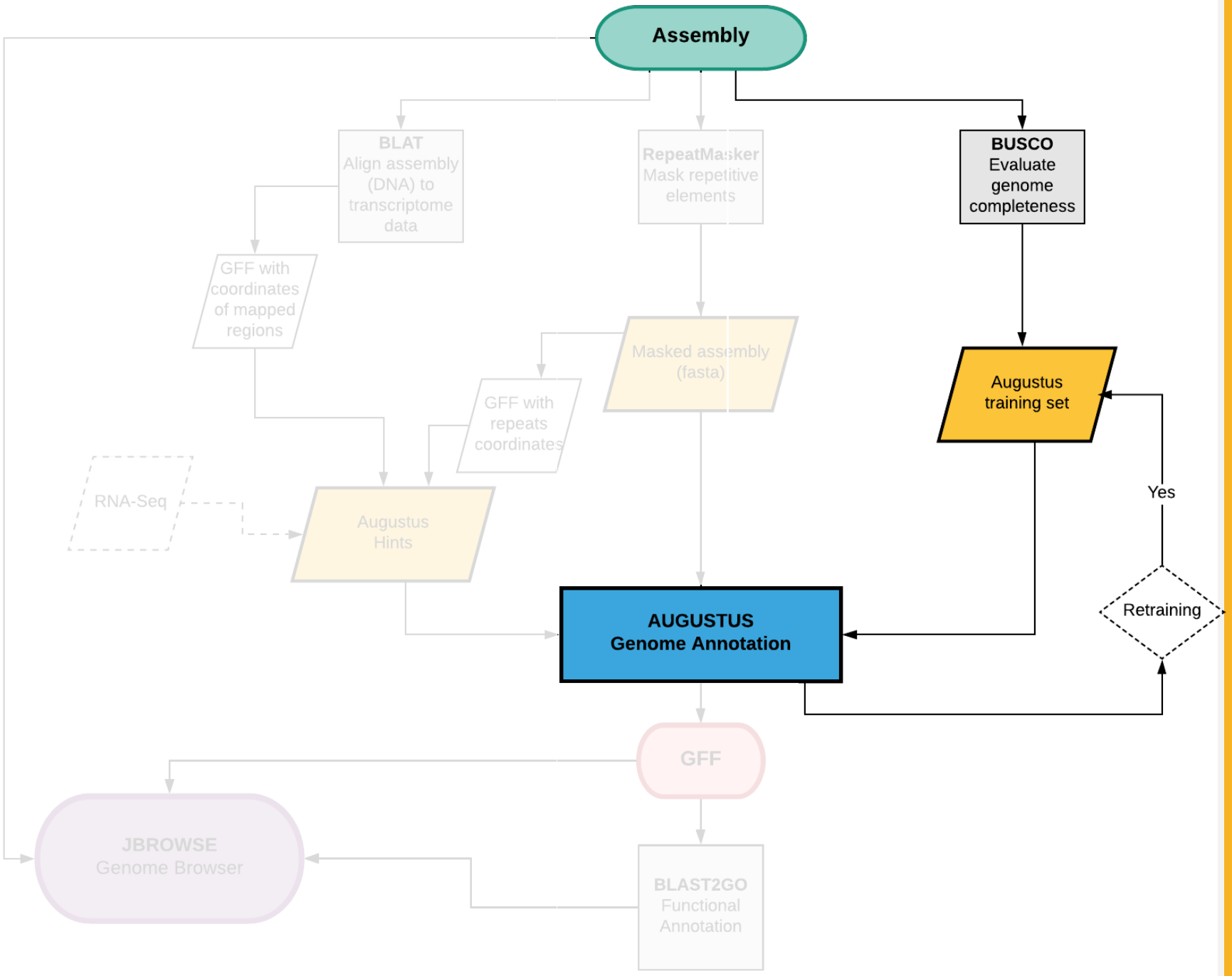
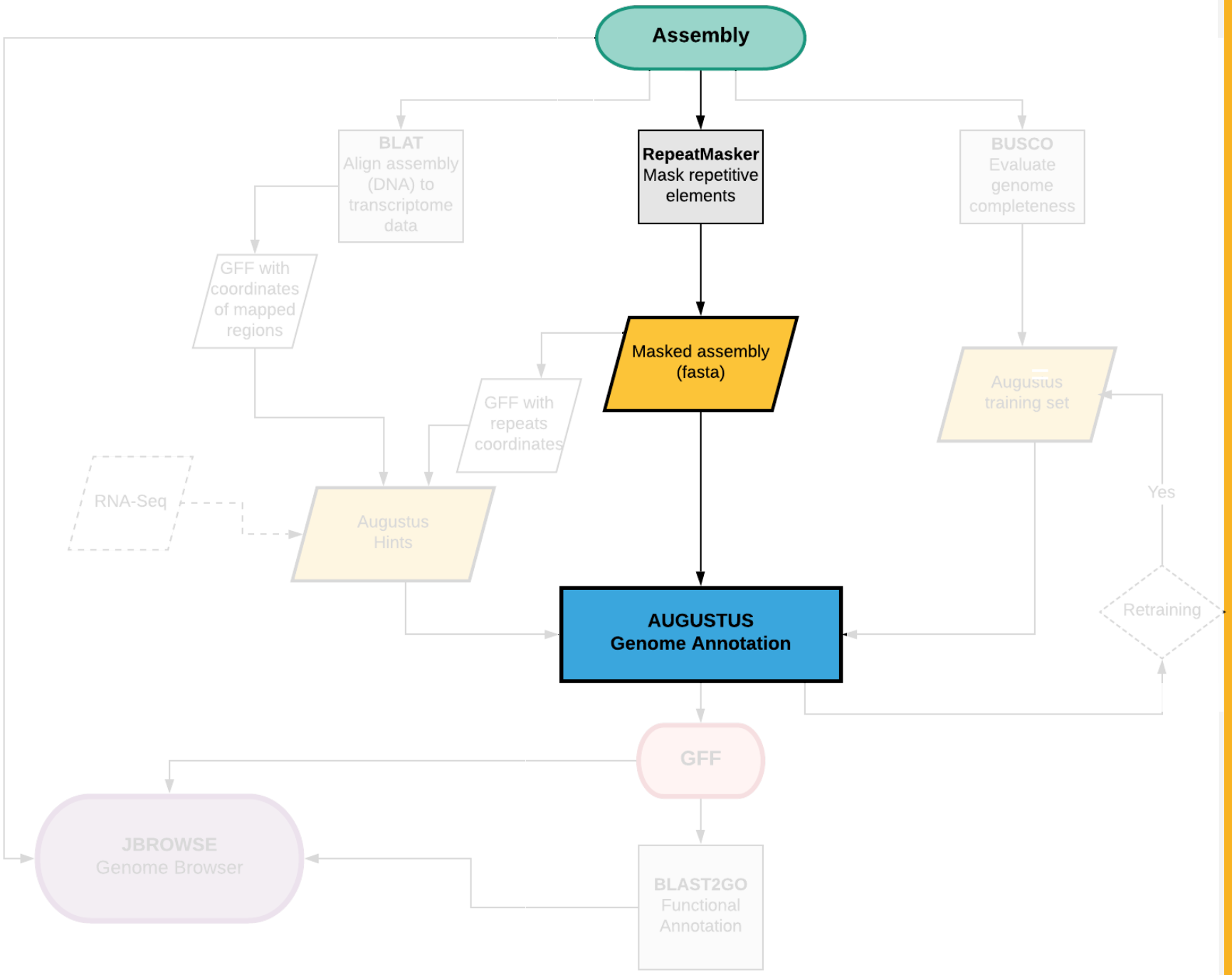
GENOME ANNOTATION

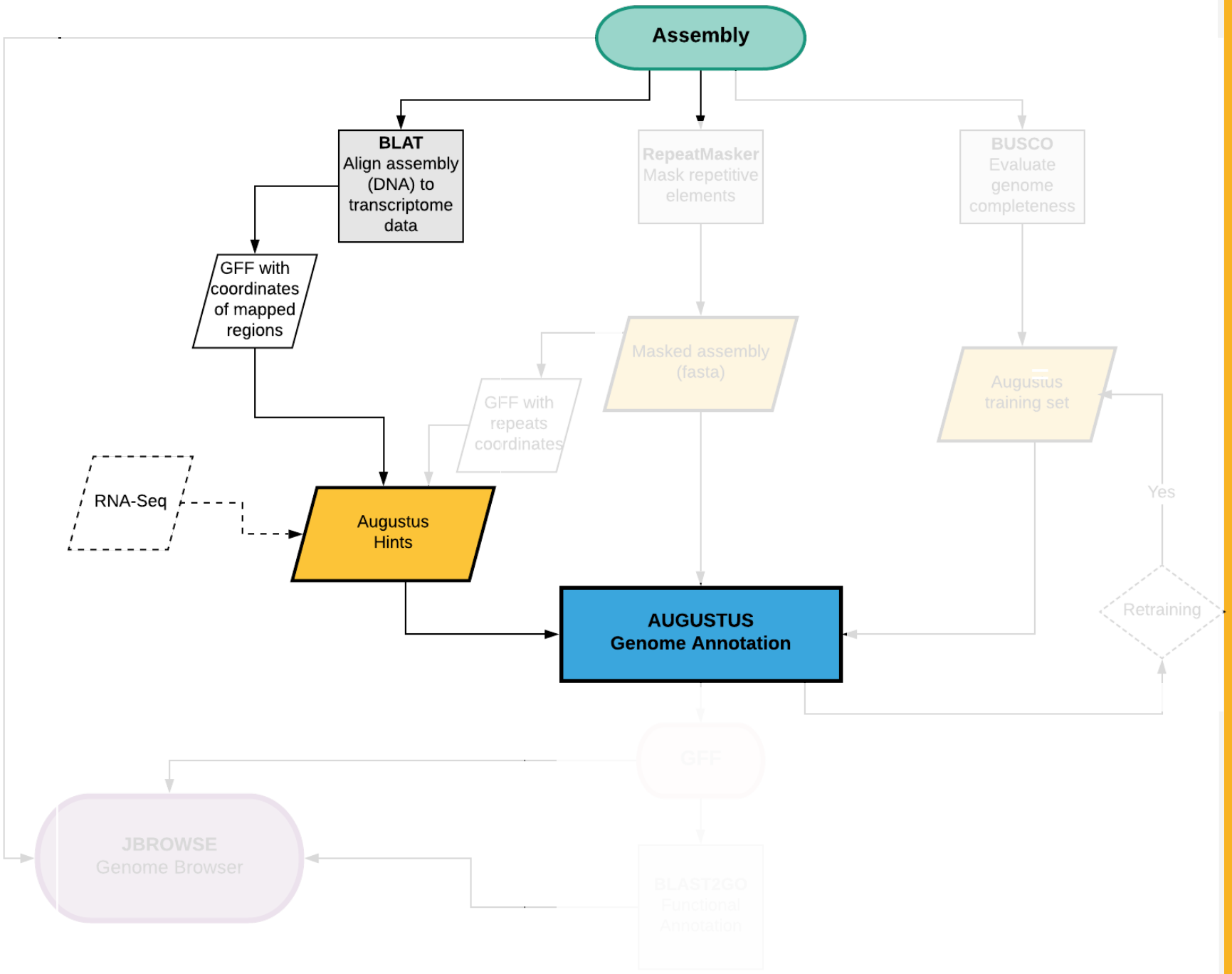
MIRIAN T. N. TSUCHIYA
DATA SCIENCE POSTDOCTORAL FELLOW
DATA SCIENCE LAB - OCIO

GENOME ANNOTATION

RECAP – DAY I


Assembly
BLAT
Align assembly (DNA) to transcriptome data
RepeatMasker
Mask repetitive elements
BUSCO
Evaluate genome completeness
GFF with coordinates of mapped regions
Masked assembly (fasta)
Augustus training set
GFF with repeats coordinates
RNA-Seq
Augustus Hints
Yes
Retraining
AUGUSTUS
Genome Annotation
GFF
JBROWSE
Genome Browser
BLAST2GO
Functional Annotation


Assembly
BLAT
Align assembly (DNA) to transcriptome data
RepeatMasker
Mask repetitive elements
BUSCO
Evaluate genome completeness
GFF with coordinates of mapped regions
Masked assembly (fasta)
Augustus training set
GFF with repeats coordinates
RNA-Seq
Augustus Hints
Yes
Retraining
AUGUSTUS
Genome Annotation
GFF
JBROWSE
Genome Browser
BLAST2GO
Functional Annotation


Assembly
BLAT
Align assembly (DNA) to transcriptome data
RepeatMasker
Mask repetitive elements
BUSCO
Evaluate genome completeness
GFF with coordinates of mapped regions
Masked assembly (fasta)
GFF with repeats coordinates
Augustus training set
RNA-Seq
Augustus Hints
Yes
Retraining
AUGUSTUS
Genome Annotation
GFF
JBROWSE
Genome Browser
BLAST2GO
Functional Annotation

RESULTS

- BUSCO
- REPEAT MASKER
- BLAT

BUSCO - RESULTS

- Full assembly:

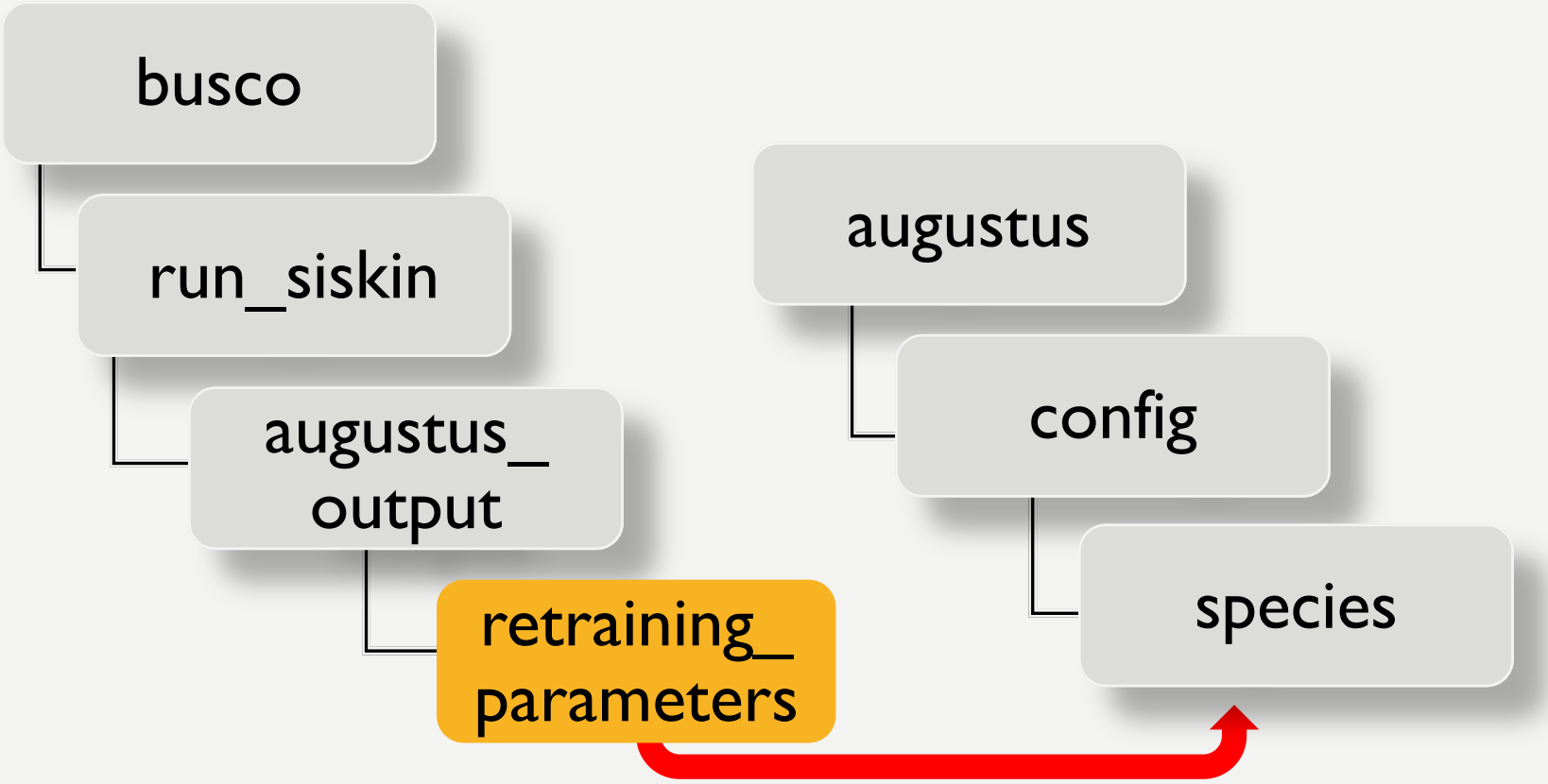
  C:93.8%[S:92.5%,D:1.3%],F:3.3%,M:2.9%,n:4915

  - 4610 Complete BUSCOs (C)
  - 4547 Complete and single-copy BUSCOs (S)
  - 63 Complete and duplicated BUSCOs (D)
  - 162 Fragmented BUSCOs (F)
  - 143 Missing BUSCOs (M)

  4915 Total BUSCO groups searched

FROM BUSCO TO AUGUSTUS

1. Copy the folder retraining_parameters to augustus/config/species



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**

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

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

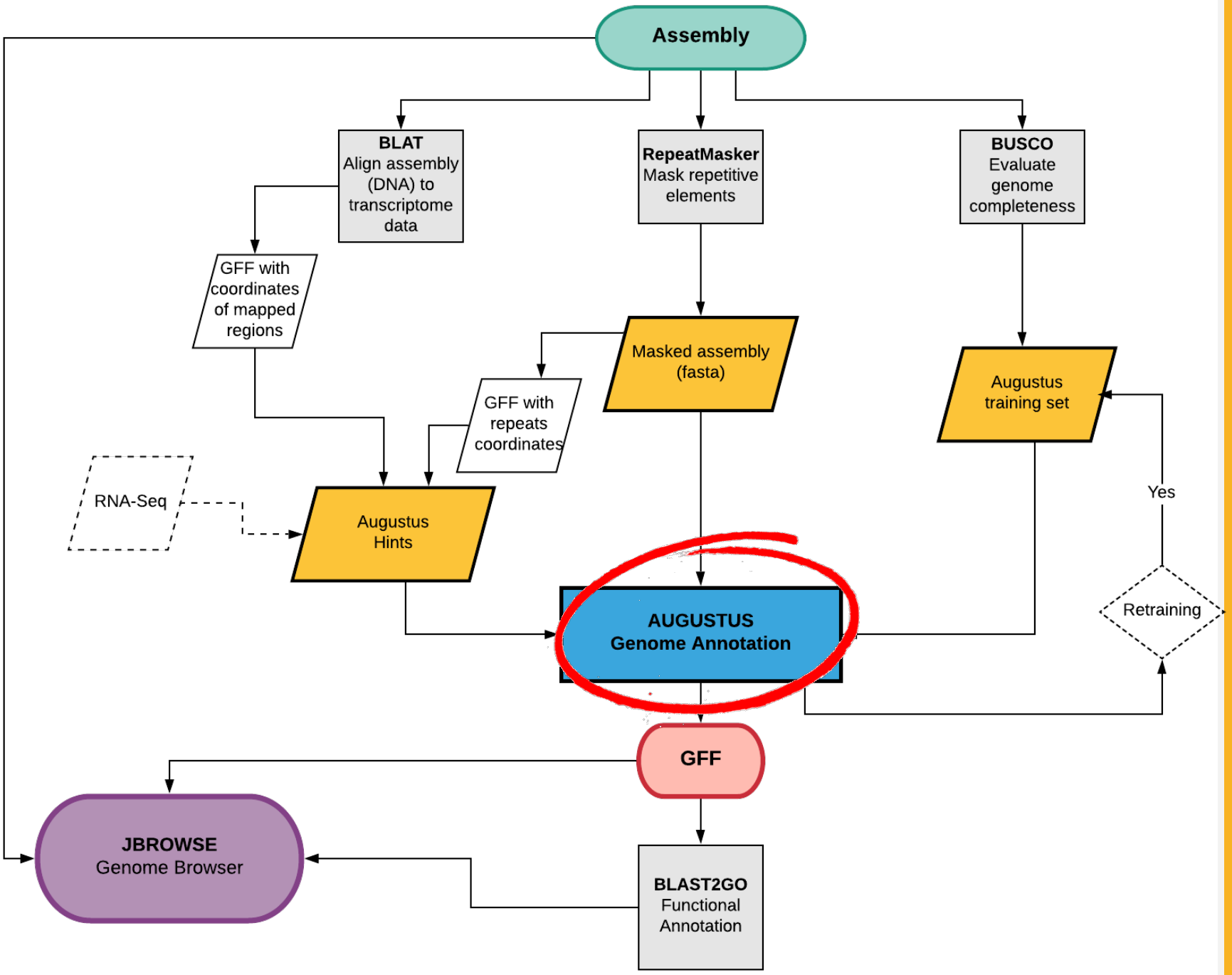
1. Load the module repeatmasker
2. Use the script rmOutToGFF3.pl to convert your .out file into GFF3
3. Use the script gff2hints to make the final conversion

AUGUSTUS: COMBINING HINTS

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

- Merge both files:

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


Assembly
BLAT
Align assembly (DNA) to transcriptome data
RepeatMasker
Mask repetitive elements
BUSCO
Evaluate genome completeness
GFF with coordinates of mapped regions
Masked assembly (fasta)
Augustus training set
GFF with repeats coordinates
RNA-Seq
Augustus Hints
Yes
AUGUSTUS
Genome Annotation
Retraining
GFF
JBROWSE
Genome Browser
BLAST2GO
Functional Annotation

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

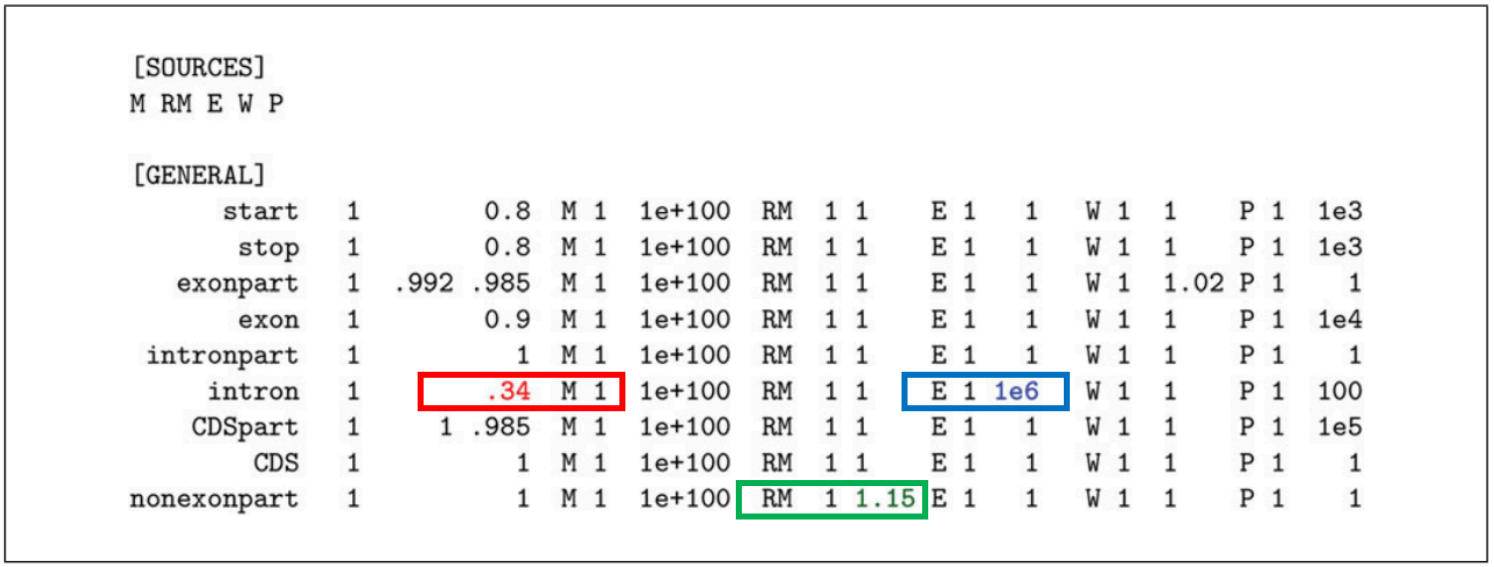
- Defines how the information from the hints will be weighted:

- 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
#
```

malus bonus bonus bonus

```
[GENERAL]
      start     1          1   M   1  1e+100  RM  1     1   E  1     1
       stop     1          1   M   1  1e+100  RM  1     1   E  1     1
        tss     1          1   M   1  1e+100  RM  1     1   E  1     1
        tts     1          1   M   1  1e+100  RM  1     1   E  1     1
        ass     1      1  0.1   M   1  1e+100  RM  1     1   E  1     1
        dss     1      1  0.1   M   1  1e+100  RM  1     1   E  1     1
   exonpart     1   .992 .985   M   1  1e+100  RM  1     1   E  1  1e2
       exon     1          1   M   1  1e+100  RM  1     1   E  1  1e4
 intronpart     1          1   M   1  1e+100  RM  1     1   E  1     1
     intron     1        .34   M   1  1e+100  RM  1     1   E  1  1e6
    CDSpart     1     1  .985  M   1  1e+100  RM  1     1   E  1     1
        CDS     1          1   M   1  1e+100  RM  1     1   E  1     1
    UTRpart     1     1  .985  M   1  1e+100  RM  1     1   E  1     1
        UTR     1          1   M   1  1e+100  RM  1     1   E  1     1
     irpart     1          1   M   1  1e+100  RM  1     1   E  1     1
nonexonpart     1          1   M   1  1e+100  RM  1     1.15 E  1     1
  genicpart     1          1   M   1  1e+100  RM  1     1   E  1     1

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

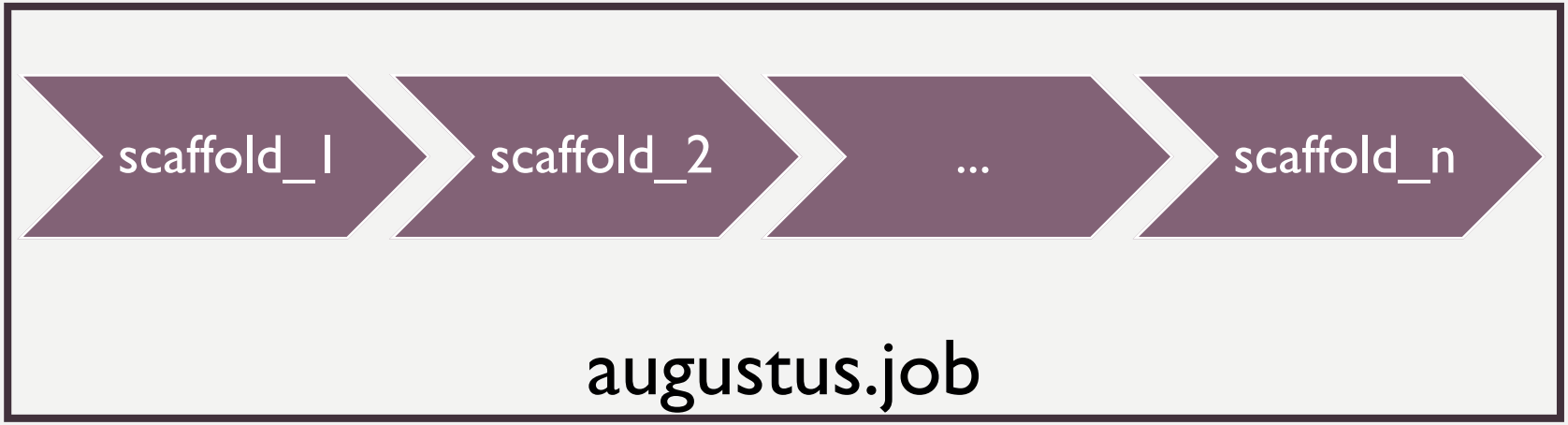


```
[SOURCES]
M RM E W P

[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
 intronpart   1           1  M 1  1e+100  RM  1 1    E 1    1   W 1  1     P 1    1
     intron   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*.

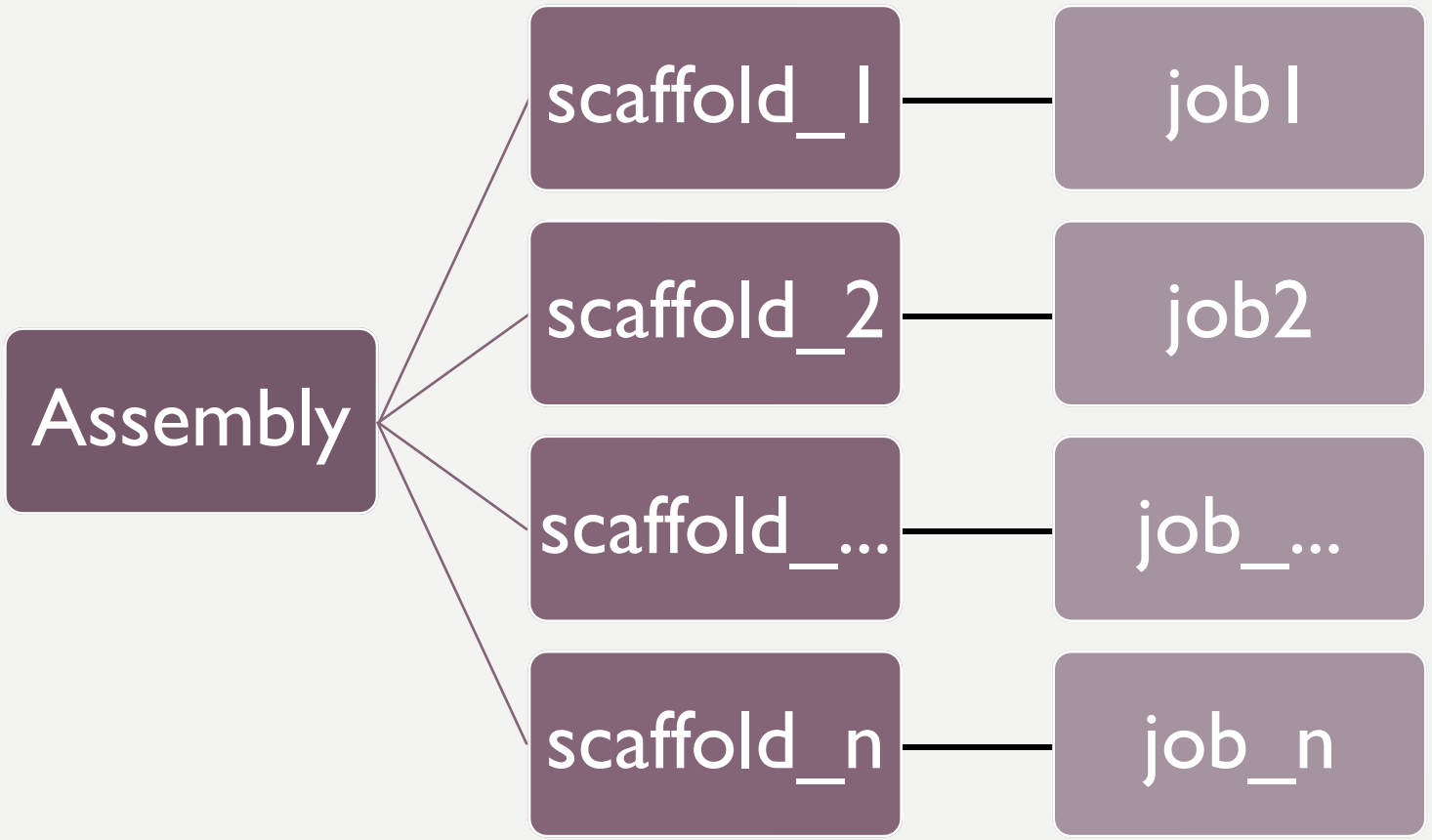
EMBARRASSINGLY PARALLEL

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



EMBARRASSINGLY PARALLEL

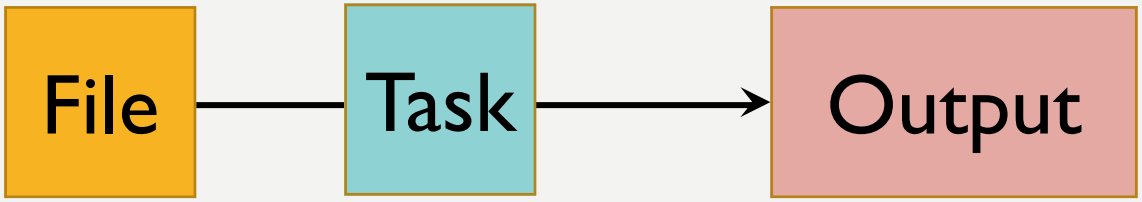
- But we can "force" it to run in parallel



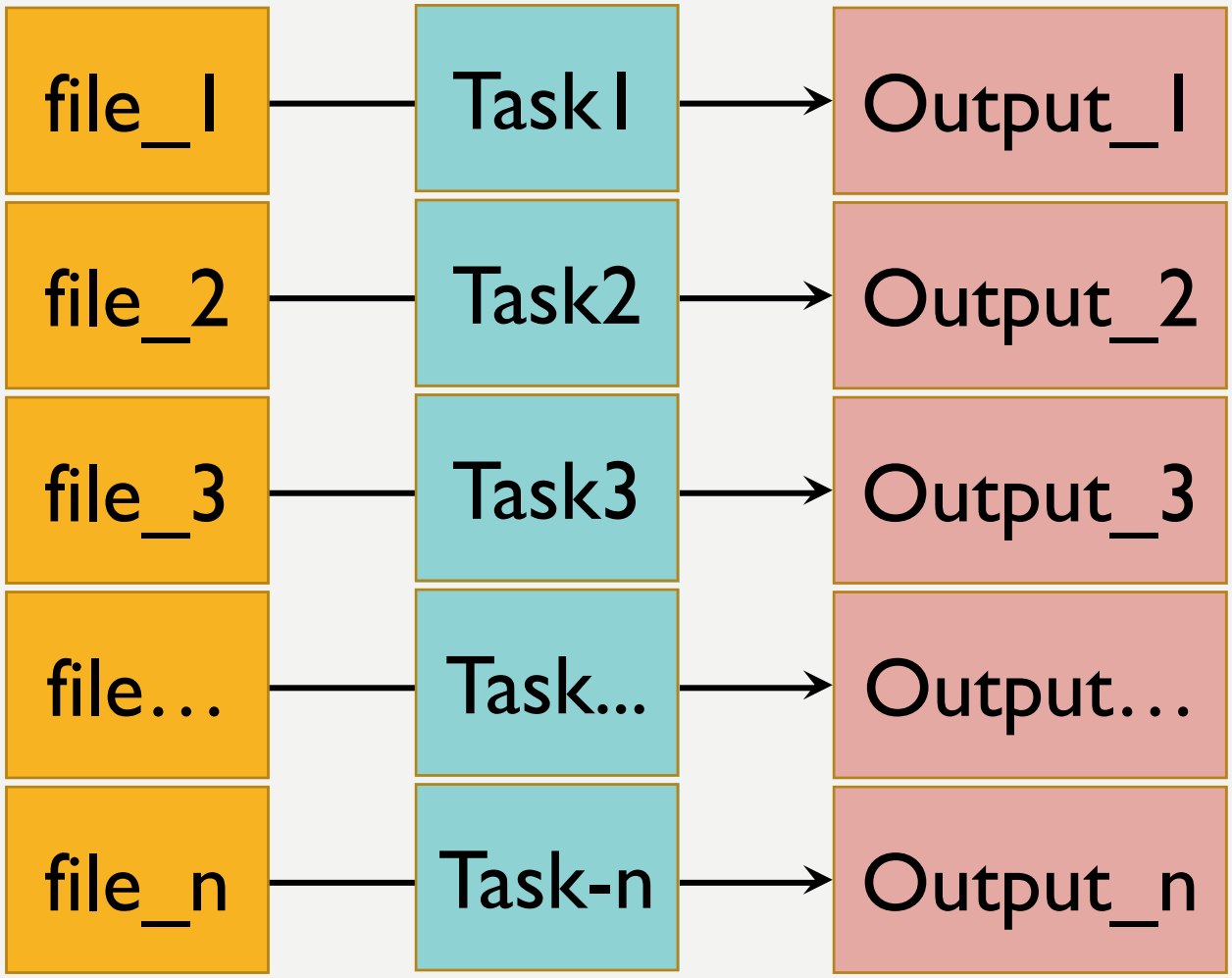
AUGUSTUS JOB FILE

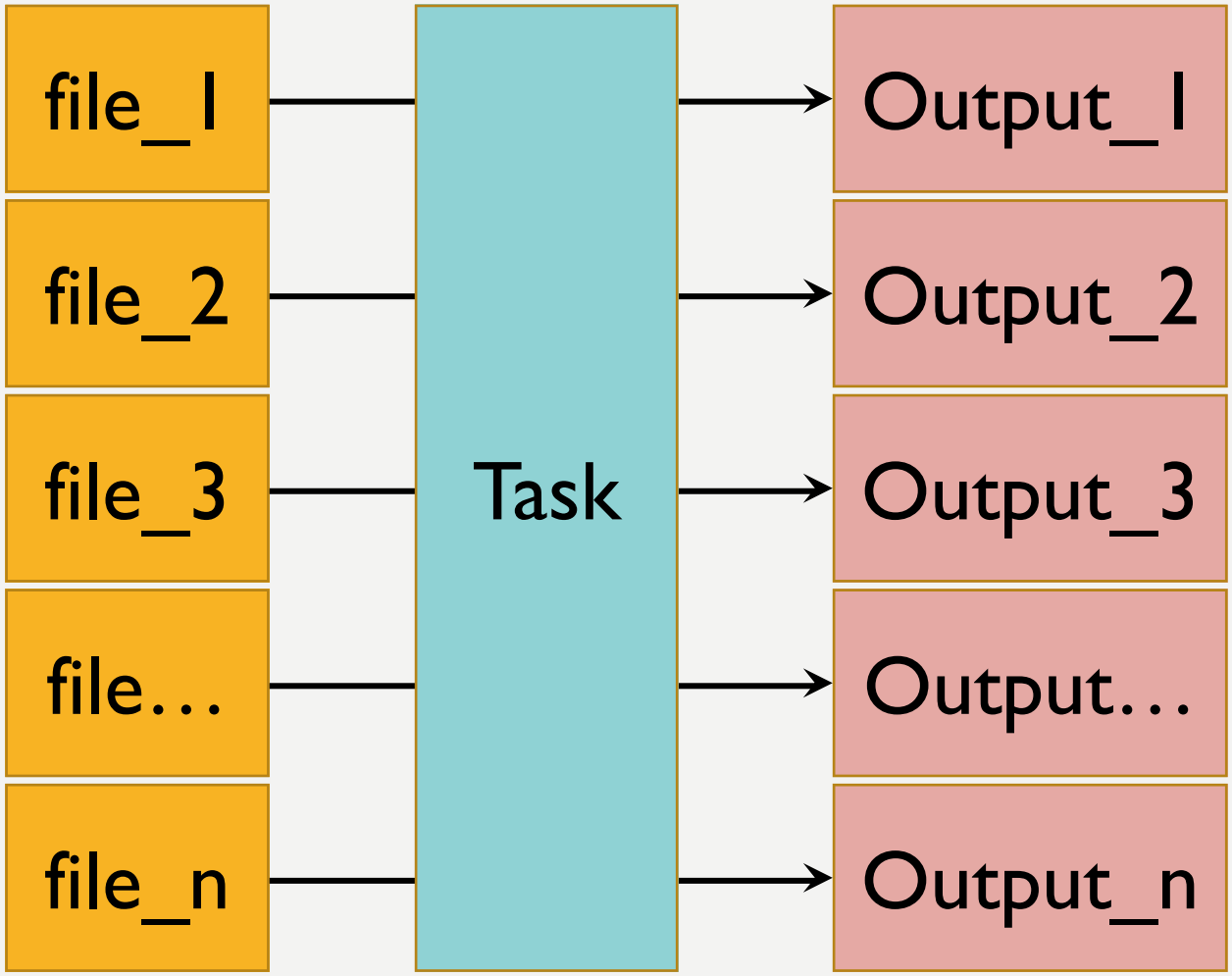
- How are we running augustus on each scaffold?
  - Option 1: Create one job file for each scaffold… manually
  - Option 2: Create one job file and use a loop to submit it.

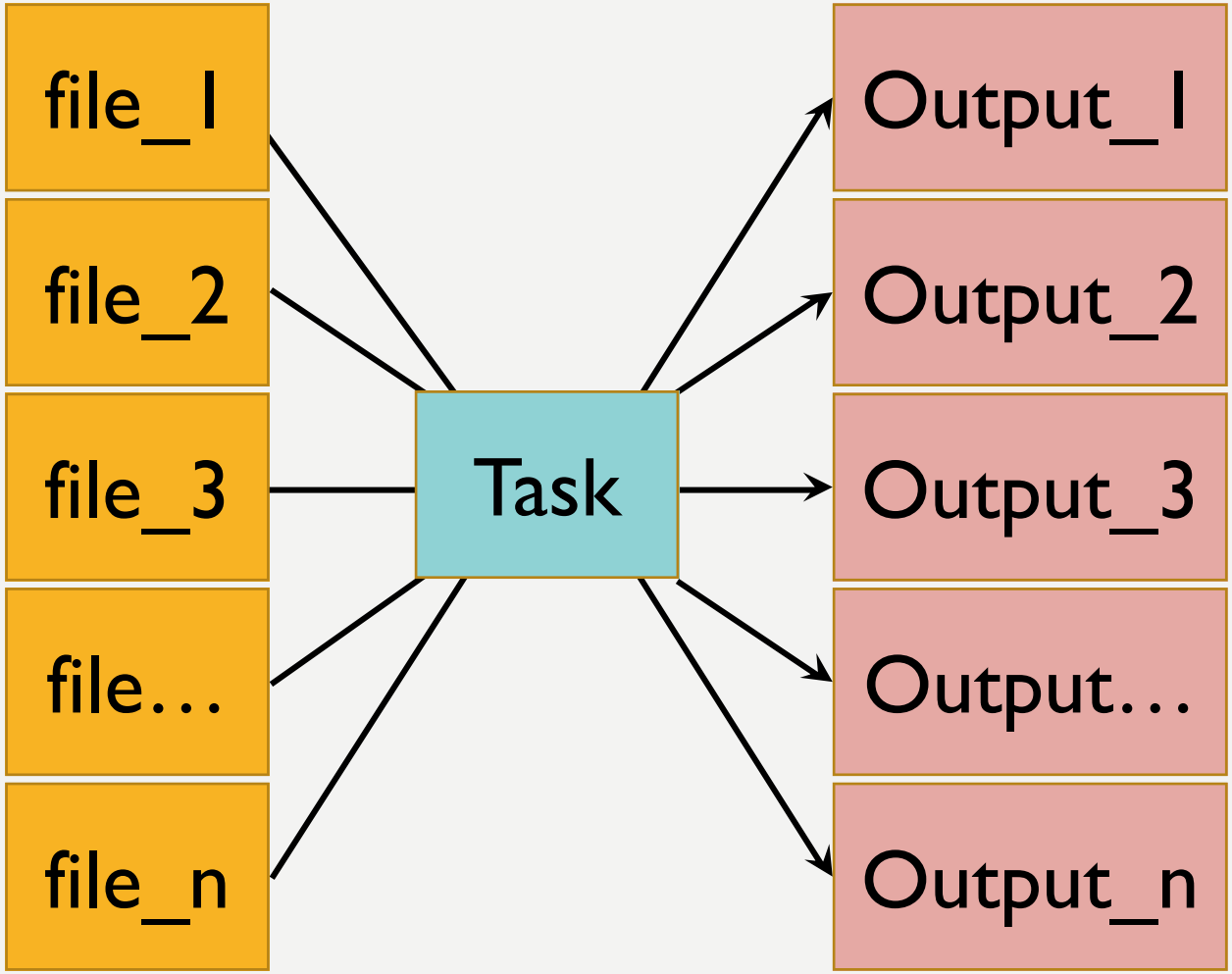
BRIEF INTRO ABOUT LOOPS



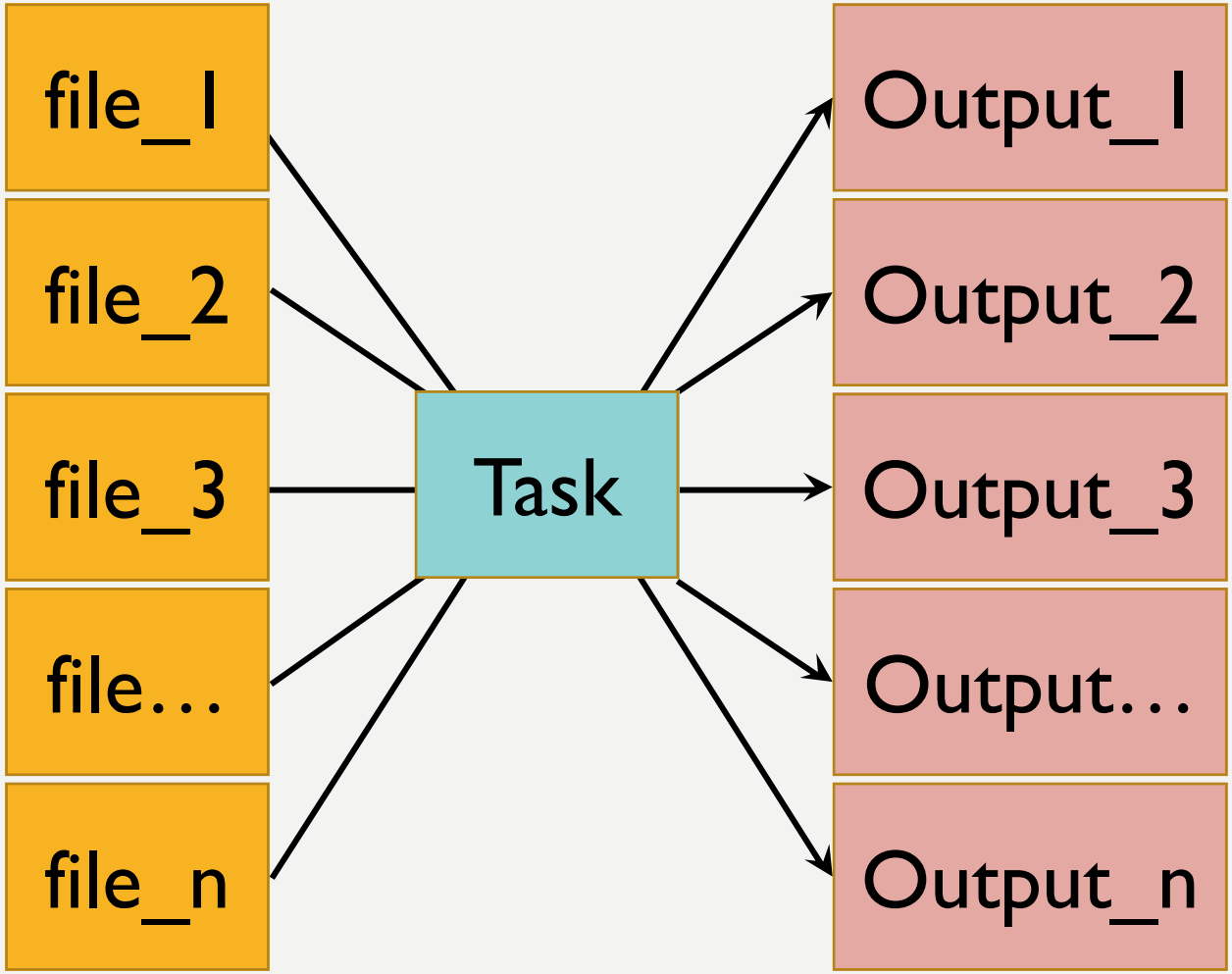
One file = One job


file_1
Task1
Output_1
file_2
Task2
Output_2
file_3
Task3
Output_3
file…
Task...
Output…
file_n
Task-n
Output_n


file_1
file_2
file_3
file…
file_n
Task
Output_1
Output_2
Output_3
Output…
Output_n


file_1
file_2
file_3
file…
file_n
Task
Output_1
Output_2
Output_3
Output…
Output_n

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



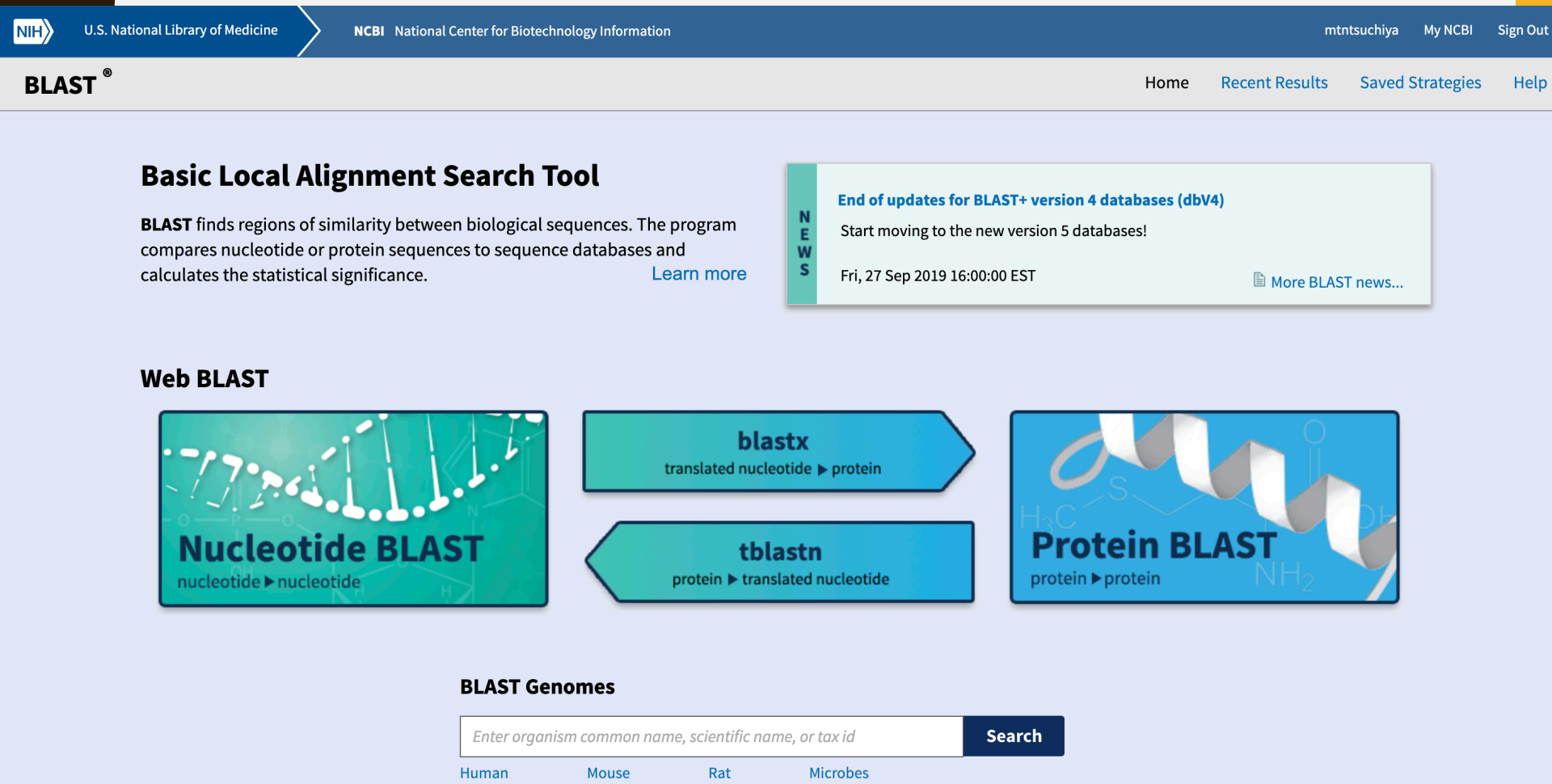
```
FOR DOG IN PUPPY_*; DO
  PLAY ${DOG} > ${DOG}.HAPPY;
DONE
```

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.



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

FROM AUGUSTUS TO BLAST

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

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

FUNCTIONAL ANNOTATION

BLAST2GO

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

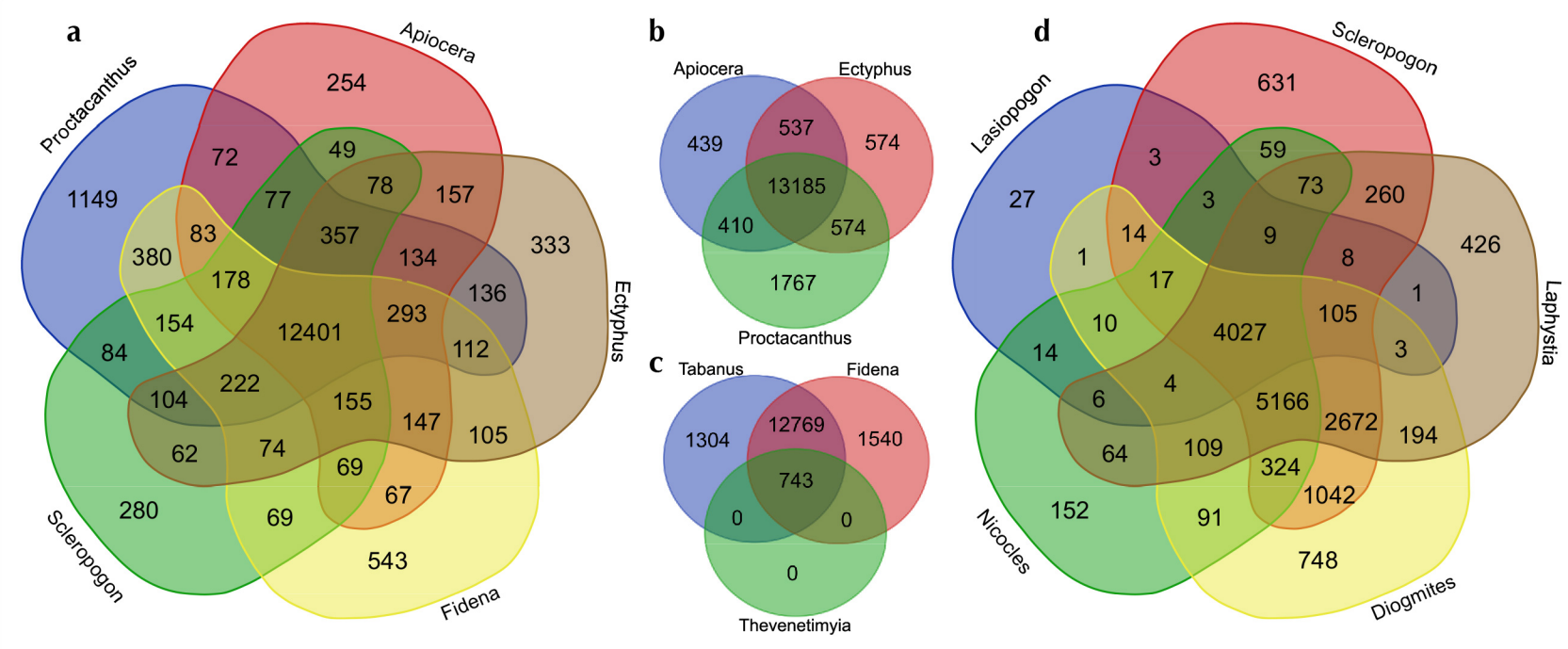
FUNCTIONAL ANNOTATION

- Gene Ontology:
  - Molecular Function (MF)
  - Cellular Component (CC)
  - Biological Process (BP)

Example: cytochrome c
-molecular function: oxidoreductase activity
-biological process: oxidative phosphorylation
-cellular component: mitochondrial matrix.

FUNCTIONAL ANNOTATION

- Software:
  - Blast2GO (available on Hydra; paid license)
  - Other (free) options: DAVID, GO FEAT



BLAST2GO

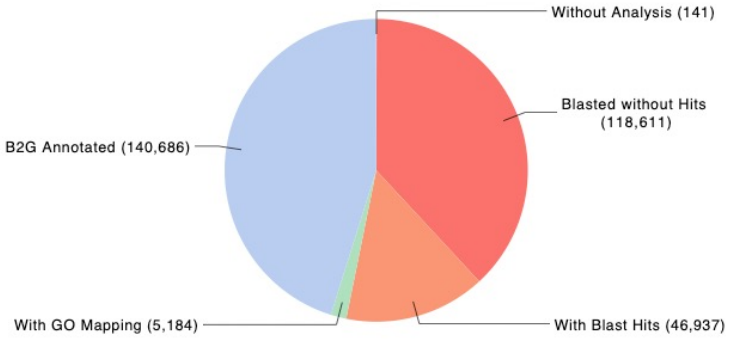


Figure: The data distribution pie chart shows the amount of sequences which could finally be annotated in comparison to the ones not annotated due to missing results in the blast, mapping or annotation step.

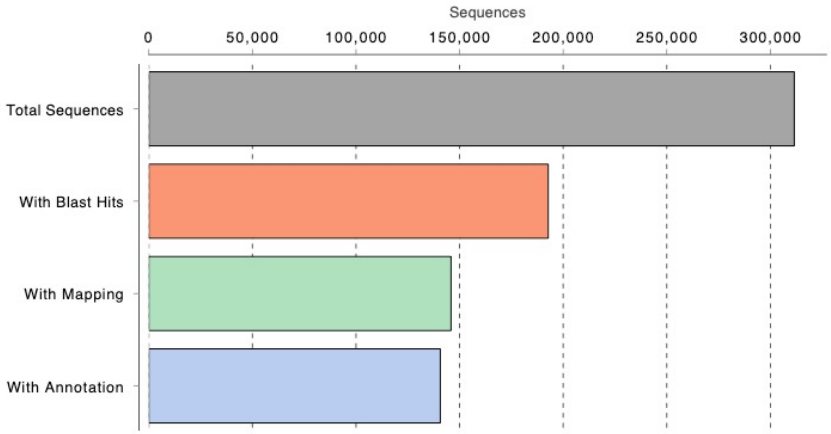
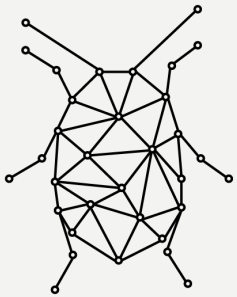


Figure: The Analysis Progress shows the total amount of sequences which obtained results during the different analysis steps. Please note that for example the total amount of mapped (green) sequences cannot be higher than the number of blasted (orange) sequences.

```
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]$
```



OCIO
DATA
SCIENCE
LAB

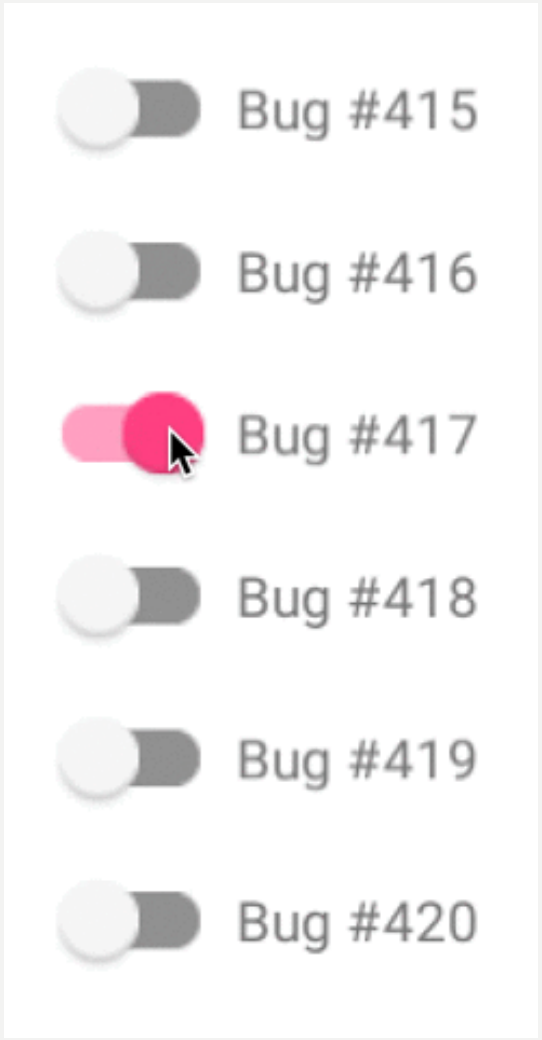
@SIDatascience

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...

JBROWSE

JBROWSE


Available Tracks
filter tracks
BAM 8
BAM - paired-read test pattern at 28kb
BAM - volvox-longreads.bam
BAM - volvox-sorted SNPs/Coverage
BAM - volvox-sorted.bam
BAM - volvox-sv.bam
CRAM - volvox-longreads.cram
CRAM - volvox-sorted.cram
CRAM - volvox-sv.cram
Miscellaneous 15
BEDTabix - volvox.bed in-memory adaptor
BigBed - volvox genes
ChromHMM
FromConfig - Features in Configuration
GFF3 - sub-subparts
GFF3 - volvox.gff3 in-memory adaptor
GFF3Tabix - volvox.gff3 in-memory adaptor
GTF - volvox.gtf in-memory adaptor
HTMLFeatures - ESTs
HTMLFeatures - Example Features
HTMLFeatures - Example motifs
HTMLFeatures - Features with right-click menus
HTMLFeatures - Fingerprinted BACs
HTMLFeatures - Name test track has a really long track label
HTMLFeatures - SNPs
Quantitative 4
Density 2
BigWig Density - volvox_microarray
BigWig Density - volvox_sine
XY Plot 2
BigWig XY - volvox_microarray
BigWig XY - volvox_sine
Reference sequence 1
Reference sequence
Transcripts 5
Genome
Track
View
Help
Volvox Example
Share
ctgA
ctgA:4411..6492 (2.08 Kb)
Go
Reference sequence
Zoom in to see sequence
CanvasFeatures - transcripts
BigWig Density - volvox_microarray
BigWig XY - volvox_microarray
mean
VCF - volvox-sorted variants
SNV C -> G
BAM - volvox-sorted SNPs/Coverage
BAM - volvox-sorted.bam

JBROWSE

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