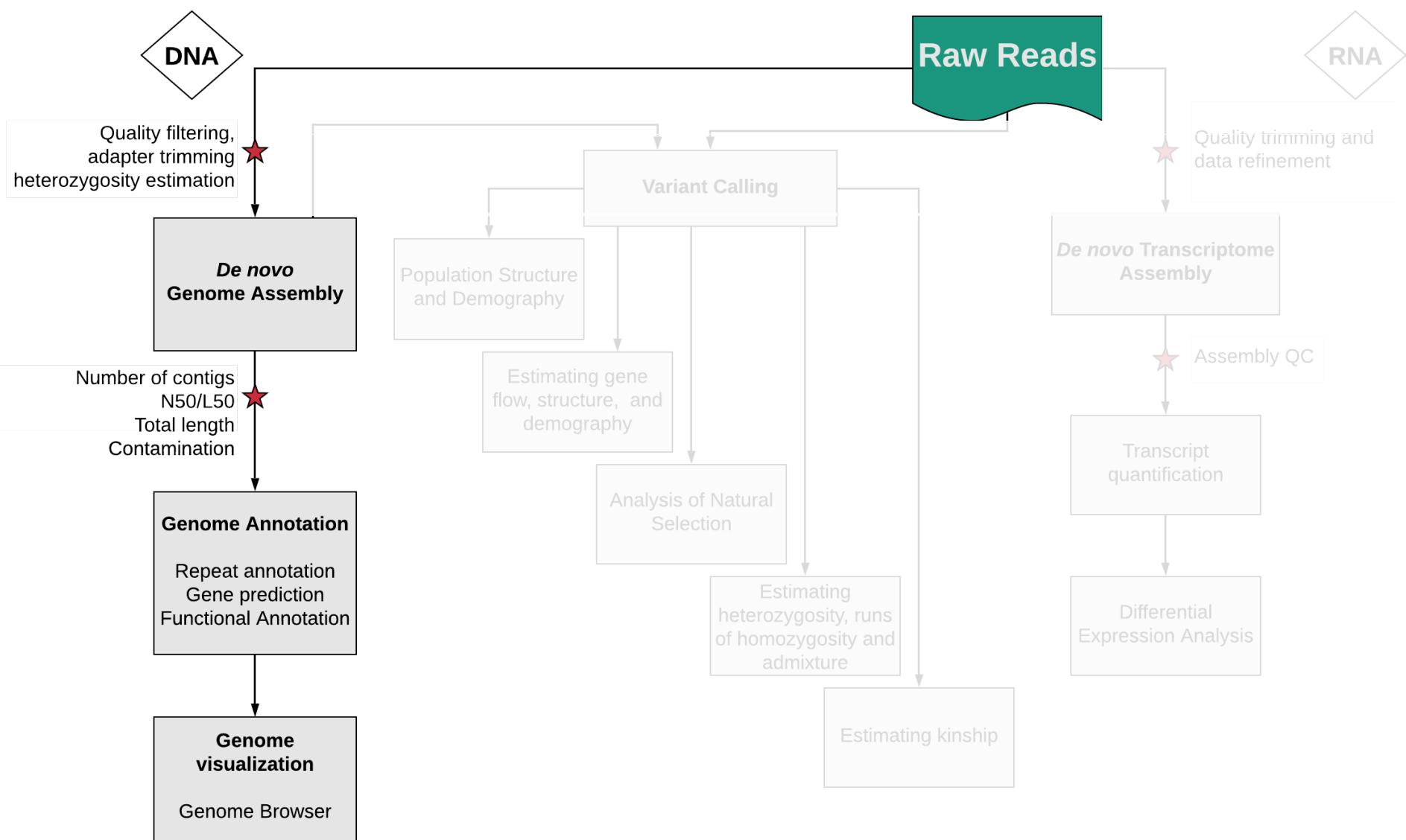
Standardized

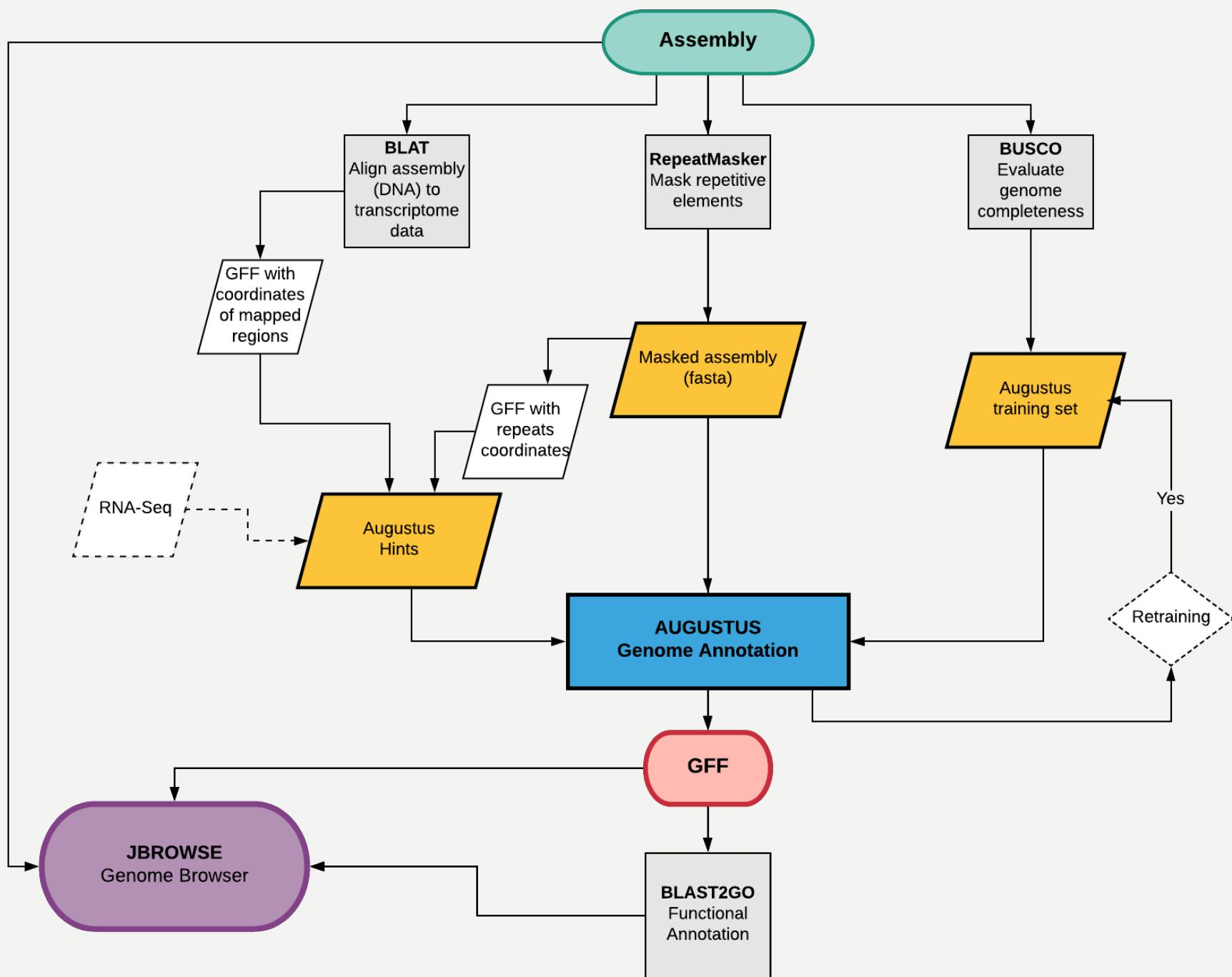
Free

Limitations

Quality requirements

Taxonomic priorities







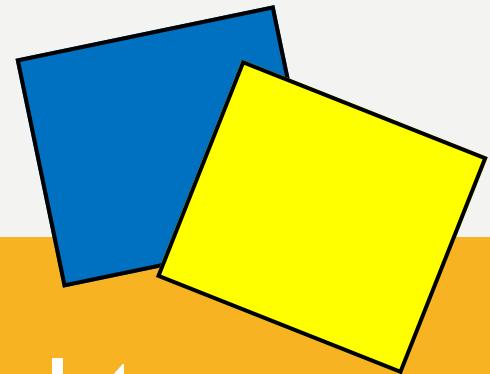
LET'S START!

FOLDER STRUCTURE

- I. Create the folder **genome_annot** inside the **smsc_2019** folder:
2. Then, create the following folders inside **genome_annot**:
 - assembly
 - busco
 - repmasker
 - blat
 - blast
 - b2go
 - augustus
 - jbrowse
 - logs
 - jobs

Why?

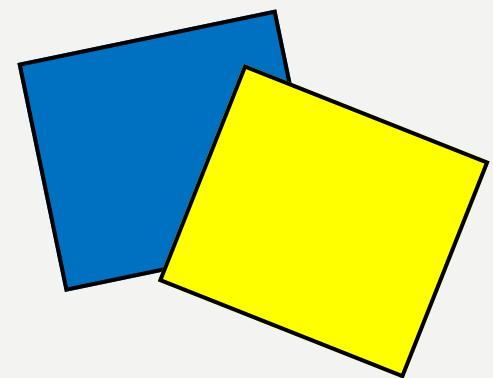
It is easier to find everything later.



ASSEMBLY STATS

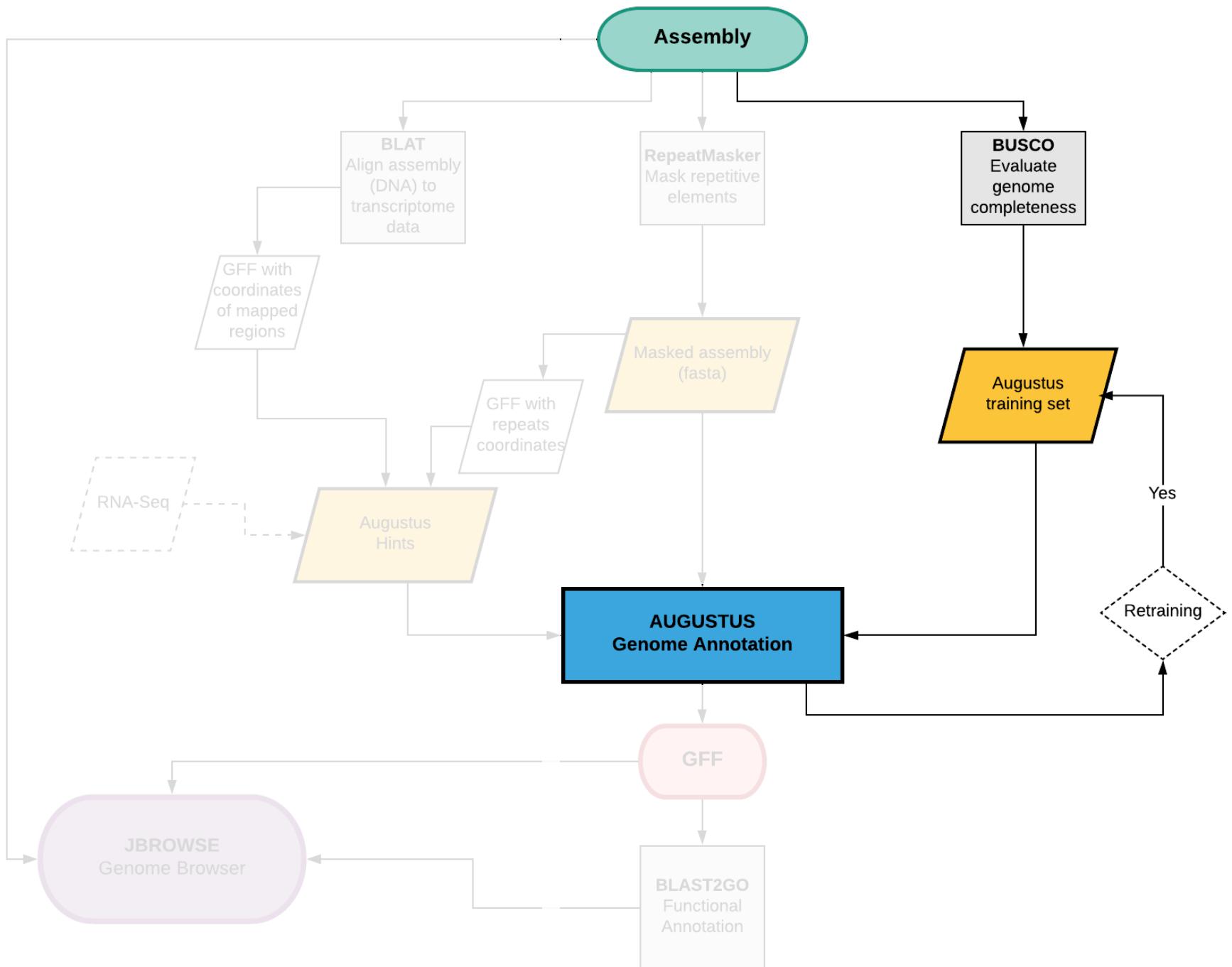
- ASSEMBLY
 - Species: *Spinus cucullatus*
(siskin_10largest.fasta)
 - Copy the file to your folder assembly. File location:
/data/genomics/workshops/SMSC_2019/siskin_10largest.fasta
- We will use a python script (available in the module assembly_stats) to get some basic info about this genome.
- We will run this part from the interactive queue. **Login using qrsh**

```
genome_annot
|   __ assembly
|   __ augustus
|   __ blast
|   __ b2go
|   __ blat
|   __ busco
|   __ jbrowse
|   __ jobs
|   __ logs
|   __ repmasker
```



BUSCO

BENCHMARKING UNIVERSAL SINGLE-COPY ORTHOLOGS



BUSCO

- Benchmarking Universal Single-Copy Orthologs

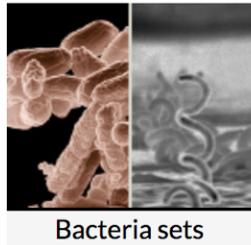
What is a ortholog?

Orthologs are genes in different species that evolved from a common ancestral gene by speciation.

BUSCO: HOW COMPLETE IS THE ASSEMBLY?

- Database: taxon-specific single copy orthologs

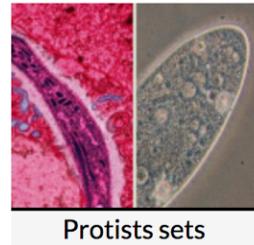
Datasets



Bacteria sets



Eukaryota sets



Protists sets



Metazoa sets



Fungi sets



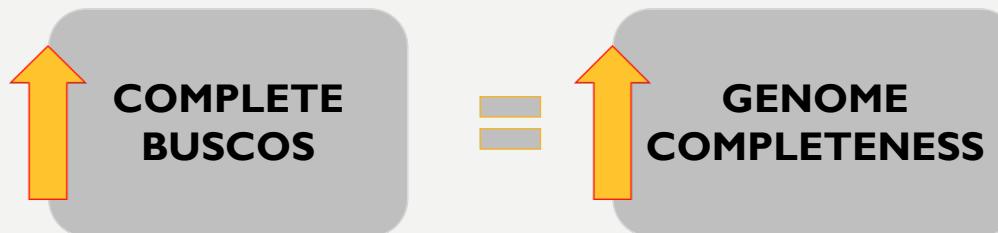
Plants set

[Download all datasets](#)

Image credits

BUSCO: HOW COMPLETE IS THE ASSEMBLY?

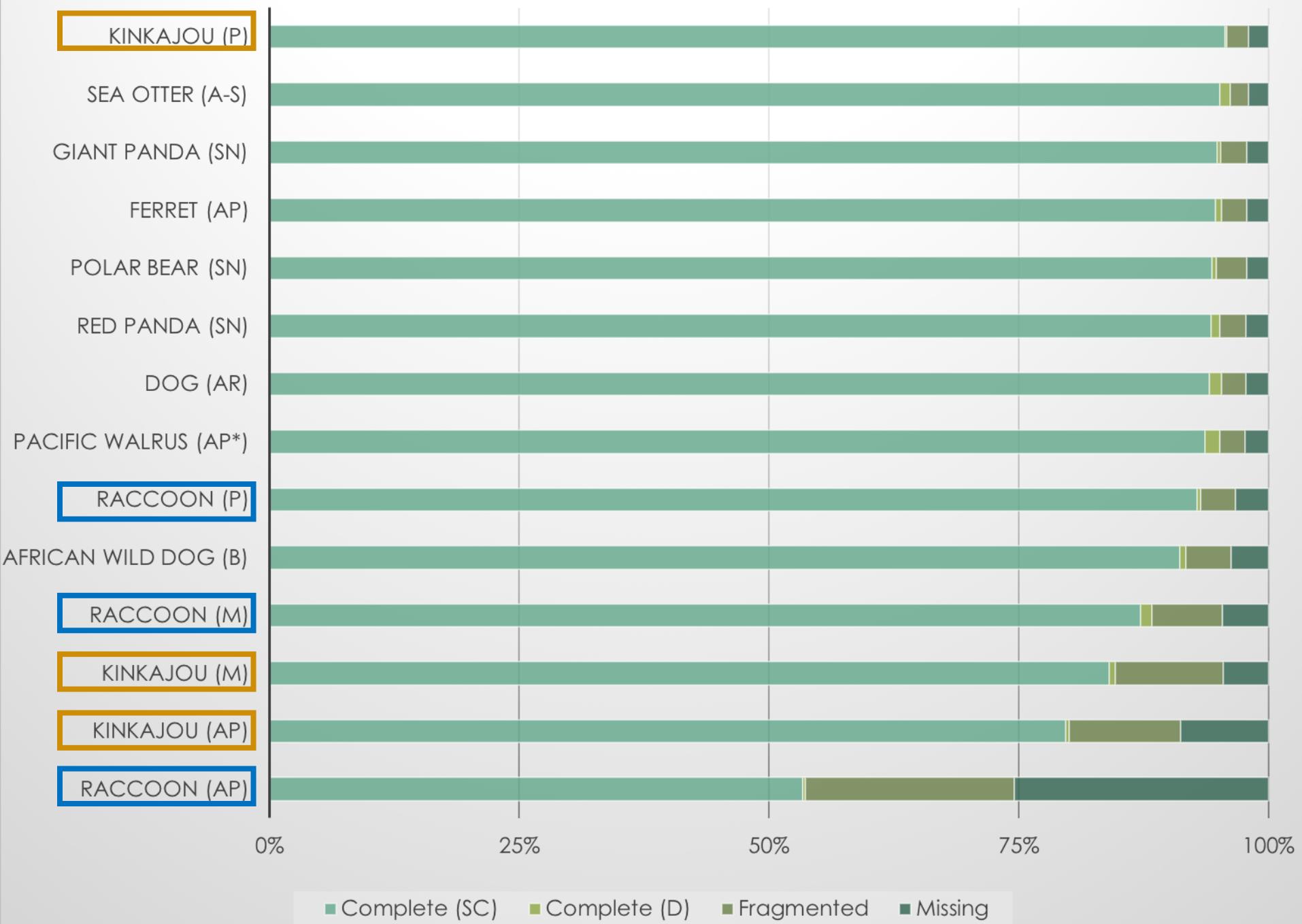
- Assessment:
 - Complete (single copy or duplicate)
 - Fragmented
 - Missing



ASSEMBLERS

		ALLPATHS-LG	Platanus	MaSuRCA
Raccoon (34X)	Number	42,696	50,007	277,099
	N50 (Mb)	0.11	1.45	0.38
	Longest (Mb)	1.83	10.59	3.43
	Total Length (Gb)	1.79	2.25	2.78
Kinkajou (48X)	Number	23,505	15,879	67,074
	N50 (Mb)	0.29	3.55	0.12
	Longest (Mb)	3.91	15.44	1.01
	Total Length (Gb)	2.05	2.21	2.3

3 paired end libraries (350 bp) + 2 mate pair libraries (3 kb and 8 kb)

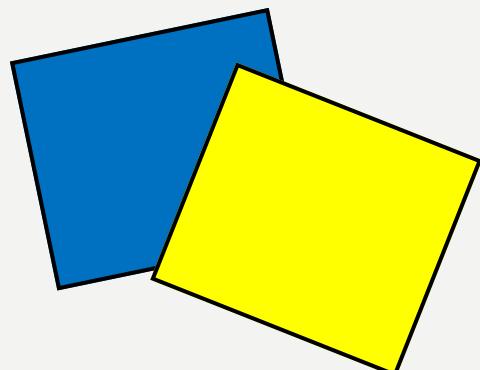


BUSCO: TASKS

```
genome_annot
|__ assembly
|__ augustus
|__ blast
|__ b2go
|__ blat
|__ busco
|__ jbrowse
|__ jobs
|__ logs
|__ repmasker
```

- I. Copy the augustus config folder to YOUR augustus folder

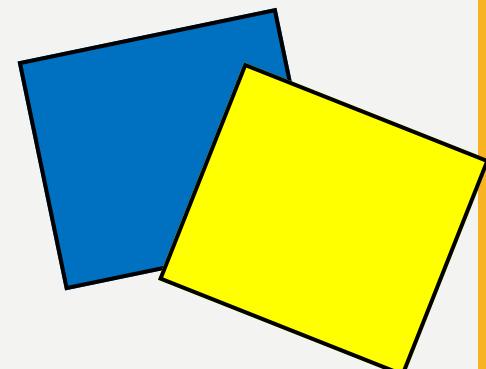
```
cp -r /share/apps/bioinformatics/augustus/conda/3.3.2/config/ .
```



BUSCO: TASKS

1. Download the most appropriate database to your busco folder
(in this case, we will use Aves)
2. Create and submit the BUSCO job:

```
genome_annot
|__ assembly
|__ augustus
|   |__ config
|__ blast
|__ b2go
|__ blat
|__ busco
|__ jbrowse
|__ jobs
|__ logs
|__ repmasker
```



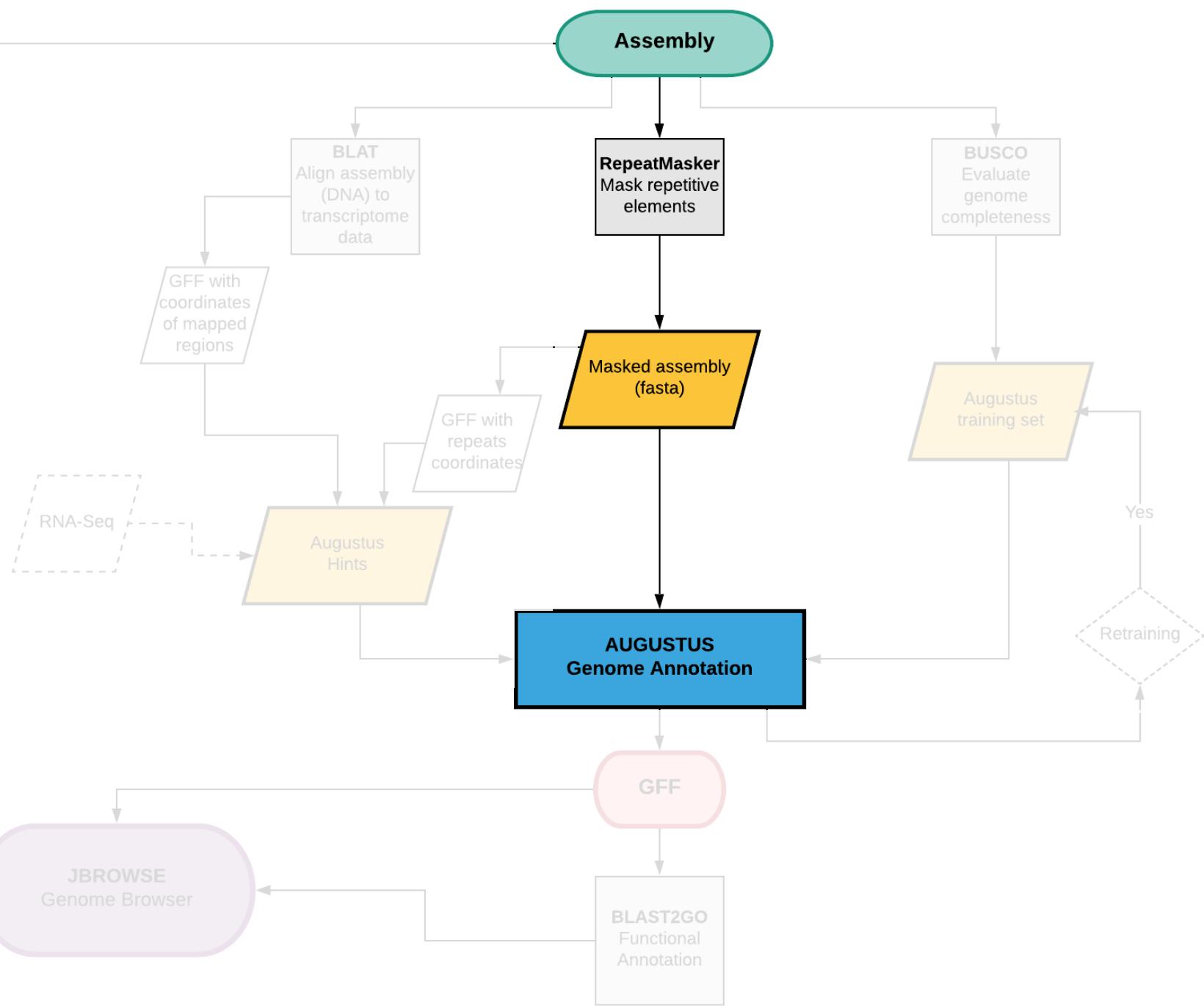
BUSCO

- Tomorrow we will look at the BUSCO results. It will take about ~6 hours for it to run.



MASKING AND ANNOTATION OF REPETITIVE ELEMENTS

REPEATMASKER



REPEATMASKER

- RepeatMasker is a program that screens DNA sequences for interspersed repeats and low complexity DNA sequences.

(from the RepeatMasker website)



>Contig3141_pilon

GACGGTCTGTGCTAGTCAGCACTCATCCACTGTTCCAGCTGGCAGCCCG
CTCGAAGGTGAAAAAAATACTATAGATAGATATGTAGATGTAAAAAGCAGA
TGAGAACCGGTCGCATATATAGACAAGCTCGCGTAACTCTGGAGGAGGAG
AGTGAGCCCTGCGCTCCAGAGATGTCCTTAGAGACTCCGCCGCTCTGATA
CTGTGTCTAGCCATATATAGACAACCAACGAGTCAACCCACTACCGAAAAA
GTTGCGTGGTACTGAACGCTCATTCGCGCTGCCACGCTCTCTCGAGA
ACAGCTCTAGCACCAAATATAAGGACCATCCGCACATCGAGCACCTATGG
AGGCTATCTCACCCCGTGGTGGCGAACACTCAGCGCTCACACACTATGC
GTAAGCCTCTGATTCTACTAGCCAATTGTAACATCAGCCCCAAAACAGCAT
TGTTTGAGAACTTGCCCTCTACTTCTACAGATTATCGCTCAAATCTAAC
TATGCTGAGGTTCAATCTGCCCTCCGCTACTTAGTGGAGACACATGTAA
ATACAGTCGAGGTGCATCACATTAGTAACTCTCGAGAGACTGAGTCTA
CAAGctctctctctctccctctatct ATGCCTATATGCGAGTGCTGAGA
GGACTAGCTAGTGACACGTGCTGGTATCTCTATGCTCTCTCATTGCTA
GATGGACACATCGTGGTGGCAGCGCACGCGAGAGACGCATACTAGCTG
AACGTCTCCATTATTCACTCCCTTACATCAGATAGCATAACATCTAGAAG
TTACTGTGGTAGCACCCATTtctatttatttttttttttttatttttt
ttttttCCTACAGAAGTGCTATCTCAGCACTGAGCGGCGAAAACAGCTA
TTTGCGAGATATTCCATCGAAAATAGACTCCAAAAACATCATCATGAAGA
TCGACCACCTCCCCCTGATATAGGATTTCACAAGTTAAGATGGAAGATGG
CTATTTCAAGGTAGCAGTGGAAAGCTGCTTCTCTCGCCTTGGTAG
CAGCCGACGGCTGGATTGGGCCTCCACATTGCACCCCCAGGTCTTGGCT
GCAAGGACAAGAGAGAGATACTTGGCAAAAAAAATCCACGGTTGGTGTCT
CTGTCCCCCTCTCGGCCTTGACCTGTTATGACTCCTGAGGTATATAT

file name: siskin_10largest.fasta
 sequences: 10
 total length: 41463291 bp (41459480 bp excl N/X-runs)
 GC level: 42.07 %
 bases masked: 1027439 bp (2.48 %)

	number of elements*	length occupied	percentage of sequence
DNA transposons	256	40450 bp	0.10 %
hobo-Activator	41	7588 bp	0.02 %
Tc1-IS630-Pogo	13	2092 bp	0.01 %
En-Spm	0	0 bp	0.00 %
MuDR-IS905	0	0 bp	0.00 %
PiggyBac	0	0 bp	0.00 %
Tourist/Harbinger	31	5322 bp	0.01 %
Other (Mirage, P-element, Transib)	0	0 bp	0.00 %
Rolling-circles	1	108 bp	0.00 %
Unclassified:	100	15272 bp	0.04 %
Total interspersed repeats:		538158 bp	1.30 %

Small RNA:	110	13149 bp	0.03 %
Satellites:	0	0 bp	0.00 %
Simple repeats:	8441	395878 bp	0.95 %
Low complexity:	1650	88173 bp	0.21 %
=====			

* most repeats fragmented by insertions or deletions have been counted as one element

The query species was assumed to be *gallus gallus*
 RepeatMasker Combined Database: Dfam_3.0

run with rmblastn version 2.9.0+

Run information: input file, total length, % bases masked

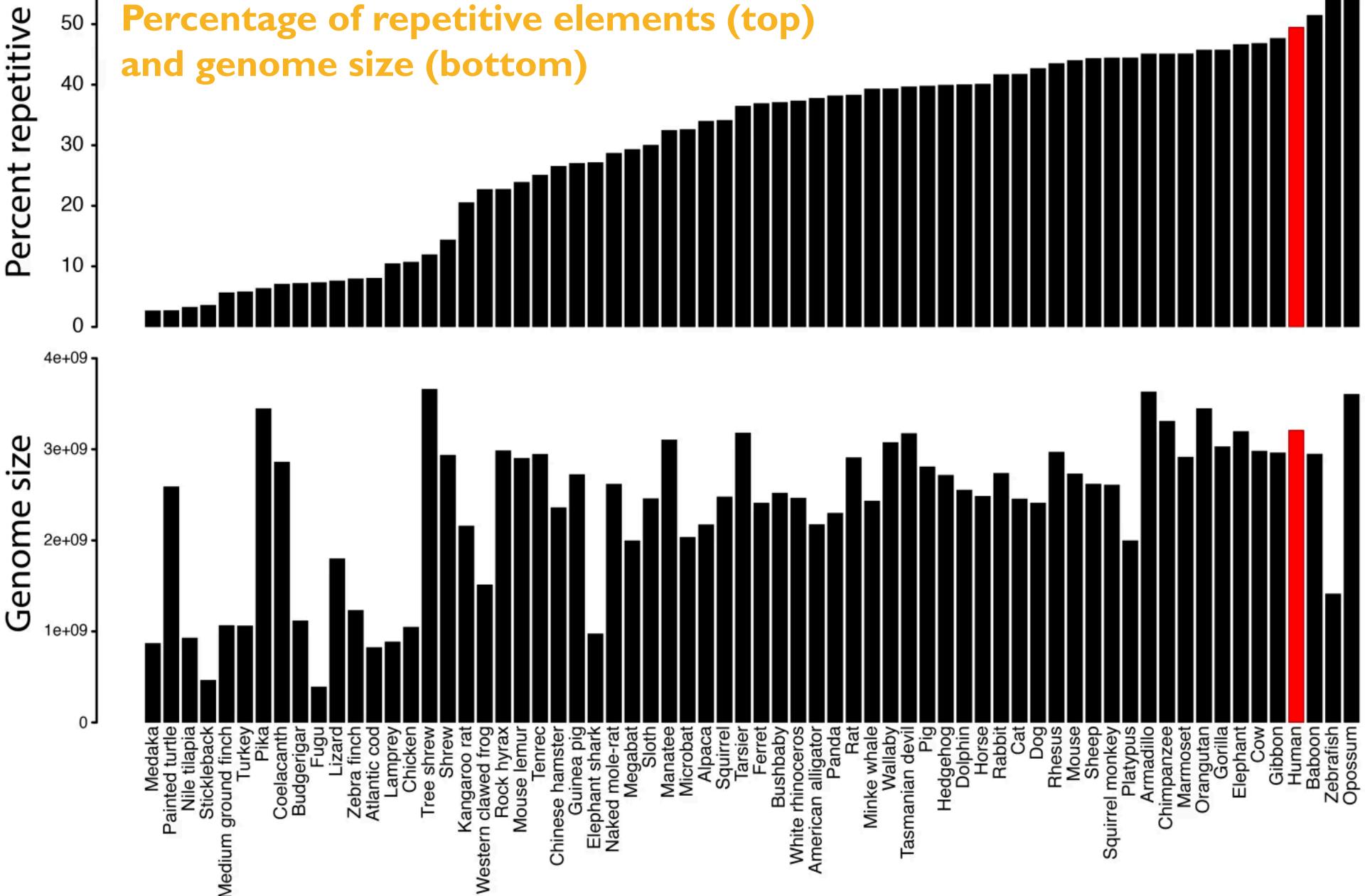
Types of repetitive elements, quantity, length and percentage of sequence

Repeat database and program version

siskin_10largest.fasta.out																
score	SW	query				position in query			matching		repeat class/family	position in repeat			ID	
		perc div.	perc del.	perc ins.	sequence	begin	end	(left)	repeat	begin		begin	end	(left)		
35	5.1	0.0	0.0	Contig1977_pilon	392	432	(3768949)	+	(GCCGCT)n	Simple_repeat	1	41	(0)	1		
24	14.2	3.5	3.5	Contig1977_pilon	6593	6650	(3762731)	+	(CCG)n	Simple_repeat	1	58	(0)	2		
18	13.3	7.0	2.2	Contig1977_pilon	7122	7174	(3762207)	+	(AT)n	Simple_repeat	1	55	(0)	3		
15	5.6	0.0	0.0	Contig1977_pilon	9974	9992	(3759389)	+	(T)n	Simple_repeat	1	19	(0)	4		
15	19.7	0.0	0.0	Contig1977_pilon	20766	20794	(3748587)	+	(T)n	Simple_repeat	1	29	(0)	5		
17	3.8	3.7	0.0	Contig1977_pilon	27637	27663	(3741718)	+	(TTTG)n	Simple_repeat	1	28	(0)	6		
12	12.5	0.0	3.7	Contig1977_pilon	31589	31616	(3737765)	+	(TTGG)n	Simple_repeat	1	27	(0)	7		
230	29.3	5.8	6.9	Contig1977_pilon	32387	32591	(3736790)	+	CR1-3_Croc	LINE/CR1	950	1152	(2453)	8		
236	30.6	11.1	0.0	Contig1977_pilon	32767	32874	(3736507)	+	CR1-L3A_Croc	LINE/CR1	3284	3403	(885)	9		
12	14.7	3.3	0.0	Contig1977_pilon	33646	33675	(3735706)	+	(CTGCTG)n	Simple_repeat	1	31	(0)	10		
13	17.0	0.0	5.7	Contig1977_pilon	40521	40557	(3728824)	+	(CCGCC)n	Simple_repeat	1	35	(0)	11		
14	4.0	7.7	0.0	Contig1977_pilon	42391	42416	(3726965)	+	(CAC)n	Simple_repeat	1	28	(0)	12		
13	18.9	0.0	0.0	Contig1977_pilon	43596	43625	(3725756)	+	(TTA)n	Simple_repeat	1	30	(0)	13		
13	26.2	4.9	1.6	Contig1977_pilon	46810	46870	(3722511)	+	(GCCCG)n	Simple_repeat	1	63	(0)	14		
14	16.3	4.7	2.3	Contig1977_pilon	46897	46939	(3722442)	+	(GGCGG)n	Simple_repeat	1	44	(0)	15		
13	15.9	2.7	2.7	Contig1977_pilon	53355	53391	(3715990)	+	(TTTCTT)n	Simple_repeat	1	37	(0)	16		
264	16.3	1.4	17.2	Contig1977_pilon	58730	58803	(3710578)	C	CR1-3_Croc	LINE/CR1	(1)	3604	3541	17		
24	16.4	0.0	0.0	Contig1977_pilon	83019	83059	(3686322)	+	(T)n	Simple_repeat	1	41	(0)	18		
22	21.9	4.6	1.5	Contig1977_pilon	83423	83487	(3685894)	+	(AT)n	Simple_repeat	1	67	(0)	19		
30	0.0	0.0	0.0	Contig1977_pilon	84441	84468	(3684913)	+	(T)n	Simple_repeat	1	28	(0)	20		
88	0.0	0.0	0.0	Contig1977_pilon	85203	85279	(3684102)	+	(AAT)n	Simple_repeat	1	77	(0)	21		
12	10.3	6.1	2.9	Contig1977_pilon	87513	87545	(3681836)	+	(ACTC)n	Simple_repeat	1	34	(0)	22		
32	23.3	1.5	7.1	Contig1977_pilon	91116	91248	(3678133)	+	(GCCCG)n	Simple_repeat	1	126	(0)	23		
125	0.0	3.1	7.6	Contig1977_pilon	94671	94863	(3674518)	+	(TCCCTT)n	Simple_repeat	1	185	(0)	24		
11	10.6	6.2	8.5	Contig1977_pilon	98745	98792	(3670589)	+	(TCTGATAA)n	Simple_repeat	1	47	(0)	25		
78	21.1	2.8	2.3	Contig1977_pilon	105934	106149	(3663232)	+	(GCAGG)n	Simple_repeat	1	217	(0)	26		
24	13.6	9.2	0.0	Contig1977_pilon	106238	106302	(3663079)	+	(CCCGCGC)n	Simple_repeat	1	71	(0)	27		
21	20.5	0.0	0.0	Contig1977_pilon	110256	110295	(3659086)	+	A-rich	Low_complexity	1	40	(0)	28		
267	22.9	4.3	0.0	Contig1977_pilon	111594	111663	(3657718)	+	CR1-3_Croc	LINE/CR1	2683	2755	(850)	29		
92	0.0	0.0	0.0	Contig1977_pilon	116028	116108	(3653273)	+	(AAAAT)n	Simple_repeat	1	81	(0)	30		
12	11.1	0.0	9.4	Contig1977_pilon	119685	119719	(3649662)	+	(GCTGA)n	Simple_repeat	1	32	(0)	31		
13	28.4	0.0	0.0	Contig1977_pilon	122110	122152	(3647229)	+	A-rich	Low_complexity	1	43	(0)	32		
11	9.7	8.6	2.7	Contig1977_pilon	126315	126349	(3643032)	+	(GGAGGCT)n	Simple_repeat	1	37	(0)	33		
43	2.5	0.0	0.0	Contig1977_pilon	136676	136716	(3632665)	+	(TA)n	Simple_repeat	1	41	(0)	34		
13	10.8	3.1	3.1	Contig1977_pilon	154695	154726	(3614655)	+	(TTGGT)n	Simple_repeat	1	32	(0)	35		
13	0.0	7.7	3.7	Contig1977_pilon	157644	157669	(3611712)	+	(CTAAT)n	Simple_repeat	1	27	(0)	36		
15	5.5	0.0	0.0	Contig1977_pilon	157828	157846	(3611535)	+	(AC)n	Simple_repeat	1	19	(0)	37		
12	5.5	0.0	0.0	Contig1977_pilon	176097	176115	(3593266)	+	(CCA)n	Simple_repeat	1	19	(0)	38		
15	24.4	5.8	1.4	Contig1977_pilon	176667	176735	(3592646)	+	A-rich	Low_complexity	1	72	(0)	39		
13	7.8	7.4	0.0	Contig1977_pilon	177664	177690	(3591691)	+	(CTC)n	Simple_repeat	1	29	(0)	40		
225	11.3	6.5	4.3	Contig1977_pilon	179927	179972	(3589409)	+	CR1-3_Croc	LINE/CR1	1299	1345	(2260)	41		
15	24.1	0.0	2.2	Contig1977_pilon	182585	182630	(3586751)	+	G-rich	Low_complexity	1	45	(0)	42		

Vertebrates:

Percentage of repetitive elements (top)
and genome size (bottom)



REPEATMASKER

- RepeatMasker has several repetitive elements databases (Eukaryotic species only)
- Commonly used species include: mammal, carnivore, rodentia, rat, cow, pig, cat, dog, chicken, fugu, danio, "ciona intestinalis" drosophila, anopheles, elegans, diatoaea, artiodactyl, arabidopsis, rice, wheat, and maize

**HOW TO KNOW WHICH
SPECIES TO USE?**

REPEATMASKER

- To query the RepeatMasker database by taxonomy, you can use the following command:

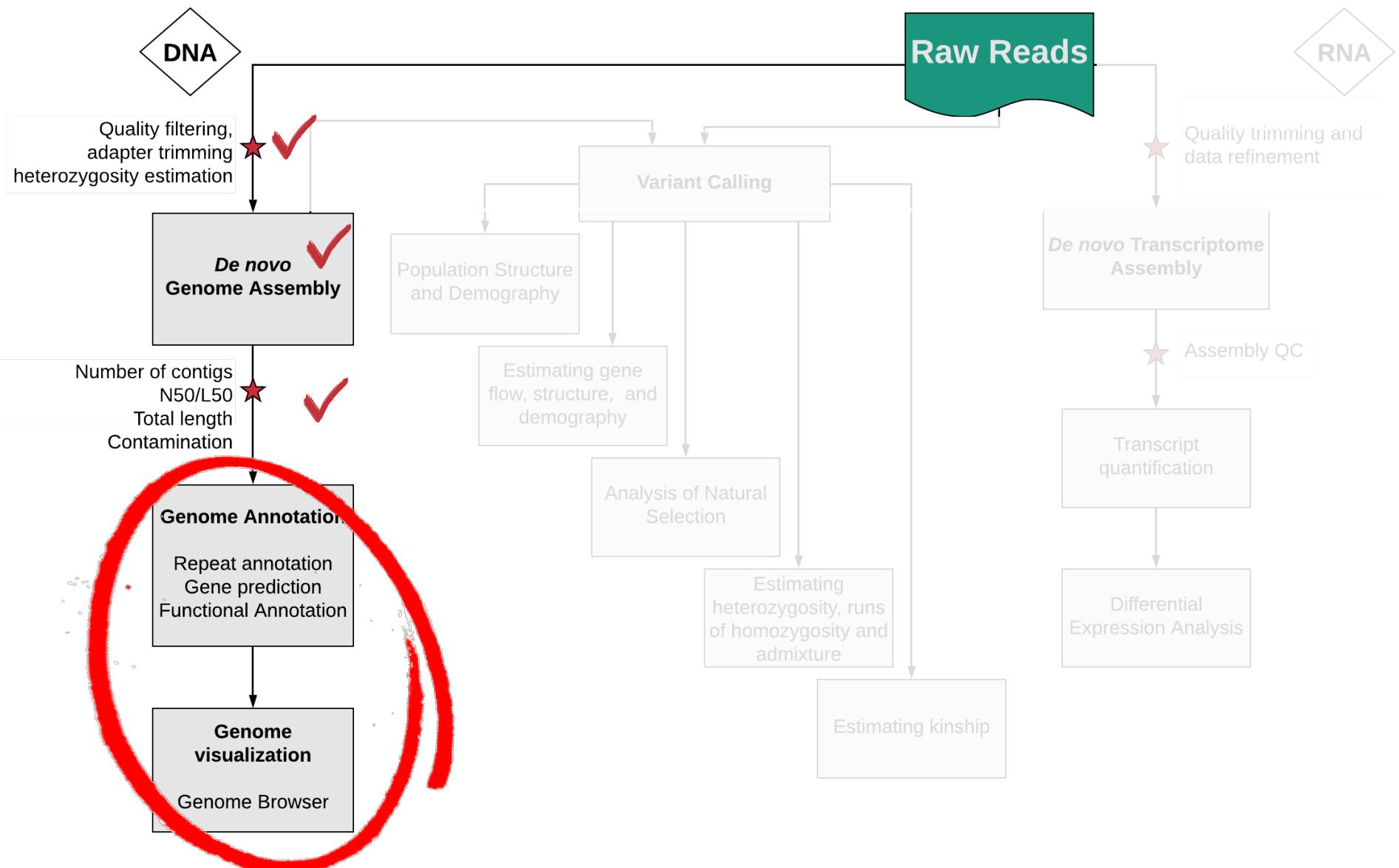
```
queryTaxonomyDatabase.pl -species Spinus
```

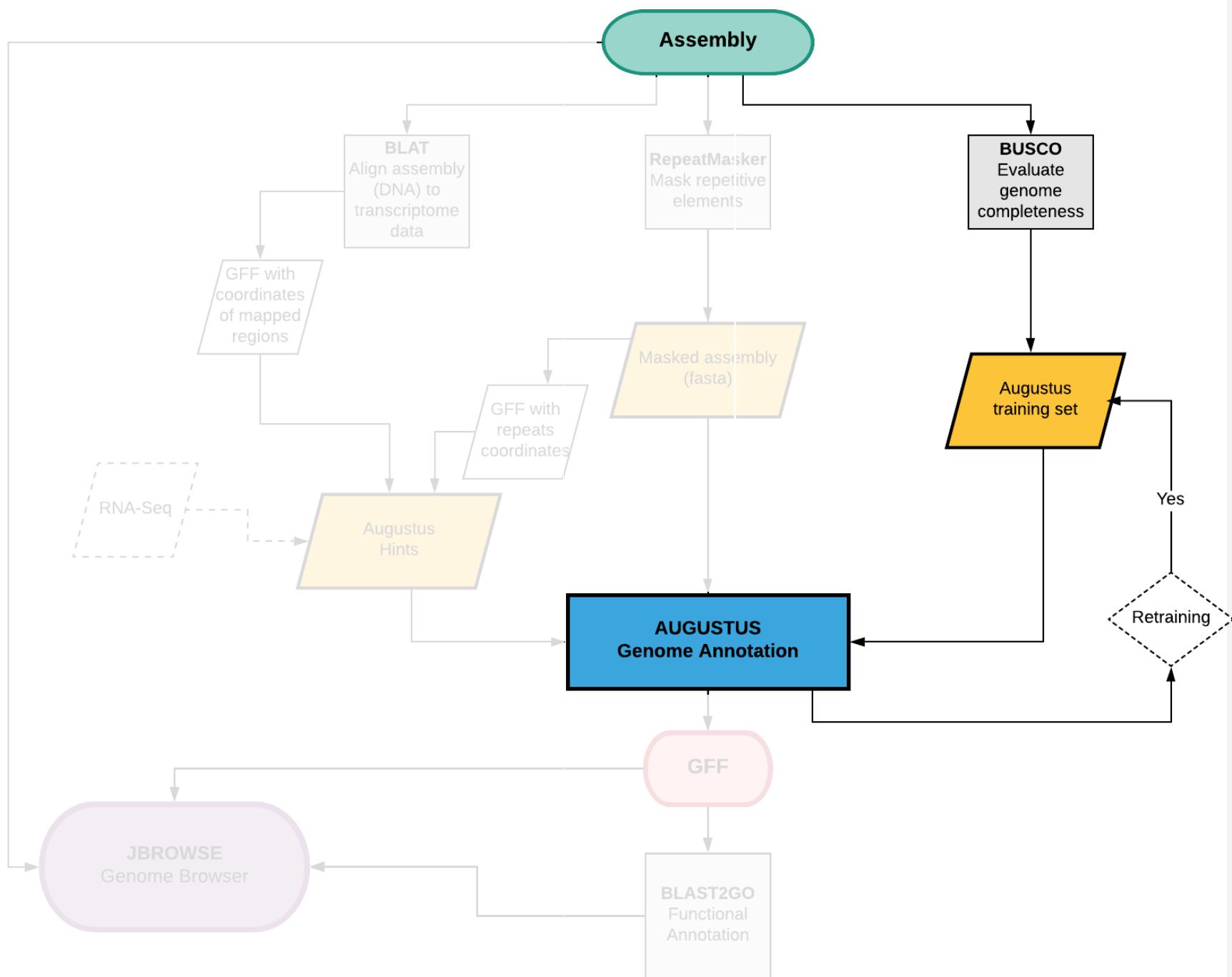
- You can also see the repeat library available for your species

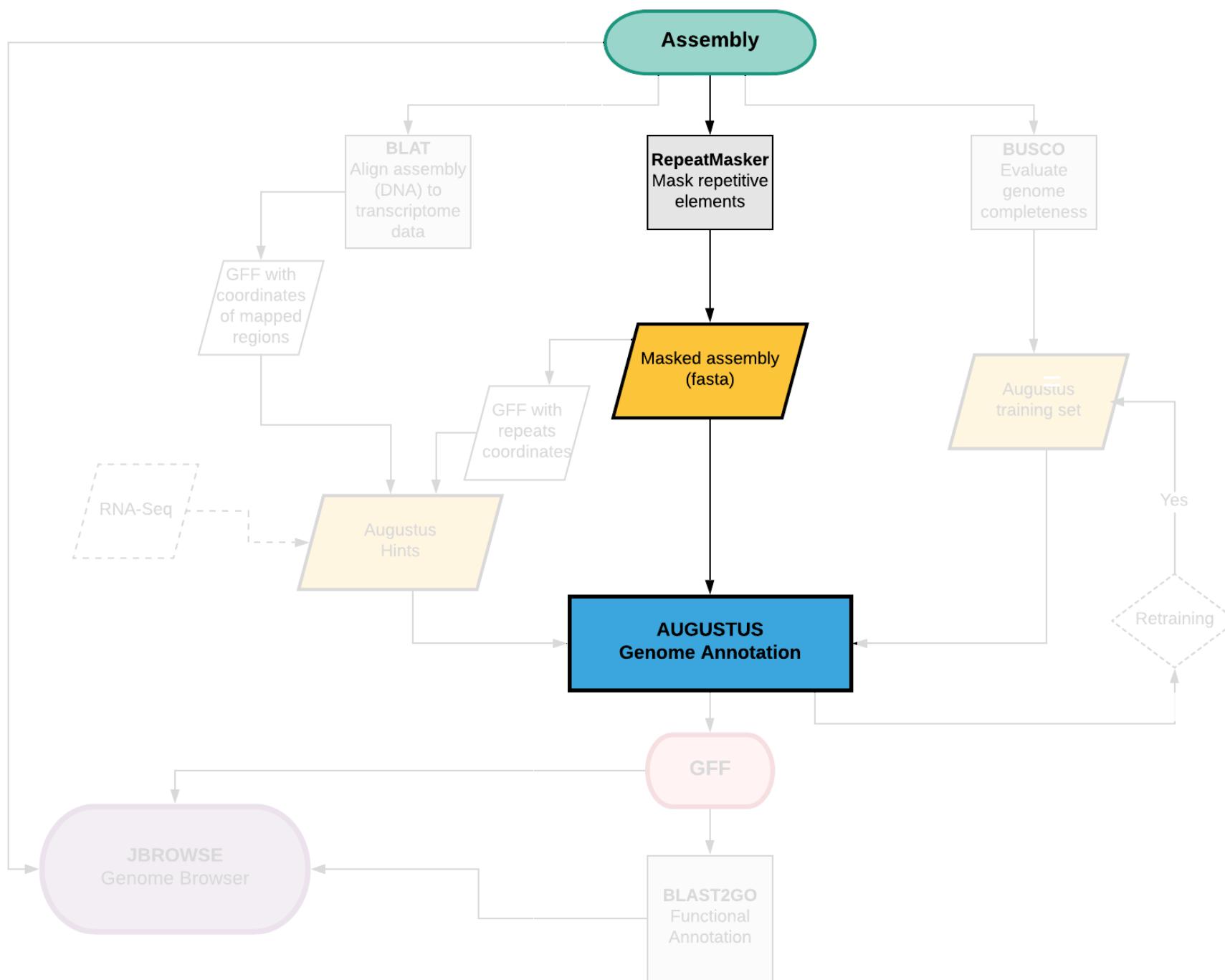
```
queryRepeatDatabase.pl -species Spinus
```

GENOME ANNOTATION

**RECAP –
DAY I**









REPEATMASKER

>Contig3141_pilon
GACGGTCTGTGCTAGTTCAGCACTCATCCACTGTTCCAGCTGGCAGCCCG
CTCGAAGGTGAAAAAAACTATAGATAGATATGTAGATGTAAAAAAGCAGA
TGAGAACCGGTCGCATATATAGACAAGCTCGCGTAACCTGGAGGGAGGAG
AGTGAGCCCTGCGCTCCAGAGATGTCCTTAGAGAGACTCCGCCGCTCTGATA
CTGTGTCTAGCCATATATAGACAACCAACGAGTCAACCCACT
GTTGCGTGGTACTGAACGCTCATTCCCC
ACAGCTCTAGCAACCA
ACCA
REPETITIVE ELEMENTS

MASK REPETITIVE

A
G
T
TA
ATACAGTCGAGGTGCATCACATTAGTA
CA
GGACTAGCTAGTGACACGTGCTGGTATCTCTATGCTCTCTATTGCTA
GATGGACACATCGTGGTGGCAGCGCACGCAGAGAGACGCATACTTAGCTG
AACGTCTCCATTATTCACTCCCTTACATCAGATAGCATAACATCTAGAAG
TTACTGTGGTAGCACCCATTttctatttatTTTTTTTTTTTTTTTTTTTT
tttttCCTACAGAAGTGTATCTTCAGCACTGAGCGGCGAAAACAGCTA
TTTGAGATATTCCATCGAAAAATAGACTCCAAAAACATCATCATGAAGA
TCGACCACCTCCCCCTGATATAGGATTCACAAGTTAAGATGGAAGATGG
CTATTTTCAGGTAGCAGTGGAAAGCTGCTTCTCTCTCGCCCTGGTAG
CAGCCGACGGCTGGATTGGGCCTCCACATTGCACCCCCCAGGTCTGGCT
GCAAGGACAAGAGAGAGATACTTGGAAAAAAATCCACGGTTGGTGTCT
CTGTCCCCCTCTGGCCTTGACCTTGTATGACTCCTGAGGTATATAT

REPEATMASKER

Rephbase_RepeatQuery_Taxonomy_MTNT.sh cat

Let's look at the output files:

- cat_tax.txt
- cat_RepBase_RepQuery.txt

QUESTION

Which is the furthest taxonomic level from **cat** that has the same number of sequences in the repeat database?

TASKS

- Figure out which species/taxon to use for siskin using `Repbase_RepeatQuery_Taxonomy_MTNT.sh`
 - Hint: siskin genus is *Spinus*. Compare to a more common bird species.
- Create the repeatmasker job file in the **jobs** folder following the info from the Github guide.

PARAMETERS

RepeatMasker

-species chicken: RepBase species

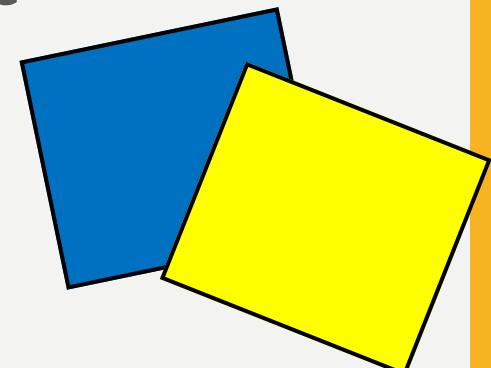
-xsmall : soft-masking (repetitive elements are masked in low caps instead of replaced by N)

-gff: additional output in gff2 format

-pa \$NSLOTS: number of cpus

-dir/repmasker/ : output the results to the specified folder

..../assembly/siskin_10largest.fasta: input file



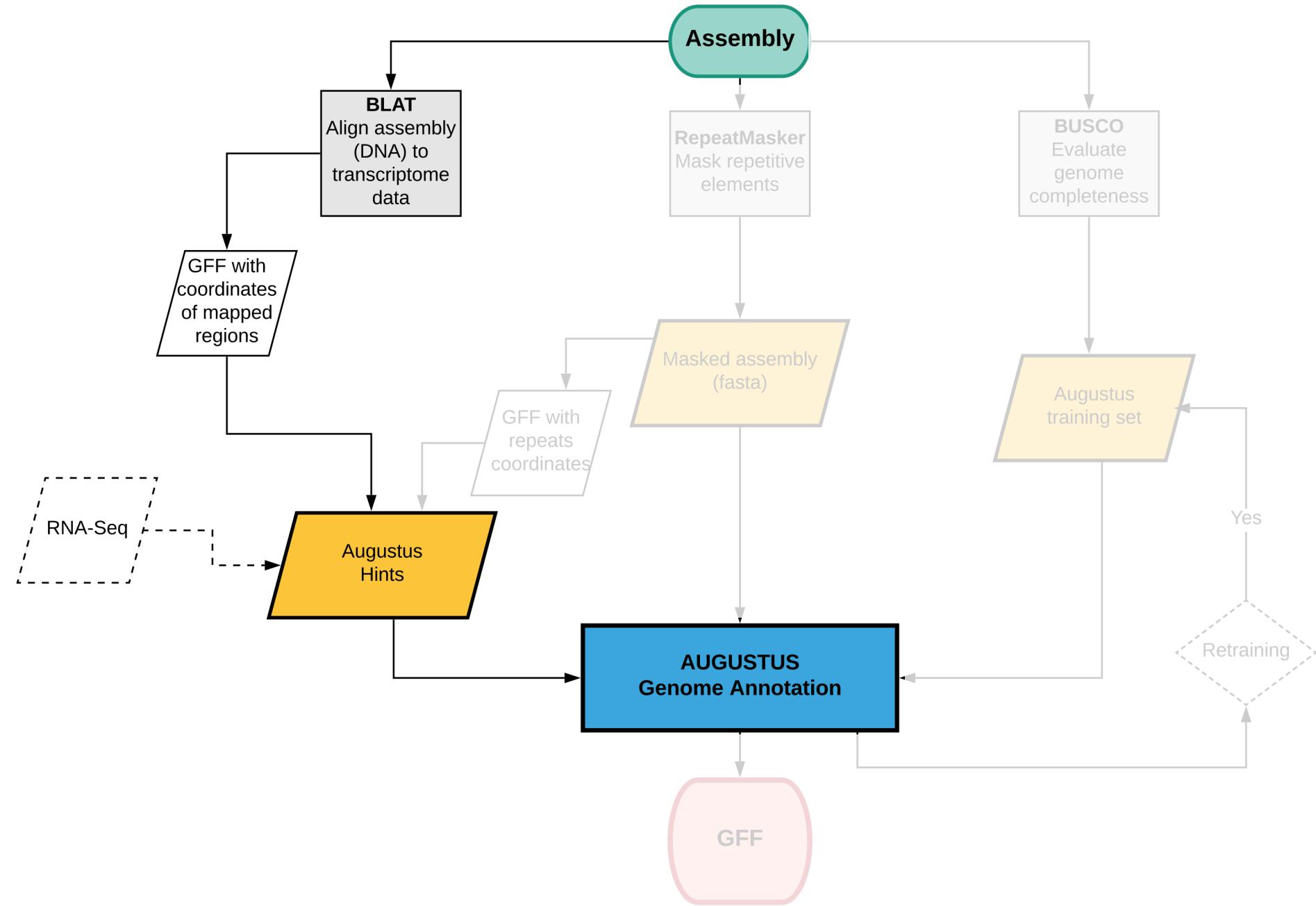
BLAT

BLAST-LIKE ALIGNMENT TOOL

OTHER SOURCES OF EVIDENCE

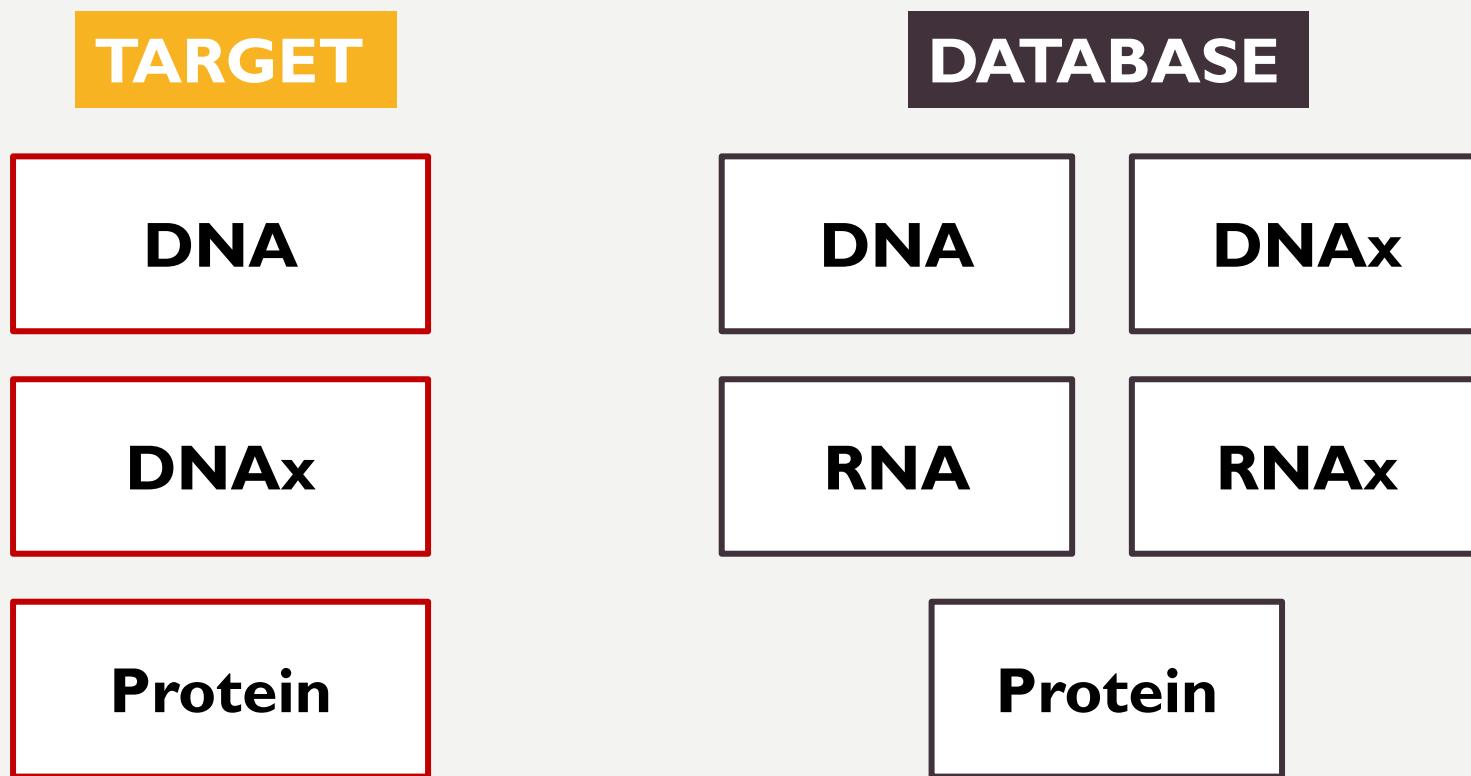
- RNA-Seq
- Transcriptomes

Today we will use the transcriptome of a different species to generate another source of information for the annotation



BLAT

- BLAST-like Alignment Tool



DNAx and RNAx correspond to 6-frame translated sequences

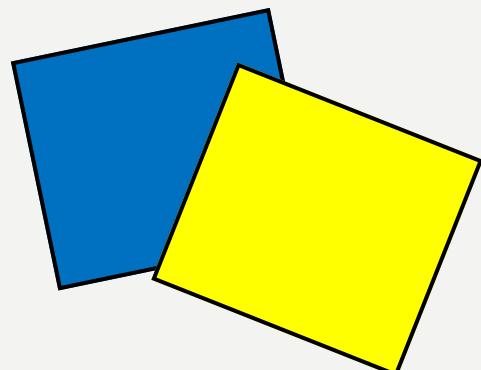
BLAT: WHAT DOES IT TELL US?

- It provides information regarding exons and introns, based on the alignment of the transcriptome sequence to our assembly.

BLAT: TASKS

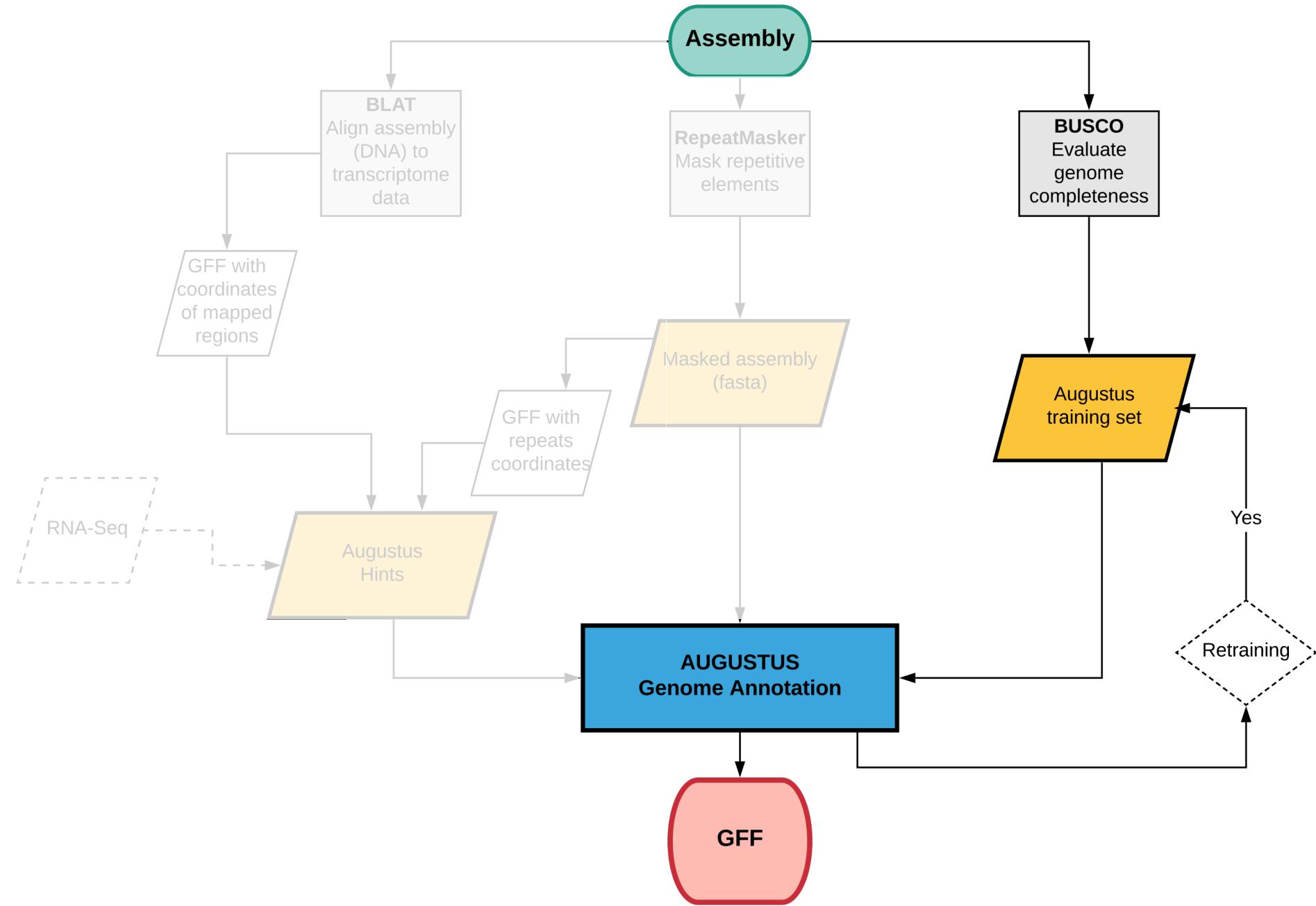
```
genome_annot
|   __ assembly
|   __ augustus
|   __ blast
|   __ b2go
|   __ blat
|   __ busco
|   __ jbrowse
|   __ jobs
|   __ logs
|   __ repmasker
```

1. Download the transcriptome of *Taeniopygia guttata* from Genbank. Extract the file.
2. Create the BLAT job
3. Submit the job





BUSCO - RESULTS



BUSCO OUTPUT: RUN_SISKIN

augustus_output

blast_output

hmmer_output

single_copy_busco_sequences

short_summary_siskin.txt

missing_busco_list_siskin.tsv

full_table_siskin.tsv

BUSCO - RESULTS

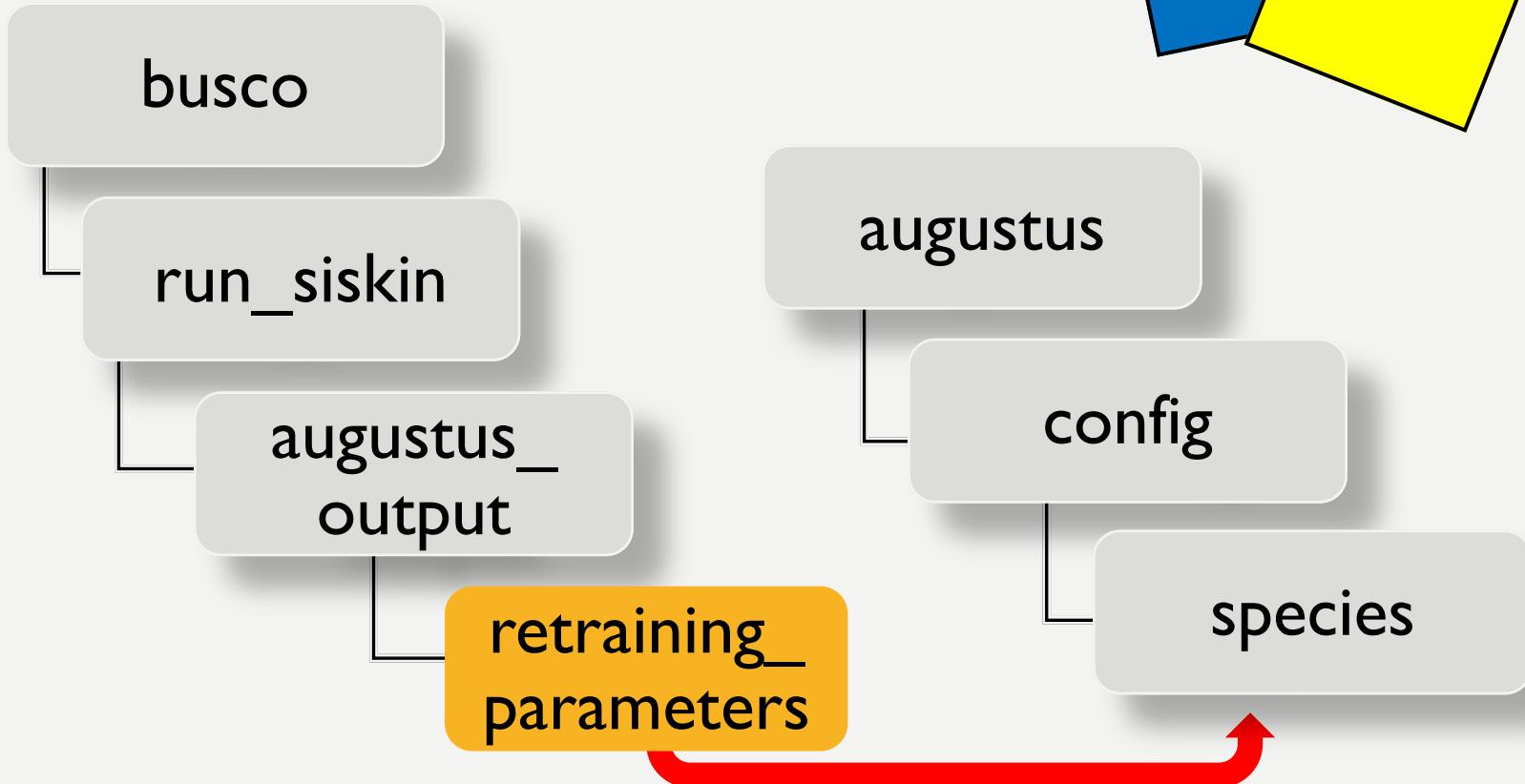
- Check the results of your BUSCO run in the `short_summary_siskin.txt`
 - How many Complete (C), Duplicated (D), Fragmented (F) and Missing (M)?
 - Do you think this is a good or a bad assembly?

BUSCO - RESULTS

- Compare it to the results from the entire assembly:

FROM BUSCO TO AUGUSTUS

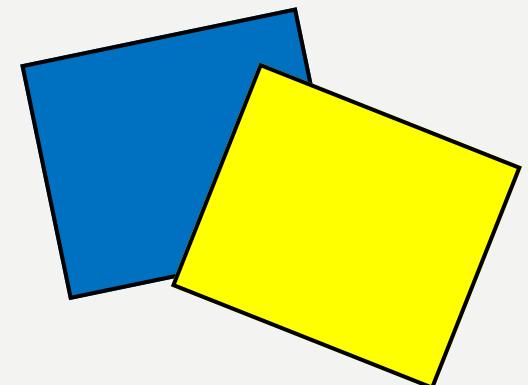
- I. Copy the folder retraining_parameters to augustus/config/species



FROM BUSCO TO AUGUSTUS

2. Rename the folder `retraining_parameters` using the prefix that appears in all files.

You can find this info by looking at the file prefix inside the folder. In this case, we will rename the folder `BUSCO_siskin_3415293029`

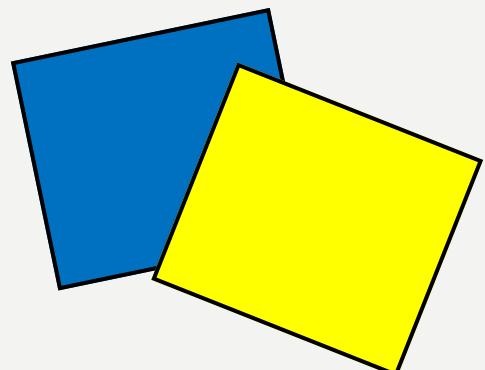




CREATING HINTS FOR AUGUSTUS

AUGUSTUS: FOLDER STRUCTURE

- Create the following folders in your **augustus** directory:
 - hints
 - output
 - scaffolds



AUGUSTUS: FOLDER STRUCTURE

- The command `ls` should return the following result:

config

hints

output

scaffolds

IMPORTANT QUESTIONS

- What sources of extrinsic evidence do we have?
- How do those files look like?

We need to convert the hints into a format
that augustus can read

GFF

- General Feature Format. Fields:
 - **seqname**
 - **source**
 - **feature**
 - **start**
 - **end**
 - **score**
 - **strand**
 - **frame**
 - **attribute**

GFF3 - REPEATMASKER

```
[tsuchiyam@compute-8-31 repmasker]$ cat test.gff3
##gff-version 3
##sequence-region Contig3141_pilon 1 5638391
Contig3141_pilon RepeatMasker dispersed_repeat 604 627 17 + . Target=(CT)n 1 24
Contig3141_pilon RepeatMasker dispersed_repeat 819 856 26 + . Target=(T)n 1 38
Contig3141_pilon RepeatMasker dispersed_repeat 3372 3459 30 + . Target=(TGTT)n 1 89
Contig3141_pilon RepeatMasker dispersed_repeat 3852 3879 12 + . Target=(GCCT)n 1 28
Contig3141_pilon RepeatMasker dispersed_repeat 5845 6049 780 + . Target=CR1-X2 3923 4133
Contig3141_pilon RepeatMasker dispersed_repeat 6253 6290 18 + . Target=(ATT)n 1 38
Contig3141_pilon RepeatMasker dispersed_repeat 17123 17168 14 + . Target=G-rich 1 44
Contig3141_pilon RepeatMasker dispersed_repeat 23515 23627 459 + . Target=CR1-X1 4023 4133
Contig3141_pilon RepeatMasker dispersed_repeat 28192 28600 1488 - . Target=CR1-F2 4088 4497
Contig3141_pilon RepeatMasker dispersed_repeat 28926 28983 254 - . Target=UCON24 206 263
Contig3141_pilon RepeatMasker dispersed_repeat 30324 30384 20 + . Target=A-rich 1 59
Contig3141_pilon RepeatMasker dispersed_repeat 40281 40303 21 + . Target=(T)n 1 23
Contig3141_pilon RepeatMasker dispersed_repeat 42725 43050 969 - . Target=CR1-X2 3783 4098
Contig3141_pilon RepeatMasker dispersed_repeat 44280 44308 15 + . Target=(TCTTC)n 1 28
Contig3141_pilon RepeatMasker dispersed_repeat 44314 44343 32 + . Target=(T)n 1 30
Contig3141_pilon RepeatMasker dispersed_repeat 44531 44573 13 + . Target=(CTGCTG)n 1 46
Contig3141_pilon RepeatMasker dispersed_repeat 47527 47560 13 + . Target=(CCTCCC)n 1 33
Contig3141_pilon RepeatMasker dispersed_repeat 51189 51380 631 + . Target=CR1-H 4602 4798
Contig3141_pilon RepeatMasker dispersed_repeat 52831 52858 19 + . Target=(AACAG)n 1 27
Contig3141_pilon RepeatMasker dispersed_repeat 57134 57152 15 + . Target=(T)n 1 19
Contig3141_pilon RepeatMasker dispersed_repeat 60288 60487 1008 + . Target=CR1-C4 4289 4508
Contig3141_pilon RepeatMasker dispersed_repeat 64723 64739 16 + . Target=(T)n 1 17
Contig3141_pilon RepeatMasker dispersed_repeat 64868 64896 15 + . Target=A-rich 1 29
Contig3141_pilon RepeatMasker dispersed_repeat 65872 65906 12 + . Target=(CTTTA)n 1 34
Contig3141_pilon RepeatMasker dispersed_repeat 72051 72093 47 + . Target=(T)n 1 43
Contig3141_pilon RepeatMasker dispersed_repeat 72913 73192 47 + . Target=(GT)n 1 276
Contig3141_pilon RepeatMasker dispersed_repeat 76071 76299 429 - . Target=GGLTR8B 2 234
Contig3141_pilon RepeatMasker dispersed_repeat 78999 79042 38 + . Target=(A)n 1 44
Contig3141_pilon RepeatMasker dispersed_repeat 80244 80288 28 + . Target=(A)n 1 45
Contig3141_pilon RepeatMasker dispersed_repeat 80590 80640 35 + . Target=(A)n 1 51
Contig3141_pilon RepeatMasker dispersed_repeat 84211 84687 1021 + . Target=CR1-C4 3956 4516
Contig3141_pilon RepeatMasker dispersed_repeat 90620 90684 235 + . Target=Chompy-2_Croc 6 72
```

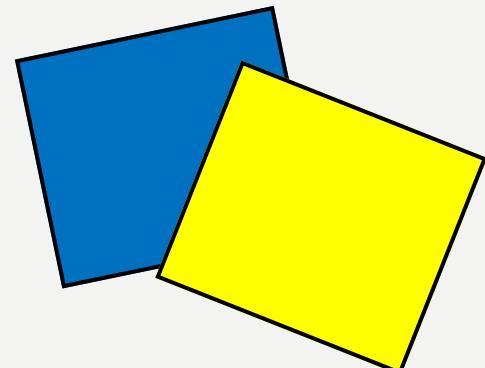
GFF3 - BLAT

mtsuchiya — tsuchiyam@login-30-1:scratch/genomics/tsuchiyam/RepeatMasker/RM_hints — ssh tsuchiyam@hydra-login01.si.edu — 144x44							
Contig3141_pilon	b2h	ep	617	623	0	.	grp=XR_003076074.1;pri=4;src=E
Contig3141_pilon	b2h	ep	617	623	0	.	grp=XR_003076075.1;pri=4;src=E
Contig3141_pilon	b2h	ep	617	623	0	.	grp=XR_003076077.1;pri=4;src=E
Contig3141_pilon	b2h	ep	617	623	0	.	grp=XR_003076078.1;pri=4;src=E
Contig3141_pilon	b2h	ep	3406	3423	0	.	grp=XM_003641786.3;pri=4;src=E
Contig3141_pilon	b2h	ep	3409	3435	0	.	grp=XM_015291998.2;pri=4;src=E
Contig3141_pilon	b2h	ep	3409	3435	0	.	grp=XM_025153839.1;pri=4;src=E
Contig3141_pilon	b2h	ep	3409	3435	0	.	grp=XM_025153840.1;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=NM_001278026.1;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=NM_001278028.1;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_015296983.2;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_015296984.2;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_015296985.2;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_015296986.2;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_015296988.2;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_015296989.2;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_025142718.1;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_025142719.1;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_025142720.1;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_025142721.1;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_025142722.1;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_025142723.1;pri=4;src=E
Contig3141_pilon	b2h	ep	5916	5932	0	.	grp=XM_417523.6;pri=4;src=E
Contig3141_pilon	b2h	intron	28014	28314	0	.	mult=5;pri=4;src=E
Contig3141_pilon	b2h	ep	28316	28334	0	.	grp=XR_001469930.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28316	28334	0	.	grp=XR_003071673.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28325	28336	0	.	grp=XR_001464165.2;pri=4;src=E
Contig3141_pilon	b2h	ep	28325	28340	0	.	grp=XR_003071393.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28325	28340	0	.	grp=XR_003071394.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28325	28341	0	.	grp=XR_003074136.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28325	28345	0	.	grp=XR_003072657.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28325	28348	0	.	grp=XR_003074120.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28335	28348	0	.	grp=XR_003073844.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28328	28349	0	.	grp=XR_003073482.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28328	28349	0	.	grp=XR_003073483.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28328	28349	0	.	grp=XR_003073484.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28315	28350	0	.	grp=XR_003074104.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28315	28350	0	.	grp=XR_003074105.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28315	28350	0	.	grp=XR_003074106.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28315	28350	0	.	grp=XR_003074107.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28315	28350	0	.	grp=XR_003074108.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28325	28350	0	.	grp=XM_015286437.2;pri=4;src=E
Contig3141_pilon	b2h	ep	28325	28350	0	.	grp=XM_025148848.1;pri=4;src=E
Contig3141_pilon	b2h	ep	28325	28354	0	.	grp=XM_025149296.1;pri=4;src=E

AUGUSTUS HINTS: BLAT

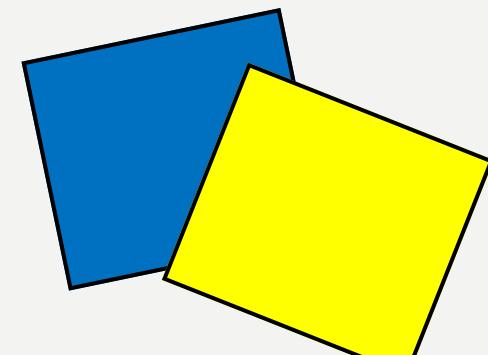
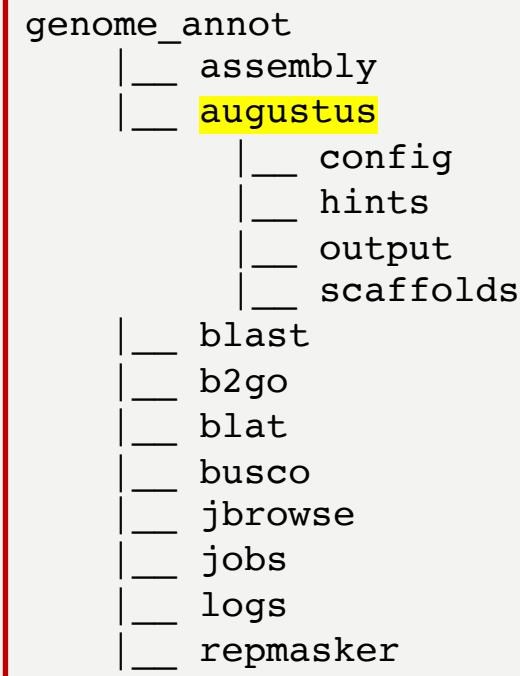
1. Log to the interactive queue
2. Sort the .psl file
3. Load the augustus/3.3 module
4. Run the script blat2hints.pl

```
genome_annot
|__ assembly
|__ augustus
|   |__ config
|   |__ hints
|   |__ output
|   |__ scaffolds
|__ blast
|__ b2go
|__ blat
|__ busco
|__ jbrowse
|__ jobs
|__ logs
|__ repmasker
```



AUGUSTUS HINTS: REPEATMASKER

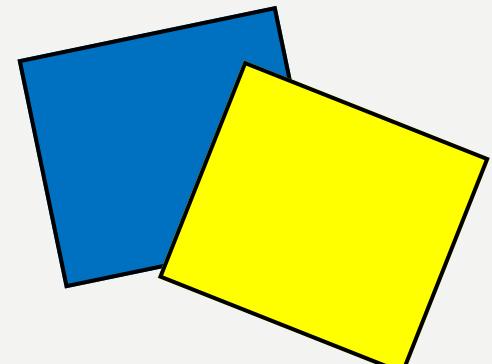
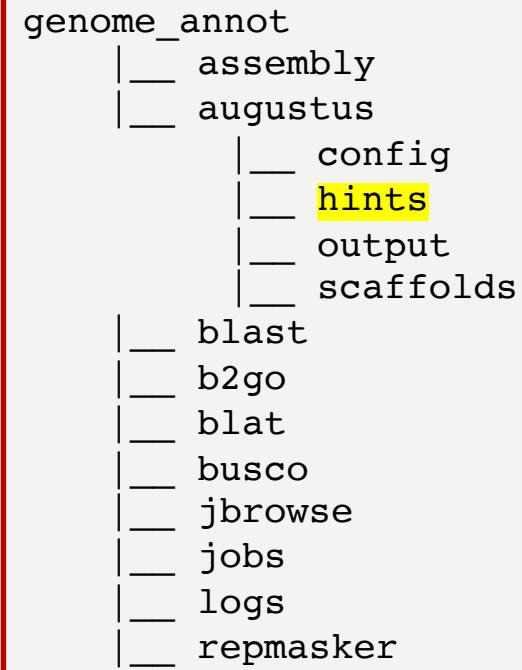
1. Load the module repeatmasker
2. Use the script rmOutToGFF3.pl to convert your .out file into GFF3
3. Use the script gff2hints to make the final conversion

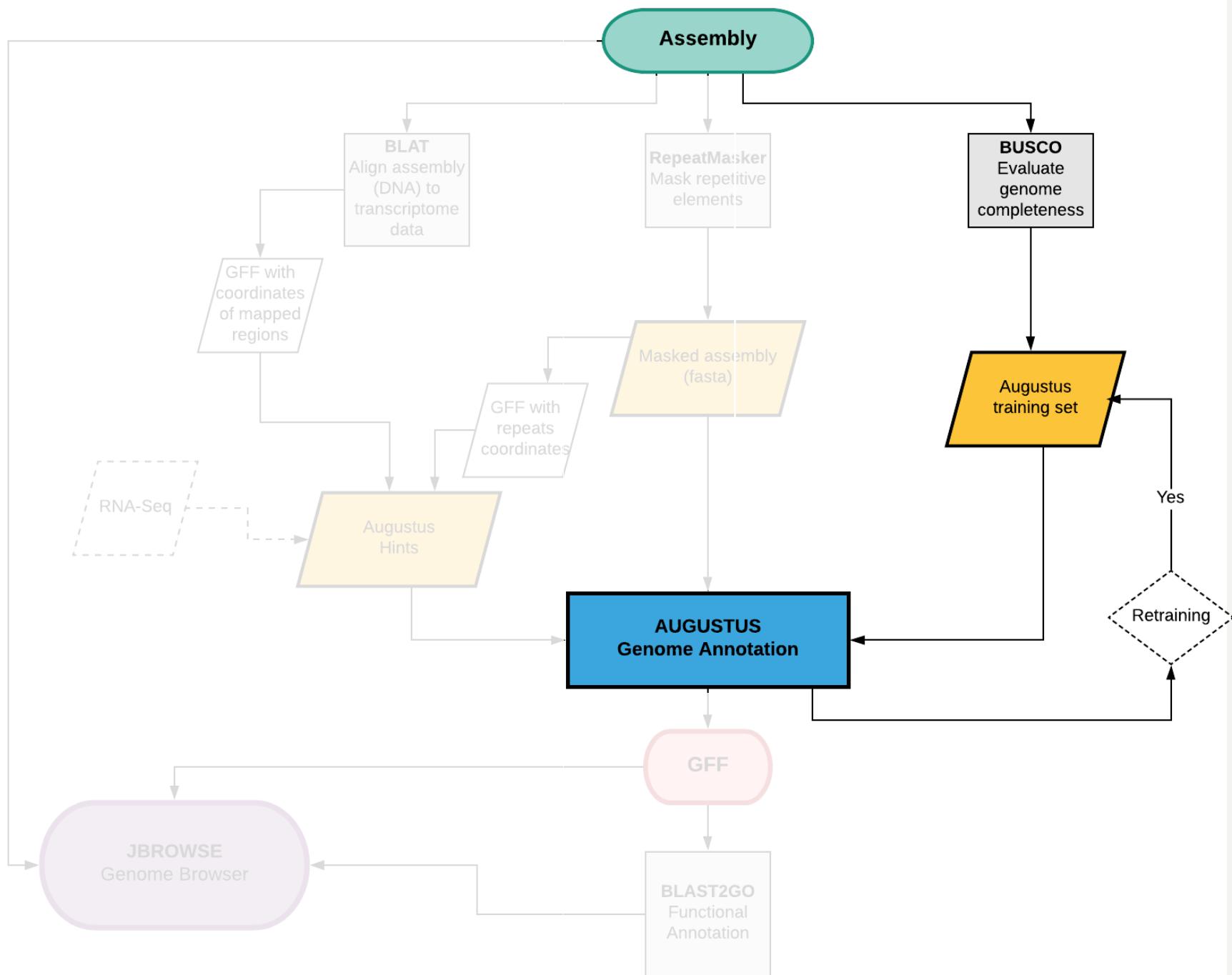


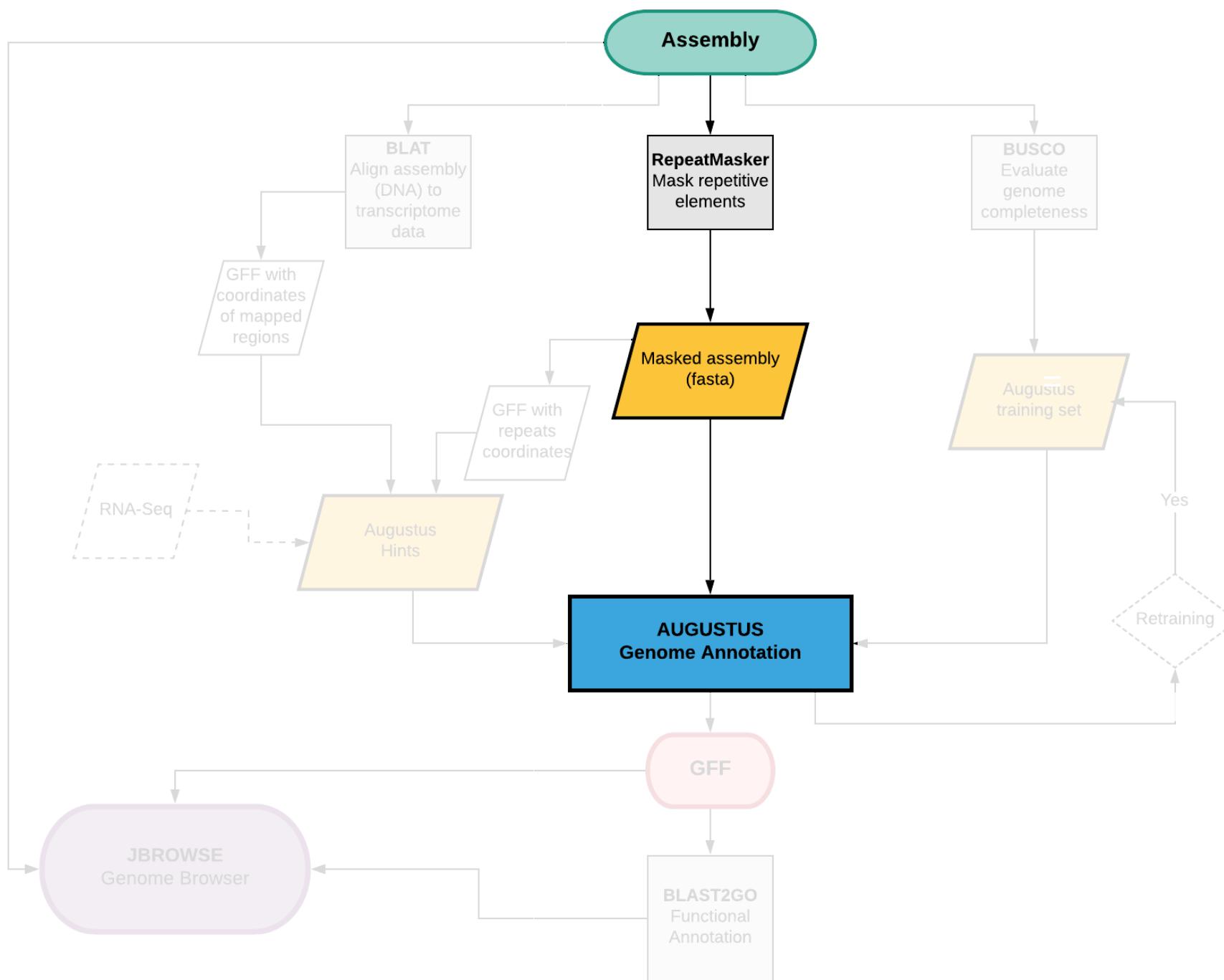
AUGUSTUS: COMBINING HINTS

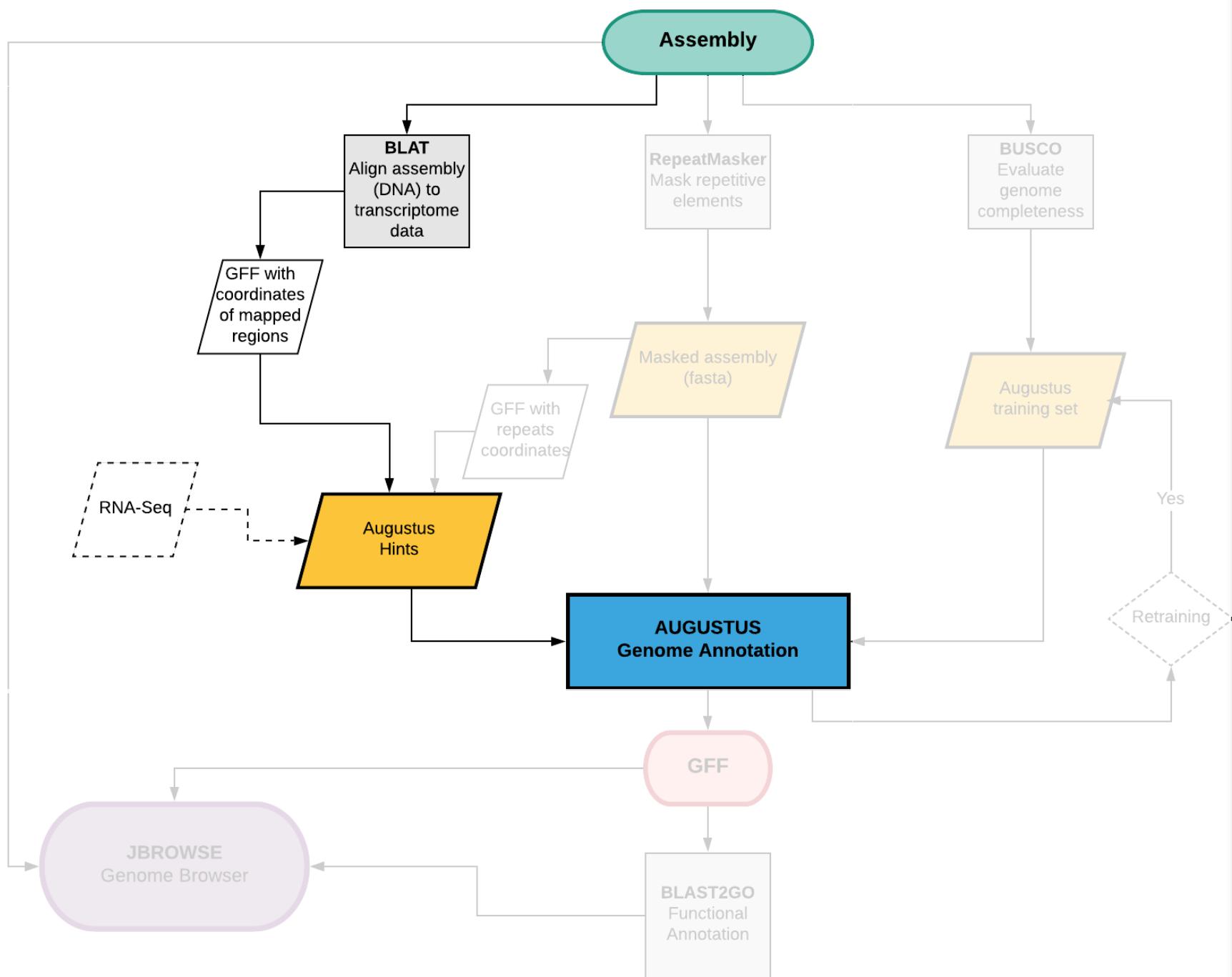
- Merge both files:

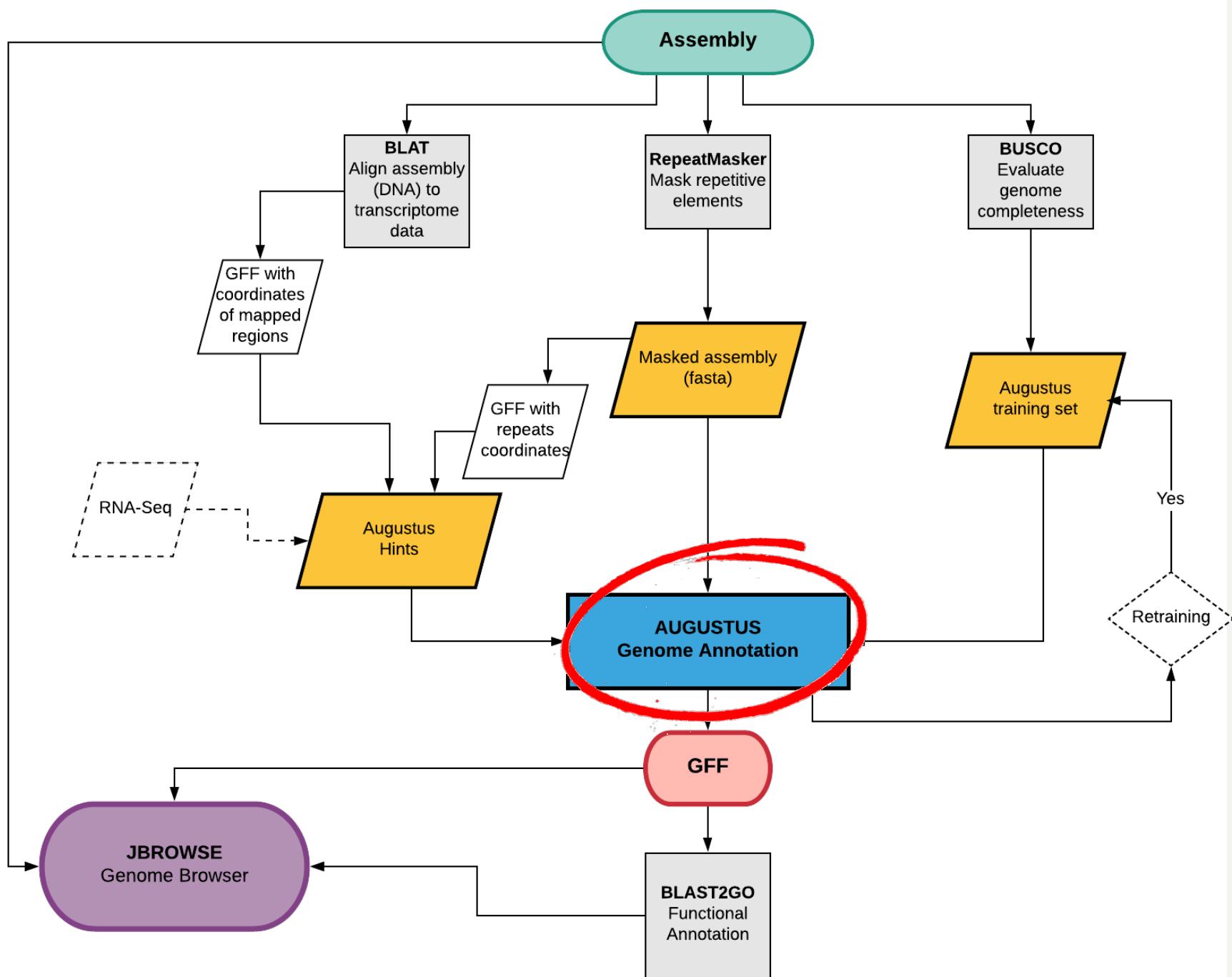
```
cat siskin_RM_hints.out  
siskin_blat_hints.out | sort -k1,1V -  
k4,4n > siskin_hints_RM_E.gff3
```













AUGUSTUS

- ab initio (internal) + evidence-driven(external)

AUGUSTUS is based on a generalized hidden Markov model (GHMM), which defines probability distributions for the various sections of genomic sequences. Introns, exons, intergenic regions, etc. correspond to states in the model and each state is thought to create DNA sequences with certain pre-defined emission probabilities.

AUGUSTUS

- Augustus needs to be trained:
 - Training consists on the generation of a training set that will more accurately predict genes.

(BUSCO solved the training issue for us)



AUGUSTUS

- Inputs:
 - Masked fasta (repeatmasker output)
 - Hints file (from repeatmasker and blat)
 - Training set (from BUSCO)

AUGUSTUS EXTRINSIC FILE

- Defines how the information from the hints will be weighted:
- It assigns “bonus” and “malus” (penalty) values to each hint used
 - M: manual annotation
 - W: RNA-Seq coverage information
 - E: EST/cDNA database hit
 - R: retroposed genes
 - RM: repeat masking

[SOURCES]
M RM E

```
#  
# individual_liability: Only unsatisfiable hints are disregarded. By default this flag is not set  
# and the whole hint group is disregarded when one hint in it is unsatisfiable.  
# 1group1gene: Try to predict a single gene that covers all hints of a given group. This is relevant for  
# hint groups with gaps, e.g. when two ESTs, say 5' and 3', from the same clone align nearby.  
#
```

[SOURCE-PARAMETERS]

```
# feature bonus malus gradelevelcolumns  
# r+/r-  
#  
# the gradelevel colums have the following format for each source  
# sourcecharacter numscoreclasses boundary ... boundary gradequot ... gradequot  
#
```

[GENERAL]

		malus	bonus	bonus	bonus	
start	1	1	M	1	1e+100	RM
stop	1	1	M	1	1e+100	RM
tss	1	1	M	1	1e+100	RM
tts	1	1	M	1	1e+100	RM
ass	1	1	0.1	M	1	1e+100
dss	1	1	0.1	M	1	1e+100
exonpart	1	.992	.985	M	1	1e+100
exon	1		1	M	1	1e+100
intronpart	1		1	M	1	1e+100
intron	1		.34	M	1	1e+100
CDSpart	1	1	.985	M	1	1e+100
CDS	1		1	M	1	1e+100
UTRpart	1	1	.985	M	1	1e+100
UTR	1		1	M	1	1e+100
irpart	1		1	M	1	1e+100
nonexonpart	1		1	M	1	1e+100
genicpart	1		1	M	1	1e+100

```
#  
# Explanation: see original extrinsic.cfg file
```

[SOURCES]
M RM E W P

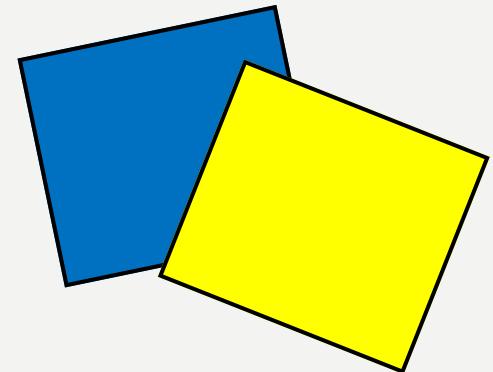
[GENERAL]

start	1	0.8	M 1	1e+100	RM	1 1	E 1	1	W 1	1	P 1	1e3
stop	1	0.8	M 1	1e+100	RM	1 1	E 1	1	W 1	1	P 1	1e3
exonpart	1	.992 .985	M 1	1e+100	RM	1 1	E 1	1	W 1	1.02	P 1	1
exon	1	0.9	M 1	1e+100	RM	1 1	E 1	1	W 1	1	P 1	1e4
intrонpart	1	1	M 1	1e+100	RM	1 1	E 1	1	W 1	1	P 1	1
intrон	1	.34	M 1	1e+100	RM	1 1	E 1 1e6		W 1	1	P 1	100
CDSpart	1	1 .985	M 1	1e+100	RM	1 1	E 1	1	W 1	1	P 1	1e5
CDS	1	1	M 1	1e+100	RM	1 1	E 1	1	W 1	1	P 1	1
nonexonpart	1	1	M 1	1e+100	RM 1 1.15	E 1	1	W 1	1	P 1	1	

Figure 5 An excerpt of an extrinsic configuration file. In this example, each number to the right of the column filled with M's that is different from 1 specifies a *bonus*. A bonus is a relative factor that the un-normalized joint probability of gene structure candidate gets for being compatible with a hint of that type and source. For example, the blue $1e6$ in the intron row after the source letter E means that for each intron hint with source tag E (src=E), gene structures that have an intron with both boundaries given as in the hint are rewarded by a factor of 10^6 relatively to gene structures disregarding the intron hint. A high bonus has the effect that many of the respective hints are respected by AUGUSTUS. The green 1.15 in the non-exonpart row after the tag RM (repeat masking) specifies that for each non exonpart hint, every gene structure gets a relative bonus factor of 1.15 *for each base* that is not an exon and not in a repeat. This discourages—but does not exclude—the overlap of exons and repeats. Repeat masking evidence can be given explicitly with hints of source RM, or implicitly with a soft-masked genome and the option *softmasking* turned on. The number(s) immediately to the left of the M column other than 1 specifies a penalty (malus) for gene structures with unsupported features. For example, the red .34 in the intron row means that every intron candidate that has no intron hints supporting it is penalized by multiplying its unnormalized probability with the factor 0.34. If you decrease this number even more (say from .3 to .001) then fewer introns unsupported by hints should be predicted. This would likely decrease the false positive intron rate, but, also, more true unsupported introns would be missed. For more information, see the file *config/extrinsic/extrinsic.cfg*.

TASKS

- I. Download and extract EVidenceModeler to your augustus folder



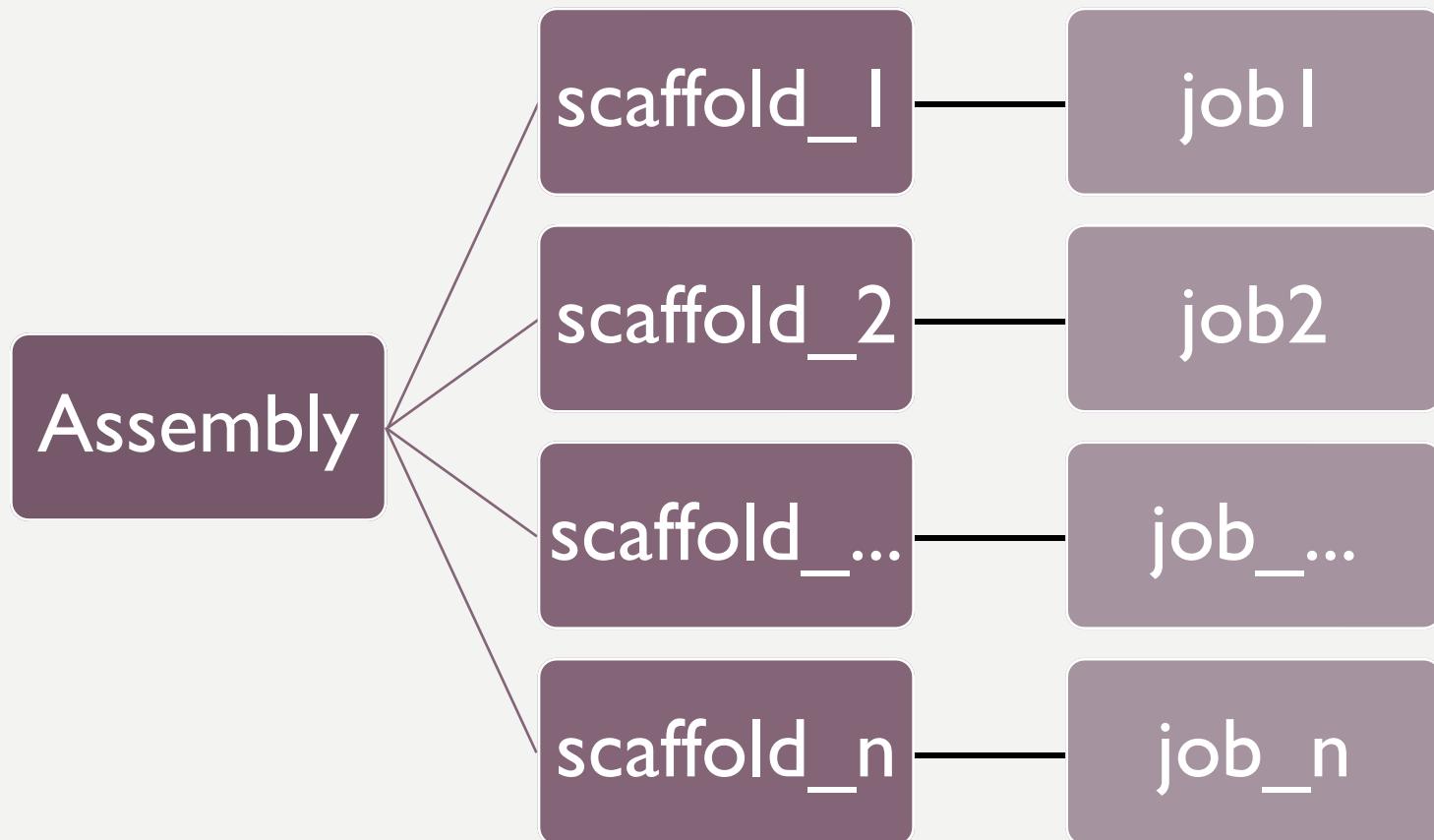
EMBARRASSINGLY PARALLEL

- AUGUSTUS runs serially (aka one scaffold at a time)



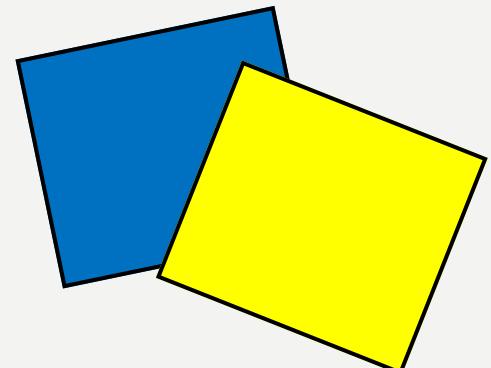
EMBARRASSINGLY PARALLEL

- But we can “force” it to run in parallel



TASKS

- Login to the interactive queue
- Run the EVM script `partition_EVM_inputs.pl` from your scaffolds folder



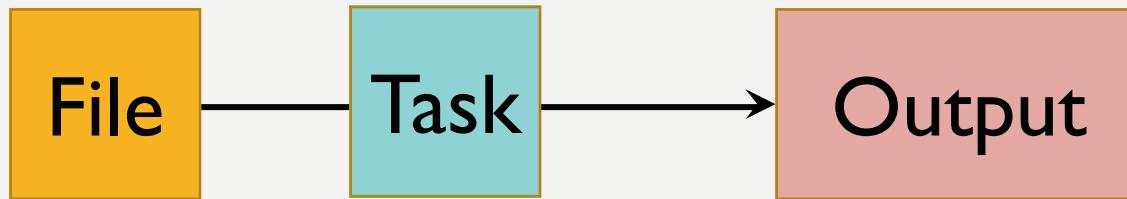
SCAFFOLDS FOLDER

- How many folders do we have?
- Take a look in the folder Contig1977_pilon.
 - What's inside?
 - Use less to look at the files.

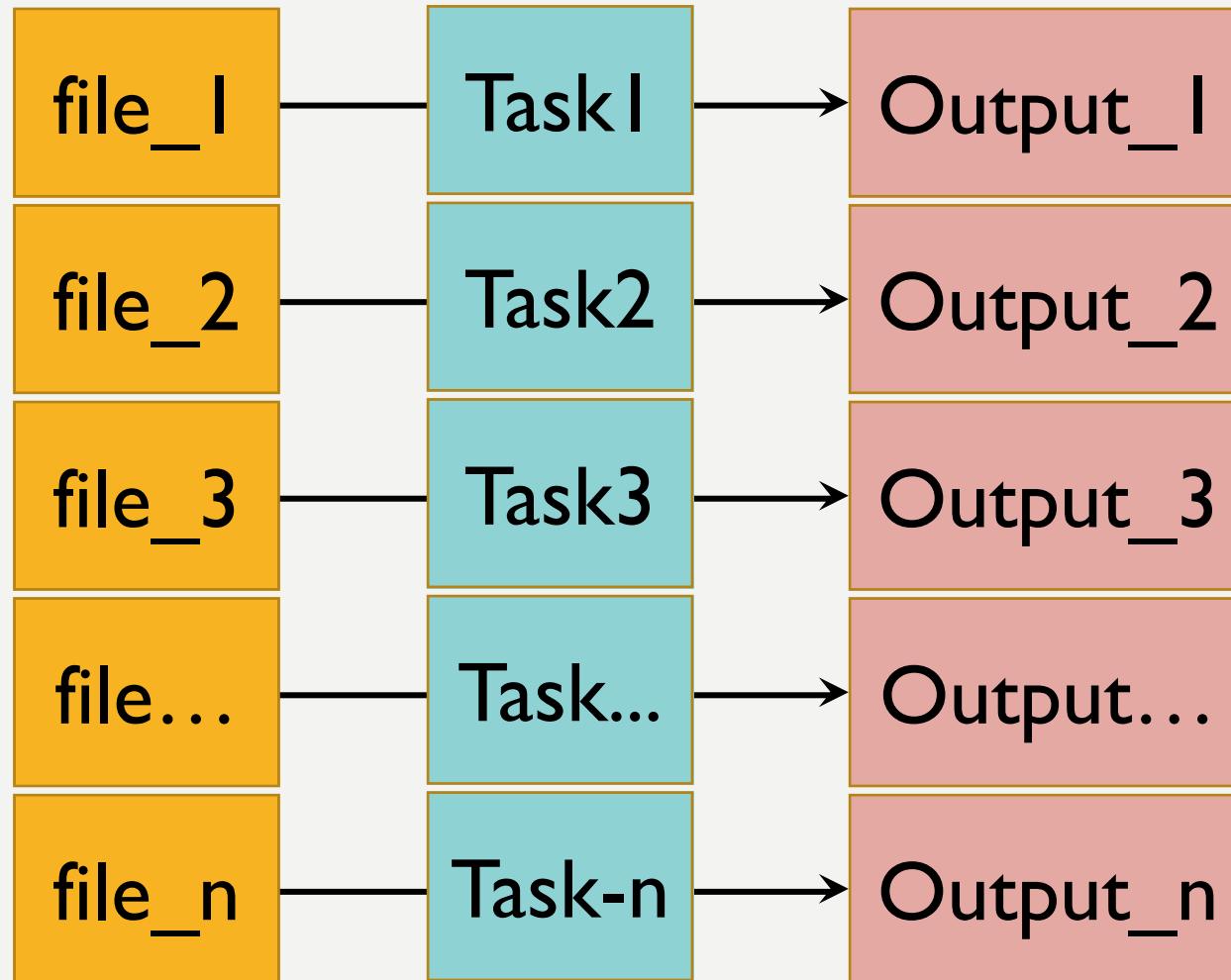
AUGUSTUS JOB FILE

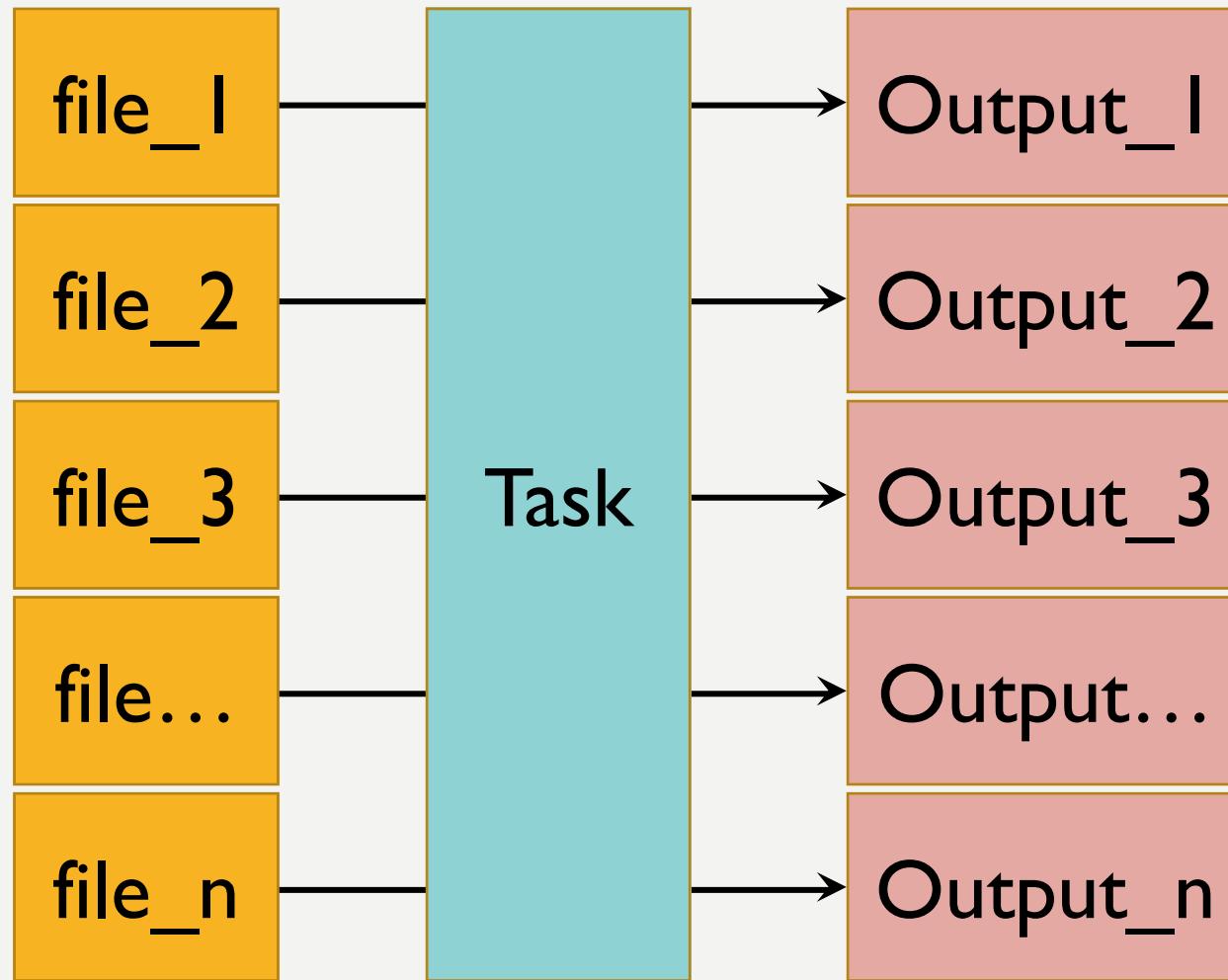
- How are we running augustus on each scaffold?
 - Option 1: Create one job file for each scaffold... manually
 - Option 2: Create one job file and use a loop to submit it.

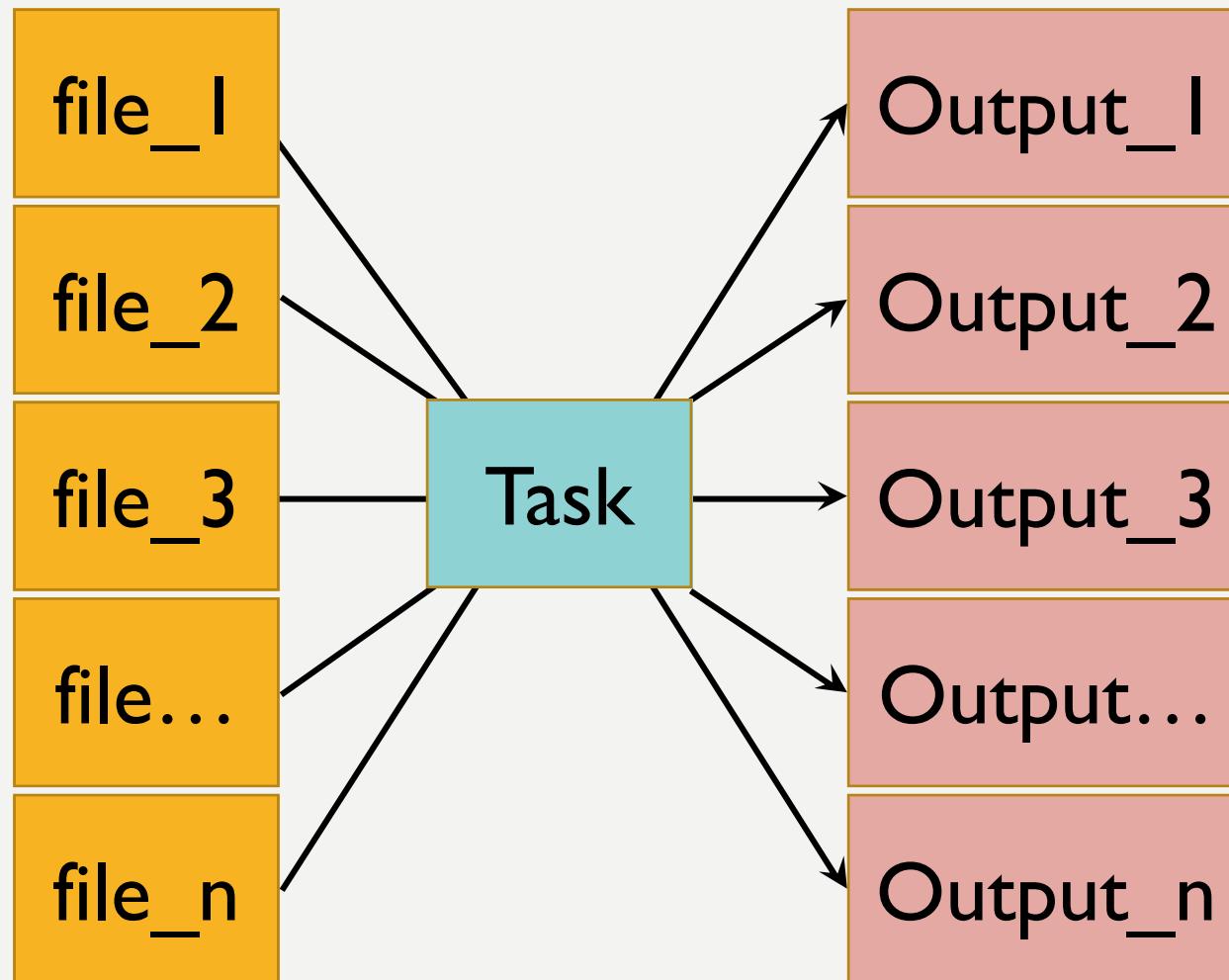
BRIEF INTRO ABOUT LOOPS



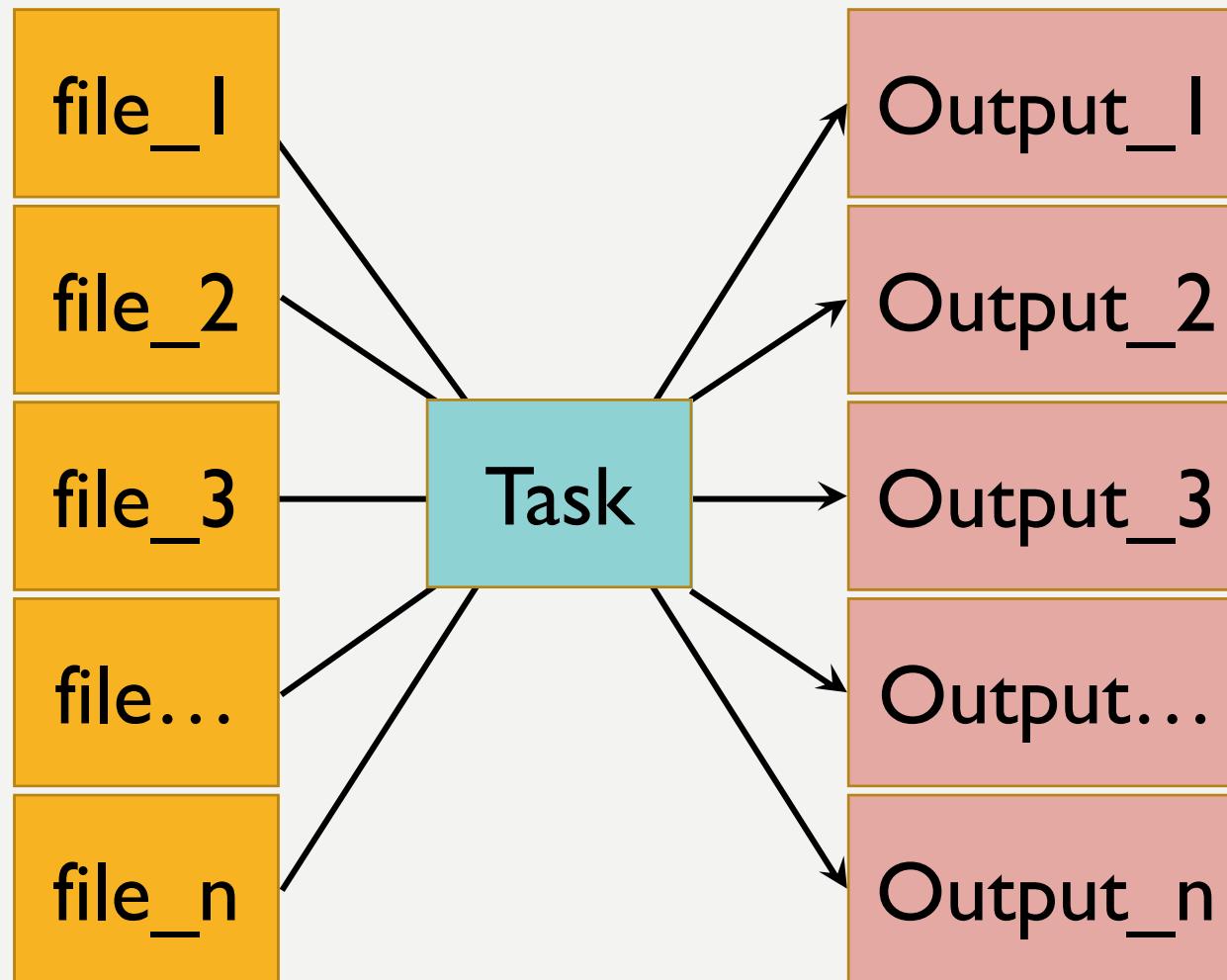
One file = One job







```
for f in file_*; do task > ${f}.output; done
```





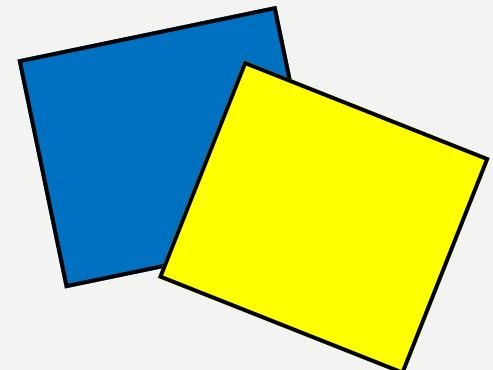
FOR DOG IN PUPPY_*; DO
PLAY \${DOG} > \${DOG}.HAPPY;
DONE

BACK TO AUGUSTUS

- We will create one job (`augustus.job`) and will submit it using a `for` loop that will iterate over each scaffold - all at the same time.

TASKS

- Create the augustus job and save it in the jobs folder.
- You will submit the job from the folder scaffolds.

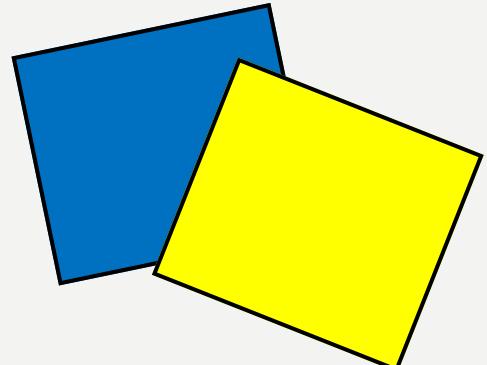


AUGUSTUS RESULTS

- List the files in the folder outputs. You should have one gff file per scaffold.
- Use cat or less to visualize file contents.

COMBINING THE RESULTS

- Now we want to combine all the gffs into a single gff file. `siskin_augustus_all.gff`
- We will use the script `'join_aug_pred.pl'` from AUGUSTUS to combine the results.

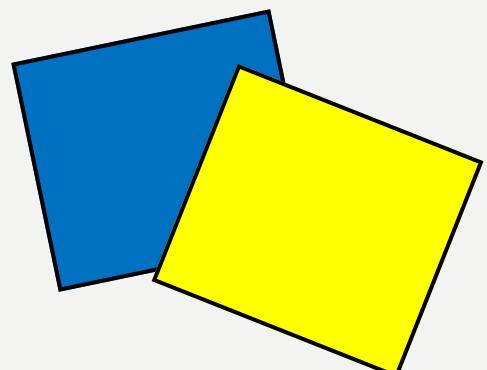


GETTING FASTA SEQUENCES

- Extract each gene model as protein (aa) and nucleotide, as well as each CDS individually:

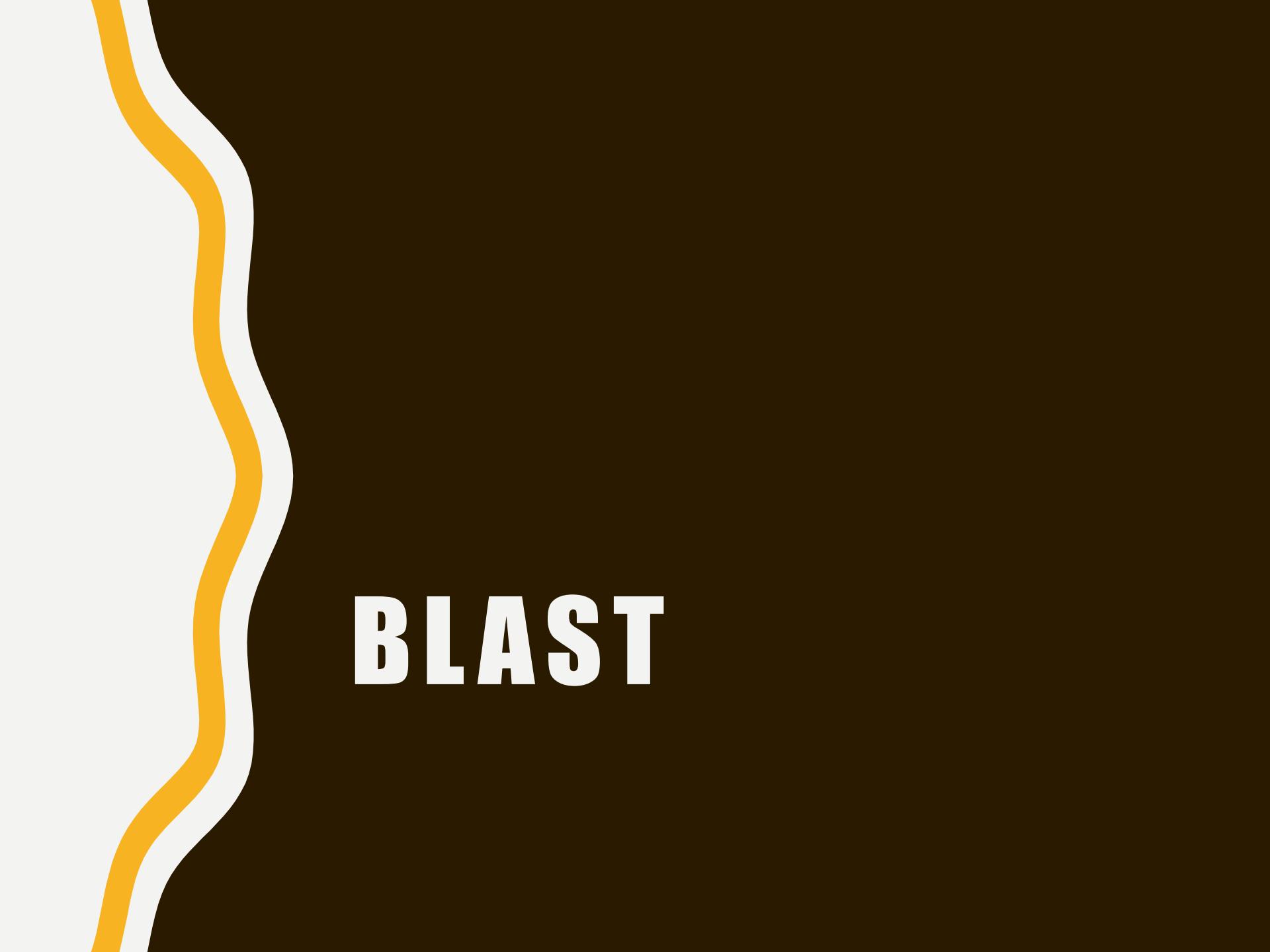
```
getAnnoFasta.pl siskin_augustus_all.gff --  
seqfile=.../.../repmasker/siskin_10largest.fasta.m  
asked --chop_cds --protein=on --codingseq=on
```

- What files do we have now? Take a look inside to figure out what kind of file we have, how many sequences, etc.



WHAT DO WE DO WITH THE ANNOTATION?

- Visualize the assembly and annotation using a genome browser.
- Functional annotation: BLAST and BLAST2GO



BLAST

BLAST

- Basic Local Alignment Search Tool

BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance.

Basic Local Alignment Search Tool

BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance.

[Learn more](#)

NEWS

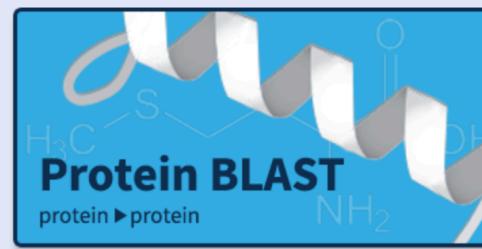
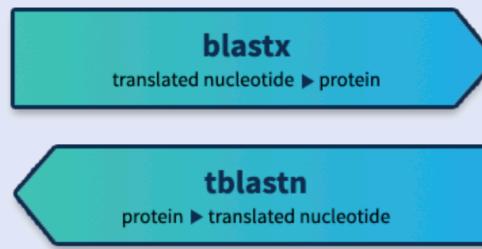
End of updates for BLAST+ version 4 databases (dbV4)

Start moving to the new version 5 databases!

Fri, 27 Sep 2019 16:00:00 EST

[More BLAST news...](#)

Web BLAST



BLAST Genomes

[Search](#)[Human](#)[Mouse](#)[Rat](#)[Microbes](#)

<https://blast.ncbi.nlm.nih.gov/Blast.cgi>

WHAT HAVE YOU USED BLAST FOR?

- Command line vs web-interface vs Geneious?
- Species?

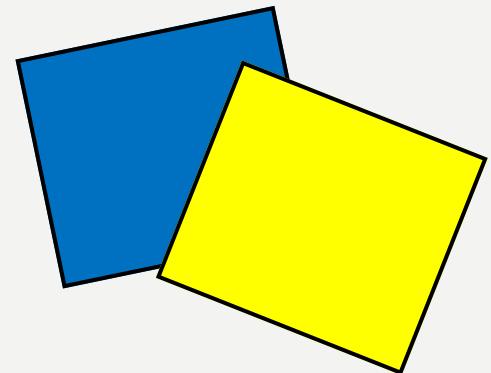
Write it down on the Etherpad

FROM AUGUSTUS TO BLAST

- Let's explore the augustus output file.
- What does it say to us about the gene names/functions?

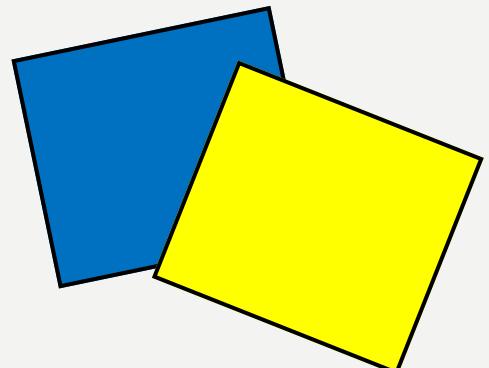
BLAST - TASKS

- Create the following folders inside your genome_annot/blast folder
 - fa
 - input_files
 - jobs
 - logs
 - results
 - xml



BLAST - TASKS

- Split the file `siskin_augustus_all.aa` into subfiles with 50 sequences each (output to the folder `scaffolds`).
- Create a txt file with a list of files in the folder `scaffolds`
- Copy the following files from
`data/genomics/workshops/SMSC_2019/` :
 - `blast_template.sh`
 - `blast_template.conf`



JOB ARRAYS

- Think about job arrays as something like multiple jobs inside one single job.
 - Day-to-day example: you need to run errands: go to the grocery store, fix the car, go to the post office.
 - Going to the grocery store is a “job”, and everything you need to buy there are your “tasks”



WHY NOT MULTIPLE INDIVIDUAL JOBS?

- You can do that, but you will end up (in our example) with 38 jobs instead of one.

(also remember that here we have only 10 scaffolds being annotated. A full assembly contains hundreds or thousands of scaffolds.)

BLAST2GO

- Blast2GO is a bioinformatics platform for functional annotation and analysis of genomic datasets.



BLAST2GO

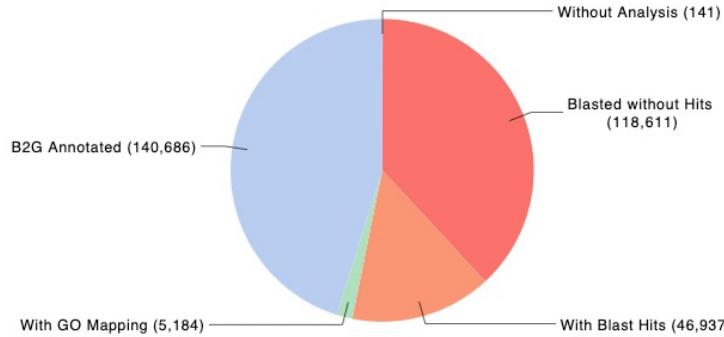


Figure: The data distribution pie chart shows the amount of sequences which could finally be annotated in comparison to the ones not annotated due to missing results in the blast, mapping or annotation step.

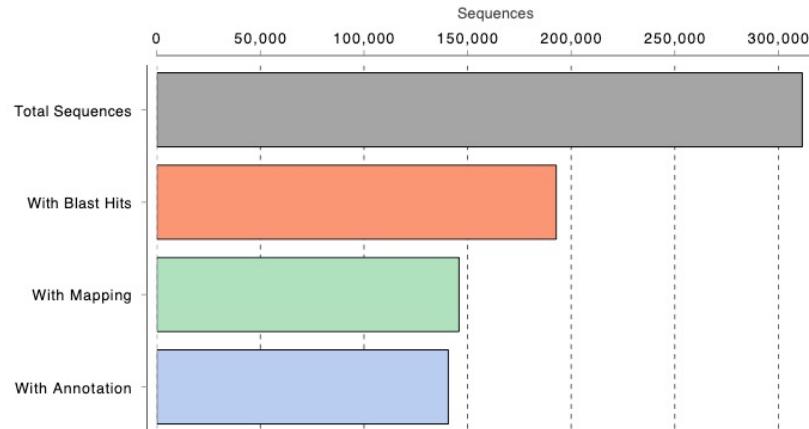


Figure: The Analysis Progress shows the total amount of sequences which obtained results during the different analysis steps. Please note that for example the total amount of mapped (green) sequences cannot be higher than the number of blasted (orange) sequences.

Contig4333_pilon_1996027-1996140	GO:0005654	stAR-related lipid transfer 9-like
Contig4333_pilon_1974461-1974668	GO:0005654	stAR-related lipid transfer 9
Contig4333_pilon_1969389-1969569	GO:0005654	stAR-related lipid transfer 9 isoform X3
Contig4333_pilon_1963057-1963195	GO:0005654	stAR-related lipid transfer 9
Contig4333_pilon_1961730-1961834	GO:0005654	stAR-related lipid transfer 9
Contig4333_pilon_1953012-1953203	GO:0005654	stAR-related lipid transfer 9
Contig4333_pilon_1952067-1952542	GO:0005654	stAR-related lipid transfer 9
Contig4333_pilon_1939734-1949907_ORF	GO:0005654	stAR-related lipid transfer 9 isoform X2
Contig4333_pilon_1939365-1939557	GO:0005654	stAR-related lipid transfer 9
Contig4333_pilon_1938748-1938932	GO:0005654	stAR-related lipid transfer 9
Contig4333_pilon_1937285-1937423	GO:0005737	stAR-related lipid transfer 9 isoform X8
Contig4333_pilon_1928071-1928183	GO:0005654	stAR-related lipid transfer 9
Contig4333_pilon_1927775-1927885	GO:0005654	stAR-related lipid transfer 9 isoform X1
Contig4333_pilon_1927085-1927288	GO:0005654	stAR-related lipid transfer 9 isoform X2
Contig4333_pilon_1926410-1926569	GO:0005654	stAR-related lipid transfer 9 isoform X1
Contig4333_pilon_641283-641411	GO:0005737	ARL14 effector
Contig4333_pilon_642447-642676	GO:0005737	ARL14 effector
Contig4333_pilon_264232-264381	GO:0005654	phosphatidylinositol transfer , cytoplasmic 1, isoform CRA_b
Contig4333_pilon_722350-722464	GO:0005515	metallophosphoesterase MPPED2 isoform X6
Contig4333_pilon_719575-719694	GO:0005515	metallophosphoesterase MPPED2 isoform X6
Contig4333_pilon_1205369-1205570	GO:0005622	doublecortin domain-containing 5
Contig4333_pilon_1196437-1196608	GO:0005622	doublecortin domain-containing 5
Contig4333_pilon_2110227-2110345	GO:0005654	synaptosomal-associated 23 isoform X2
Contig4333_pilon_2099772-2099931	GO:0005654	Synaptosomal-associated 23, partial
Contig4333_pilon_2097916-2098061	GO:0005654	synaptosomal-associated 23 isoform X2
Contig4333_pilon_116451-116592	GO:0034455	nucleolar 11
Contig4333_pilon_114545-114659	GO:0034455	nucleolar 11
Contig4333_pilon_110218-110363	GO:0005634	nucleolar 11
Contig4333_pilon_95852-96037	GO:0005634	nucleolar 11
Contig4333_pilon_94779-95010	GO:0034455	nucleolar 11
Contig4333_pilon_1599797-1599909	GO:0016021	transmembrane 62 isoform X1
Contig4333_pilon_1591423-1591548	GO:0016021	transmembrane 62 isoform X4
Contig4333_pilon_2083200-2083304	GO:0070062	leucine-rich repeat-containing 57 isoform X2
Contig4333_pilon_2083546-2083652	GO:0070062	leucine-rich repeat-containing 57 isoform X2
Contig4333_pilon_2083757-2083896	GO:0016020	leucine-rich repeat-containing 57
Contig4333_pilon_2084416-2084685	GO:0070062	leucine-rich repeat-containing 57
Contig4333_pilon_2086365-2086551	GO:0070062	leucine-rich repeat-containing 57 isoform X2
Contig4333_pilon_1747375-1747517	GO:0000151	E3 ubiquitin- ligase UBR1
Contig4333_pilon_2077598-2077964_ORF	GO:0005634	programmed cell death 6 isoform X1
Contig4333_pilon_1910552-1911040	GO:0005634	codanin-1 isoform X2
Contig4333_pilon_1911260-1911467	GO:0005634	codanin-1 isoform X3
Contig4333_pilon_1911574-1911744	GO:0005634	codanin-1 isoform X3
Contig4333_pilon_1912534-1912655	GO:0005634	codanin-1 isoform X4
Contig4333_pilon_1912846-1912956	GO:0005634	codanin-1 isoform X1
Contig4333_pilon_1914849-1915055	GO:0005634	codanin-1 isoform X2
Contig4333_pilon_1915393-1915514	GO:0005634	codanin-1 isoform X2
Contig4333_pilon_1915646-1915793	GO:0005634	codanin-1 isoform X2

[tsuchiyam@hydra-login01 b2go]\$

BLAST2GO - GUI

- Download a trial version of Blast2GO
- Copy these files to your desktop:
 - siskin_augustus_all.aa
 - siskin_augutus_all.gff
 - siskin_blastp_all.xml



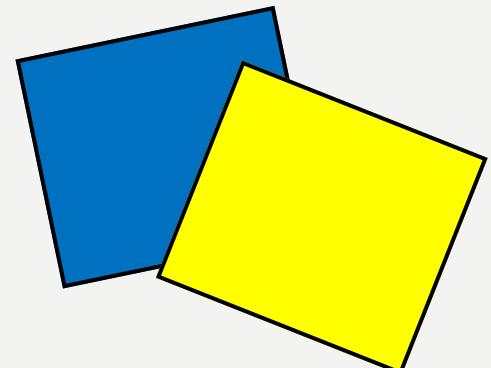
JBROWSE

JBROWSE



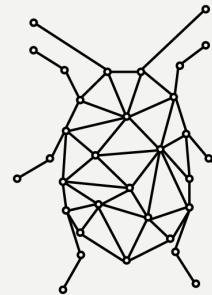
TASKS

- Run the JBrowse scripts on your files. You need:
 - Assembly
 - GFF (final)
- Compress the file and copy it to your Desktop.



JBROWSE

- You can deploy an instance locally (from your computer) or you can use a cloud service (AWS, Azure, etc)



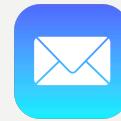
OCIO
DATA
SCIENCE
LAB



@SIDatascience



datascience.si.edu



tsuchiyam@si.edu



@MirianTsuchiya



Hydra help: SI-HPC@si.edu

Bug #415

Bug #416

Bug #417

Bug #418

Bug #419

Bug #420



**99 little bugs in the code.
99 little bugs in the code.
Take one down, patch it around.

127 little bugs in the code...**