Genome Annotation

Qi Sun

Bioinformatics Facility

Cornell University

Two Steps in Genome Annotation

1. Identify genes on the genome

Output files

GFF3 file

Chr1	AUGUSTUS	gene	12023	14578	.	.	.	ID=g122
Chr1	AUGUSTUS	mRNA	12023	14578	.	.	.	ID=t122; Parent=g122
Chr1	AUGUSTUS	exon	12023	13001	.	.	.	ID=t122_1; Parent=t122
Chr1	AUGUSTUS	exon	13995	14578	.	.	.	ID=t122_2; Parent=t122

GTF file

```
1       AUGUSTUS        gene    1       9276    0.29    +       .       g1
1       AUGUSTUS        transcript      1       9276    0.29    +       .       g1.t1
1       AUGUSTUS        intron  1       7573    0.96    +       .       transcript_id "g1.t1"; gene_id "g1";
1       AUGUSTUS        CDS     7574    8116    0.97    +       1       transcript_id "g1.t1"; gene_id "g1";
1       AUGUSTUS        exon    7574    8116    .       +       .       transcript_id "g1.t1"; gene_id "g1";
1       AUGUSTUS        intron  8117    8228    0.37    +       .       transcript_id "g1.t1"; gene_id "g1";
1       AUGUSTUS        CDS     8229    8589    0.37    +       1       transcript_id "g1.t1"; gene_id "g1";
1       AUGUSTUS        exon    8229    8589    .       +       .       transcript_id "g1.t1"; gene_id "g1";
1       AUGUSTUS        intron  8590    8667    0.84    +       .       transcript_id "g1.t1"; gene_id "g1";
```

Two Steps in Genome Annotation

2. Predict functions of each gene

Output files:

1. Gene description (human readable)

Gene ID	Gene description
GRMZM2G002950	Putative leucine-rich repeat receptor-like protein kinase family
GRMZM2G006470	Uncharacterized protein
GRMZM2G014376	Shikimate dehydrogenase; Uncharacterized protein

2. Gene Ontology (GO) (machine readable)

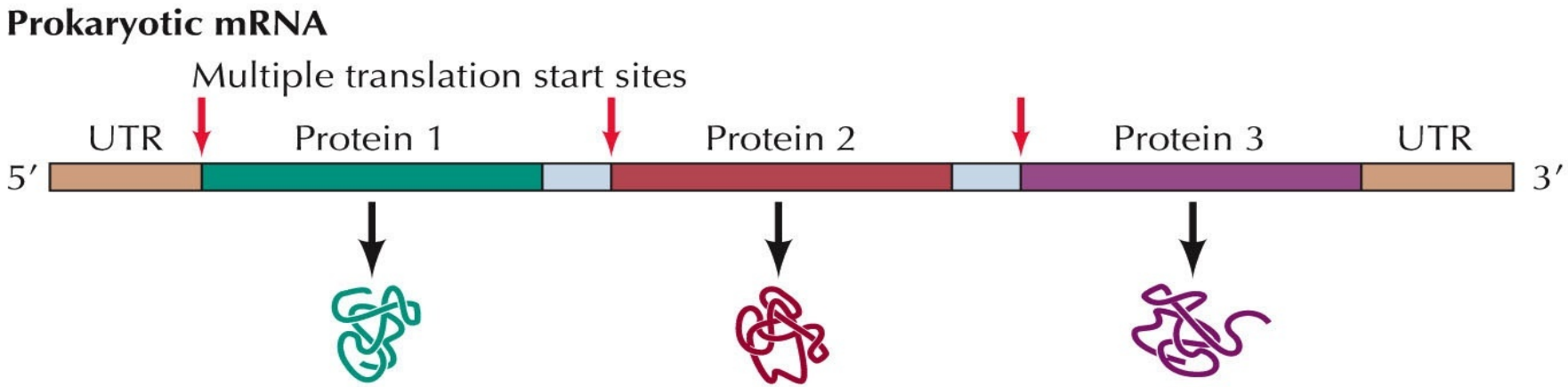
Gene ID	GO
GRMZM5G888620	GO:0003674
GRMZM5G888620	GO:0008150
GRMZM5G888620	GO:0008152

Open source software: **InterProScan**
Commercial software: **Blast2GO** (license available on BioHPC)

Eukaryotic gene

1. Detect exons; 2. Refine exon boundaries; 3. Determine gene boundary and UTRs

Prokaryotic gene



Prokaryotic genome annotation pipelines

(not the focus of this workshop)

1.NCBI PGAP

2.Prokka

Eukaryotic genome annotation

Genome Assembly (repeat masked)

Integration

(EVM)

Evidence based

RNA-seq

From the same individual

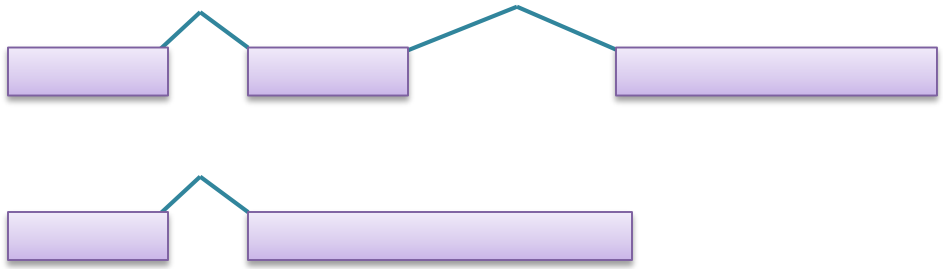
Known Proteins

(PASA)

Ab initio gene prediction

Train a model

Prediction



(Augustus through Braker2 or Maker)

Repeat masking the genome

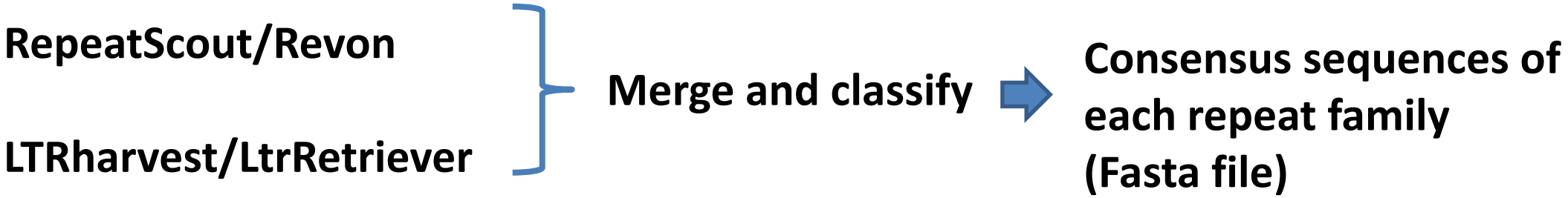
~ 1% of the human genome encode proteins. Some of the rest are regulatory, but mostly Transposable Elements (TE).

Simple repeats: E.g. “ATATATATATATAT…ATATAT”

Complex repeats: E.g. TE elements

Two steps of repeat masking

RepeatModeler : to construct a repeat database from the assembly

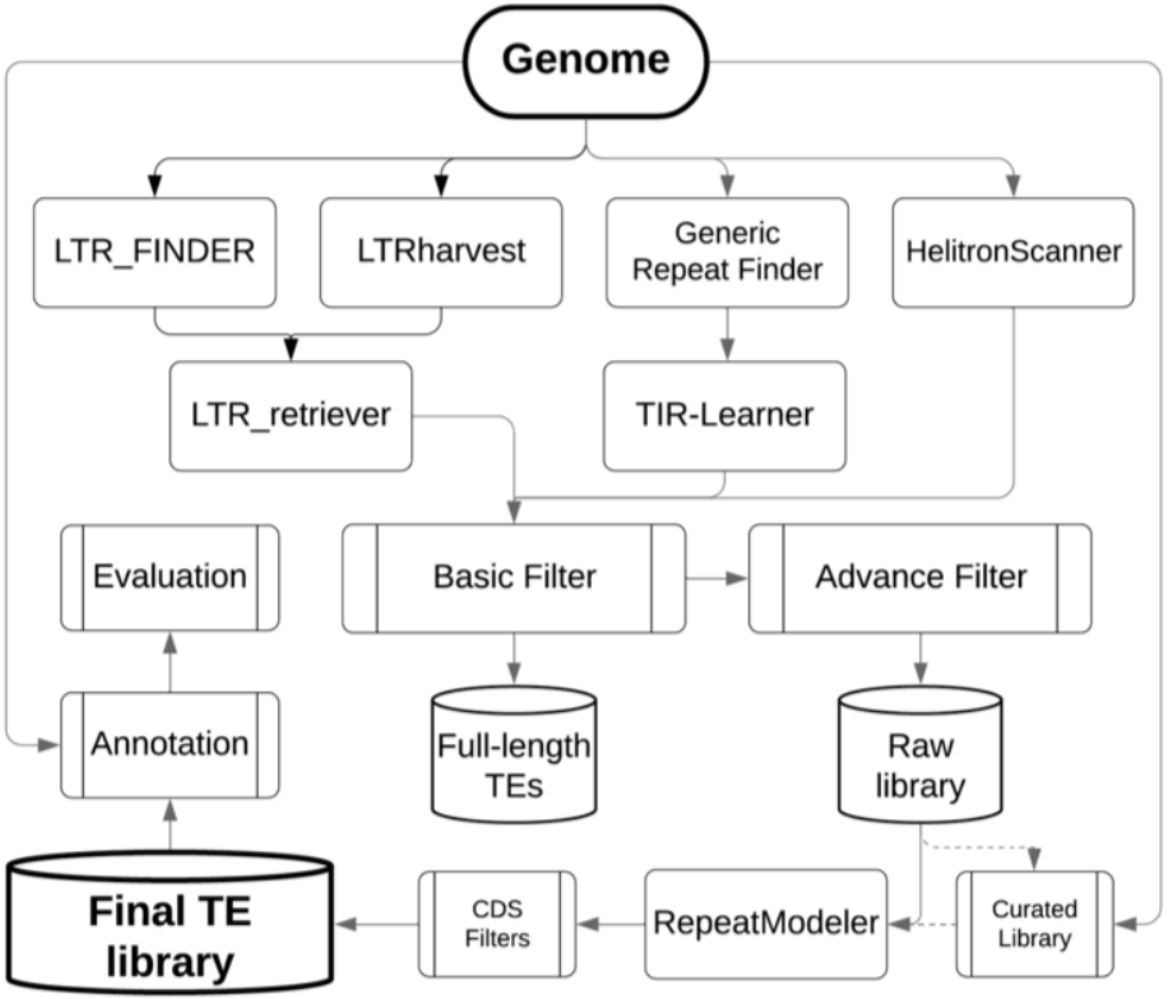


RepeatMasker:

Use the custom repeat database to mask the repeats from the assemby.

EDTA – alternative to RepeatModeler

(The Extensive *de novo* TE Annotator)



Outperform RepeatModeler for plant genomes.

Repeat masking output

Soft vs Hard Masking

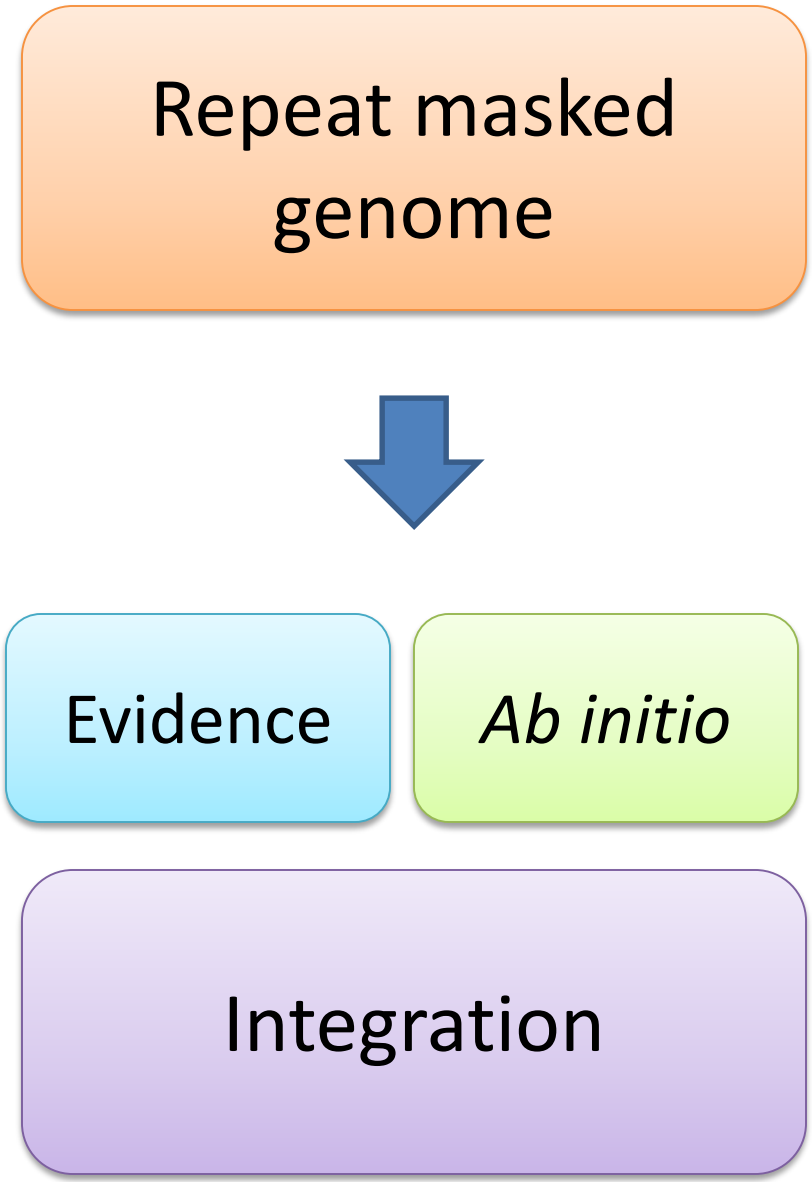
For genome annotation

Soft masking:

ACCAGAGTACTACGATAC tttttttttttttttt ACCAAACGTACAA

Hard masking:

ACCAGAGTACTACGATAC NNNNNNNNNNNNNNNNN ACCAAACGTACAA

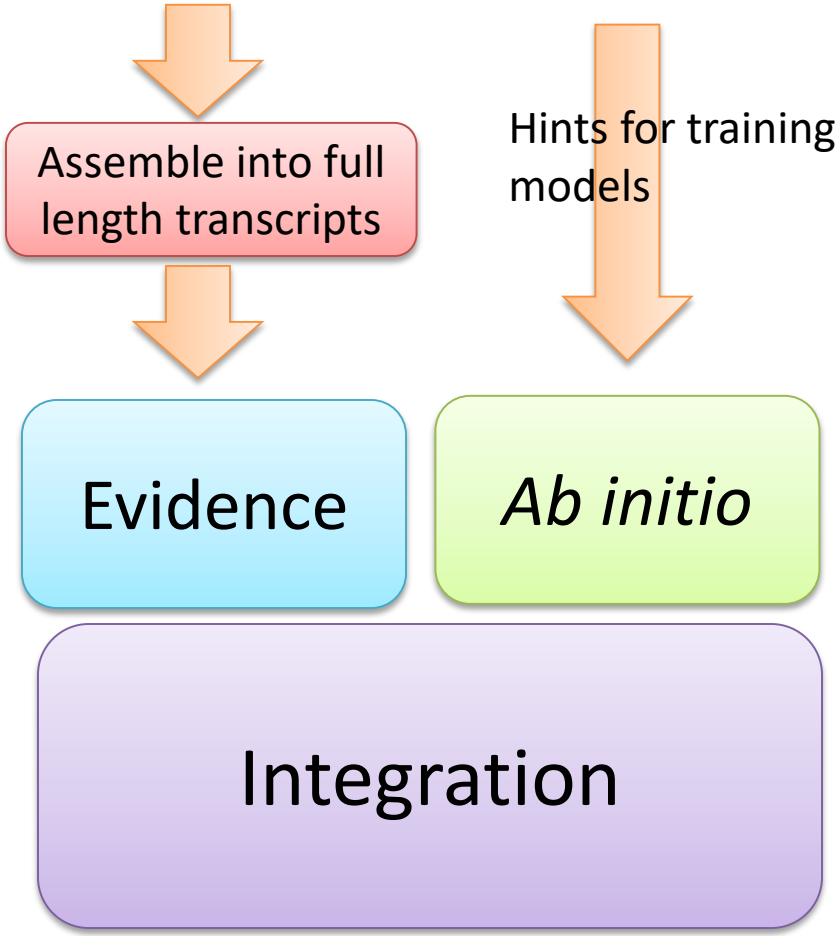


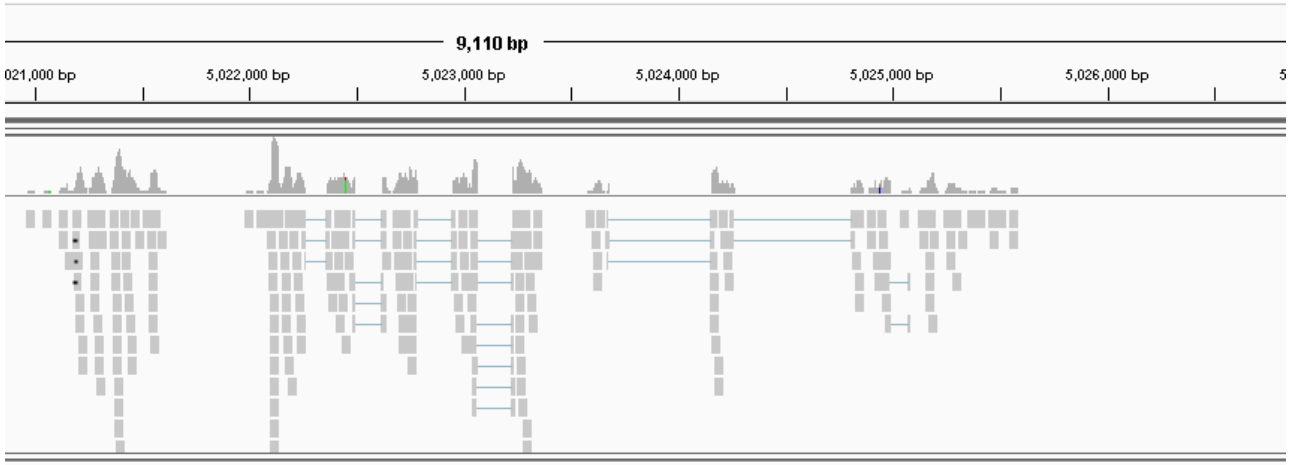
What are the evidences?

- RNA-seq (short reads): Illumina
- RNA-seq (long reads) : PacBio Iso-Seq
- Protein sequences:
  - Derived from transcripts;
  - From other genomes;
  - Mass-spec proteomics;

Illumina RNA-seq reads

2x 150 bp

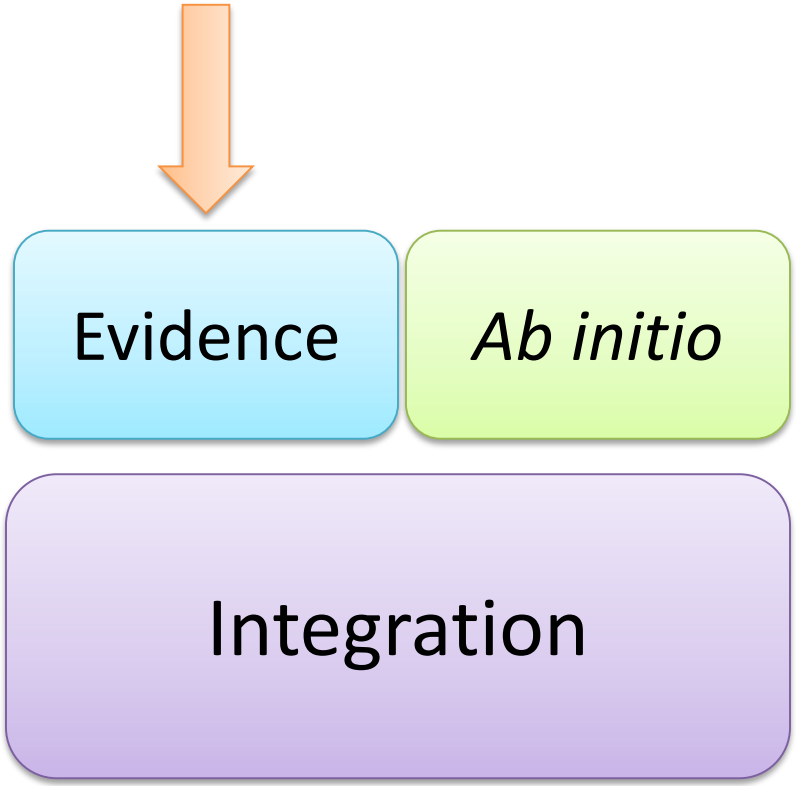




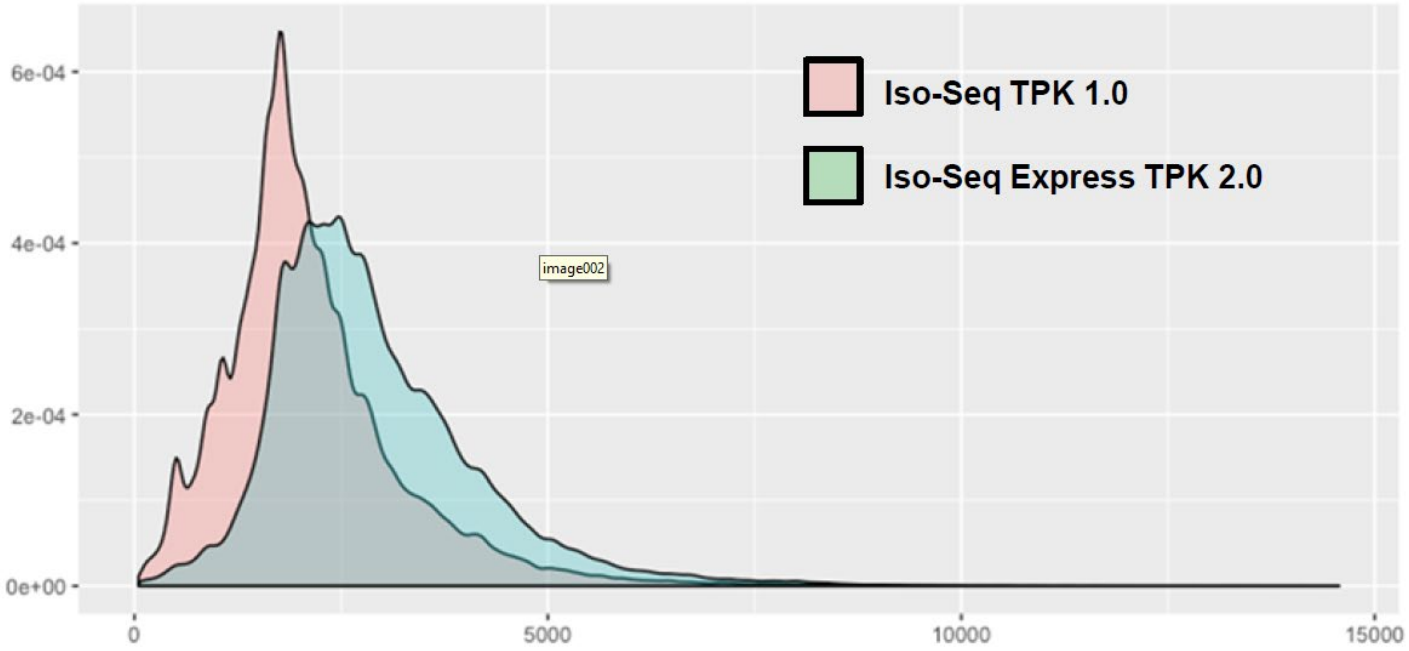
Reference guided genome assembly software

- StringTie2
- Cufflinks
- Trinity

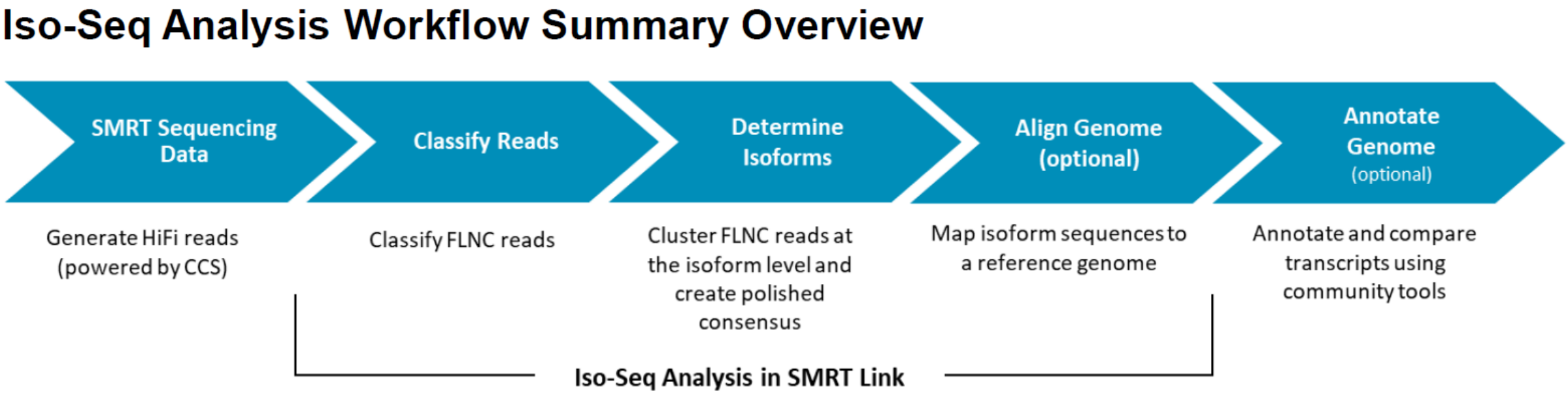
Long RNA-seq reads



Size distribution of PacBio Iso-seq reads



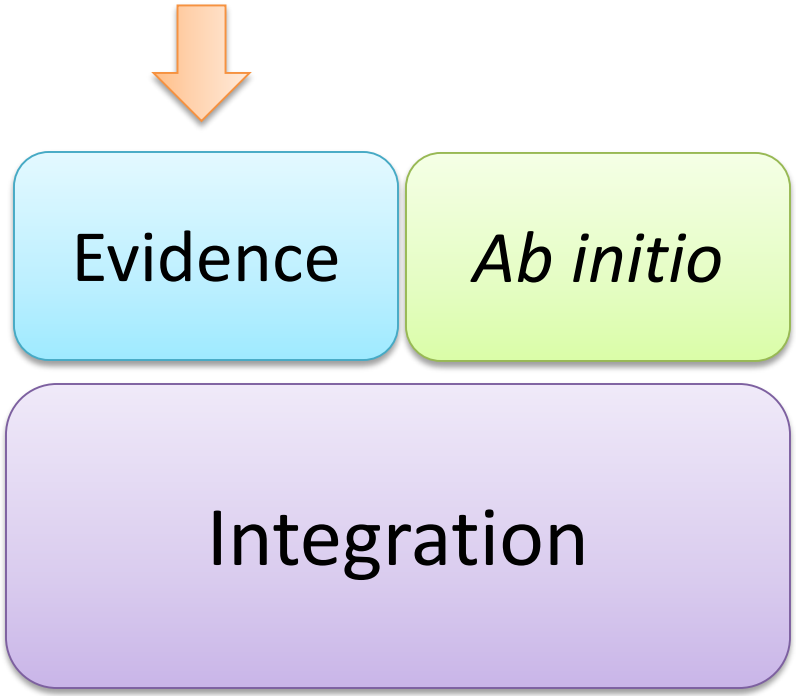
From Iso-Seq data to GFF3 annotation



GFF3

- StringTie2
- TAMA
- PASA

Protein sequences



Software

- GenomeThreader
- Exonerator
- Genewise

Data sources

Proteins from other species

- Published annotations not always reliable;
- For distantly related species, there are alignment gaps;

Mass Spec Proteomics

- Technology not mature yet

Derived from transcripts

- More reliable if from the same species (e.g. predicted with Transdecode)

Ab initio gene prediction

Why do we need it?

- RNA-seq data might not cover low-expressed genes;
- RNA might be truncated in RNA-seq data.

Ab initio gene prediction

Build a training set: evidence based genes

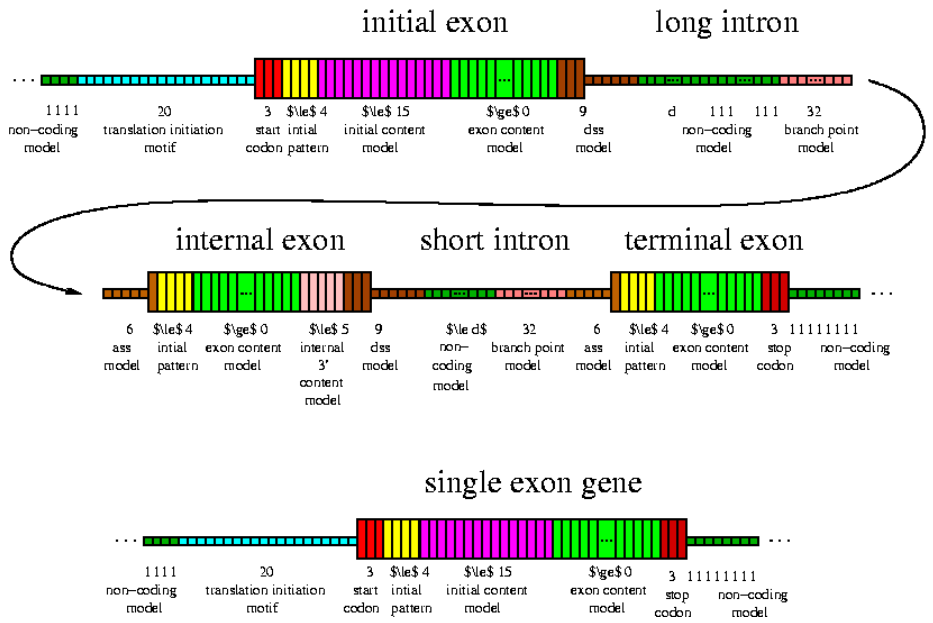
Build a model of genes

Predict more genes with the model

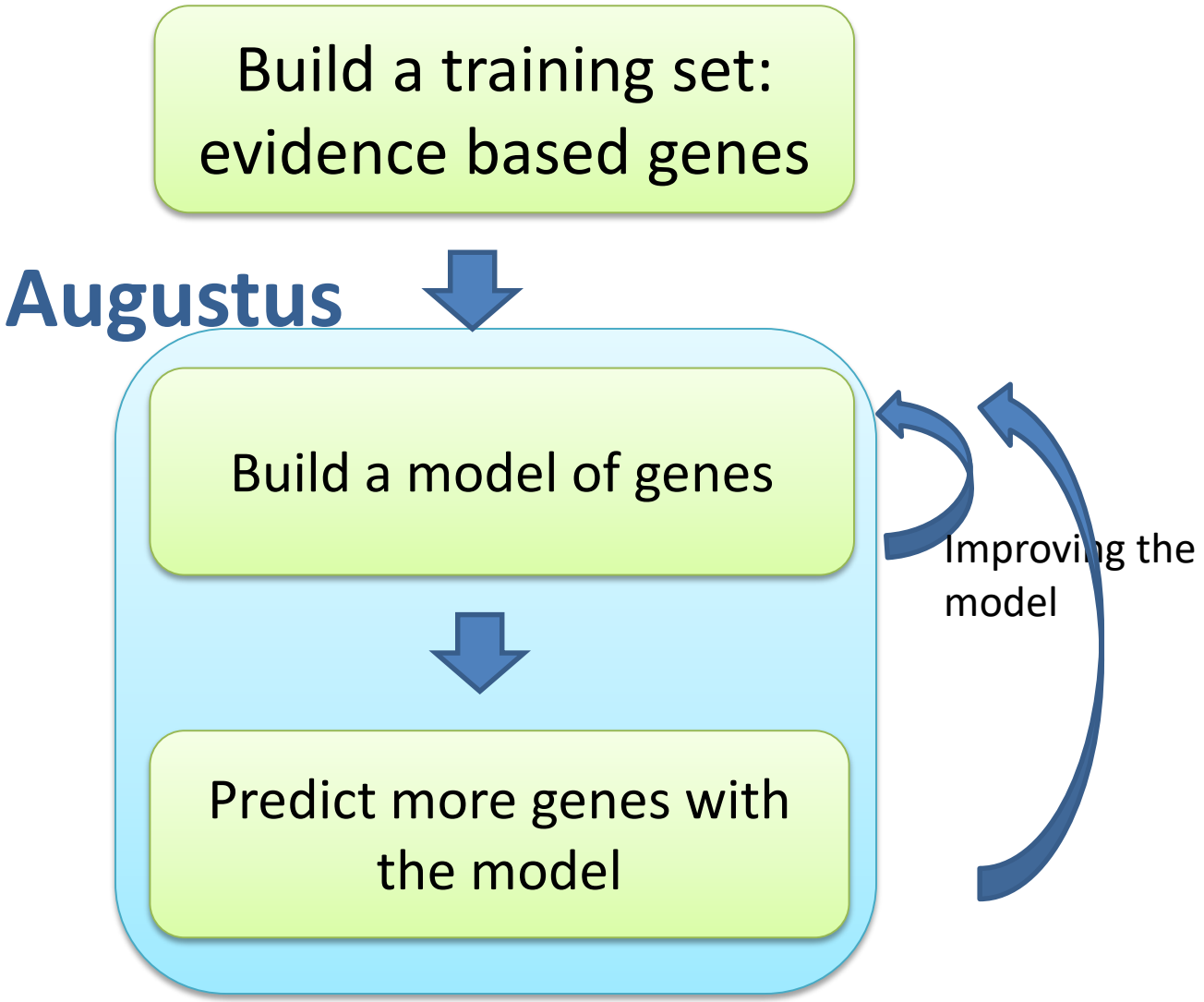
Improving the model

Model the base composition of

- Coding regions
- Introns
- Splicing donor and acceptor
- Translation start & end
- Transcription start & end
- Lengths of exons and introns
- Number of exons per gene



Gene predictor **Augustus** normally run through a pipeline



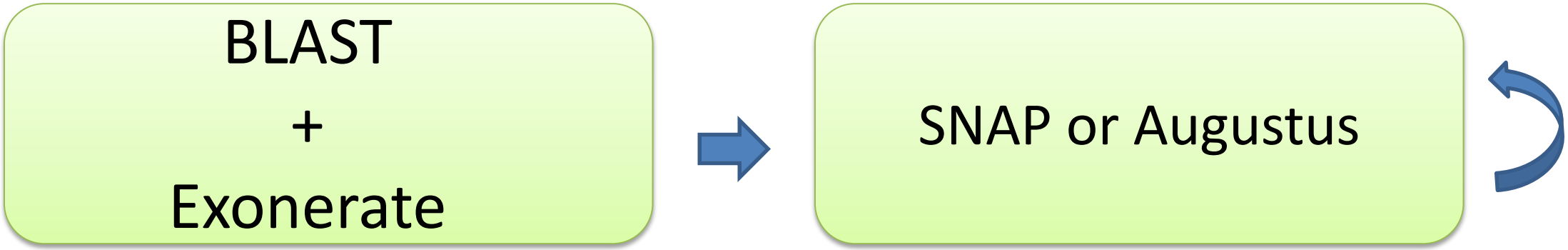
Maker v3

Braker v2

Maker pipeline

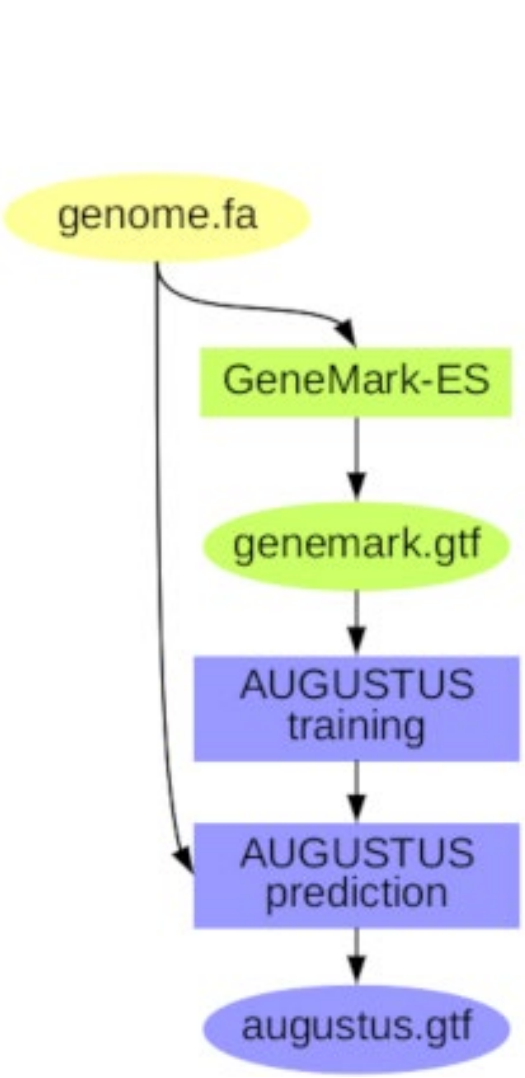
- Produce a GFF3 file by integrating known transcripts and proteins

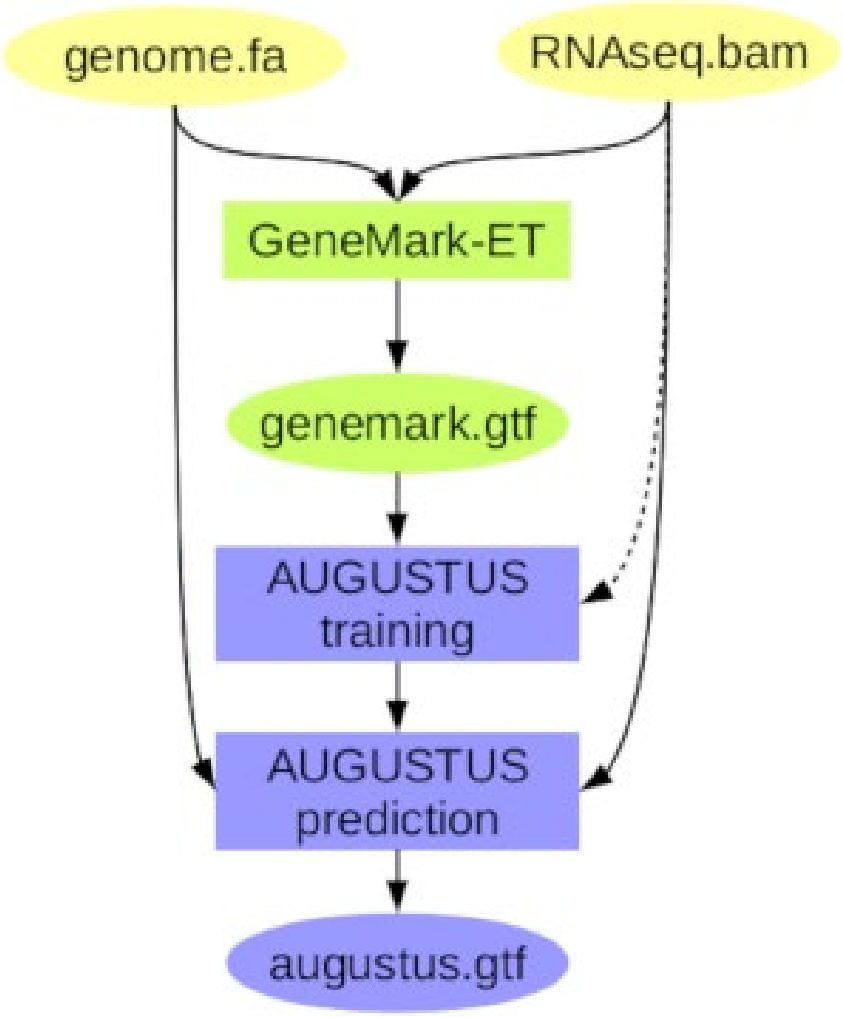
- Use GFF3 file to train prediction models;
- *Ab initio* gene prediction

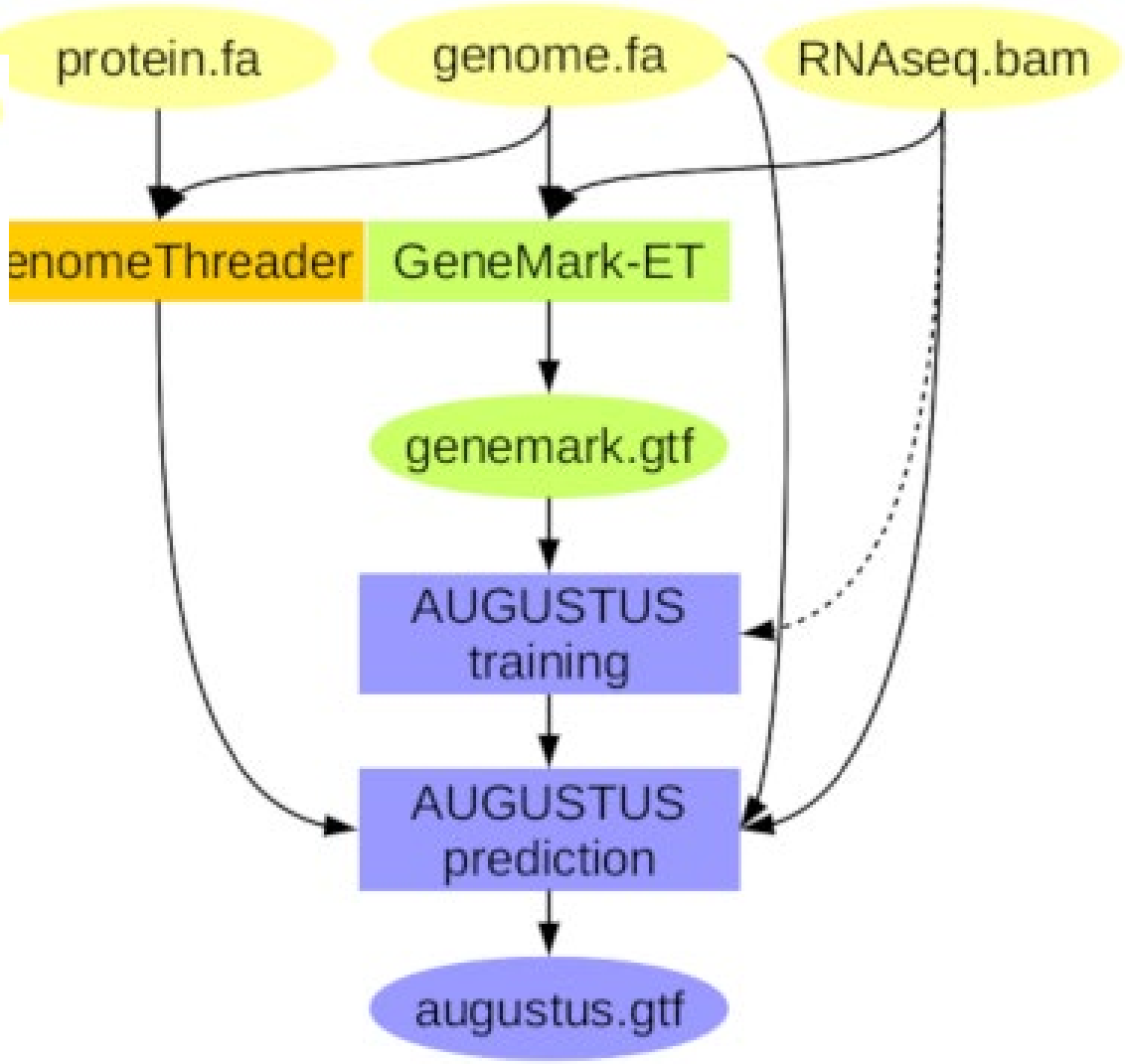


Braker2 pipelines

Use RNA and protein sequences from same individual, or at least same species as the genome







On BioHPC, run Braker2 pipeline through Docker

Pull the biohpc/braker2 Docker image from the Dockerhub repository

```
docker1 pull biohpc/braker2
```

All input files in "/workdir/$USER/project2", run command

```
docker1 run --rm \
    -v /workdir/$USER/project2:/data  \
    biohpc/braker2 \
    sh -c ". /root/source.sh; braker.pl --species=mysp --genome=genome.fa.masked
--bam=RNA.sorted.bam --softmasking --gff3 --cores=20"
```

Map "/workdir/$USER/project2" to "/data" in Docker.

"braker.pl" command is executed inside Docker.

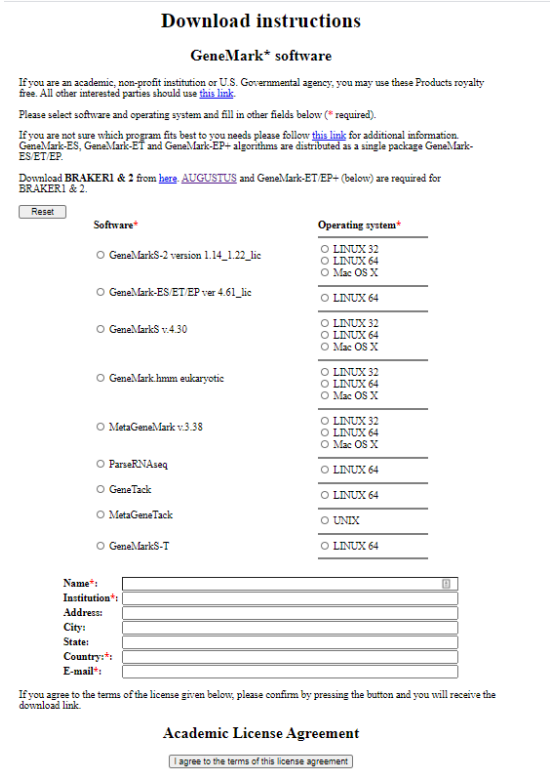
Output files are in "/workdir/$USER/project2".

About the Docker image biohpc/braker2

- All software required by the pipeline (except GeneMark) are install in /root/ directory; (Augustus: 3.3.3+ 10/16/2020; Braker2: 2.1.5)

- You need to supply GeneMarker software and license, keep in same directory as input files, and use“-v” to mount to /data directory in Docker.

- Run script “/root/source.sh” to set PATH in Docker.



Download instructions

GeneMark* software

If you are an academic, non-profit institution or U.S. Governmental agency, you may use these Products royalty free. All other interested parties should use this link.

Please select software and operating system and fill in other fields below (* required).

If you are not sure which program fits best to you needs please follow this link for additional information. GeneMark-ES, GeneMark-ET and GeneMark-EP+ algorithms are distributed as a single package GeneMark-ES/ET/EP.

Download **BRAKER1 & 2** from here. AUGUSTUS and GeneMark-ET/EP+ (below) are required for BRAKER1 & 2.

Reset

Software*	Operating system*
○ GeneMarkS-2 version 1.14_1.22_lic	○ LINUX 32 ○ LINUX 64 ○ Mac OS X
○ GeneMark-ES/ET/EP ver 4.61_lic	○ LINUX 64
○ GeneMarkS v.4.30	○ LINUX 32 ○ LINUX 64 ○ Mac OS X
○ GeneMark.hmm eukaryotic	○ LINUX 32 ○ LINUX 64 ○ Mac OS X
○ MetaGeneMark v.3.38	○ LINUX 32 ○ LINUX 64 ○ Mac OS X
○ ParseRNAseq	○ LINUX 64
○ GeneTack	○ LINUX 64
○ MetaGeneTack	○ UNIX
○ GeneMarkS-T	○ LINUX 64

Name*:
Institution*:
Address:
City:
State:
Country:*:
E-mail*:

If you agree to the terms of the license given below, please confirm by pressing the button and you will receive the download link.

Academic License Agreement

I agree to the terms of this license agreement

Fill out GeneMark registration form and download both the software and license file.

EVidenceModeler (EVM)

- Integrate multiple annotation results in GFF3 formats;
- A weight file is required

ABINITIO_PREDICTION	augustus	1
ABINITIO_PREDICTION	twinscan	1
ABINITIO_PREDICTION	glimmerHMM	1
PROTEIN	spliced_protein_alignments	1
PROTEIN	genewise_protein_alignments	5
TRANSCRIPT	spliced_transcript_alignments	1
TRANSCRIPT	PASA_transcript_assemblies	10

Run EVM in parallel

Partition the genome into chunks

```
partition_EVM_inputs.pl \
  --segmentSize 1000000 \
  --overlapSize 200000 \
```

--segmentSize < 1 Mb.
--overlapSize at least two standard deviations greater than the gene length

Create a batch command list

```
write_EVM_commands.pl
```

Run in parallel
(-j 40: 40 jobs simultaneously)

```
parallel -j 40 < commands.list
```

Merge results

EVM.all.gff3

Assessing annotation completeness using BUSCO

BUSCO mode:

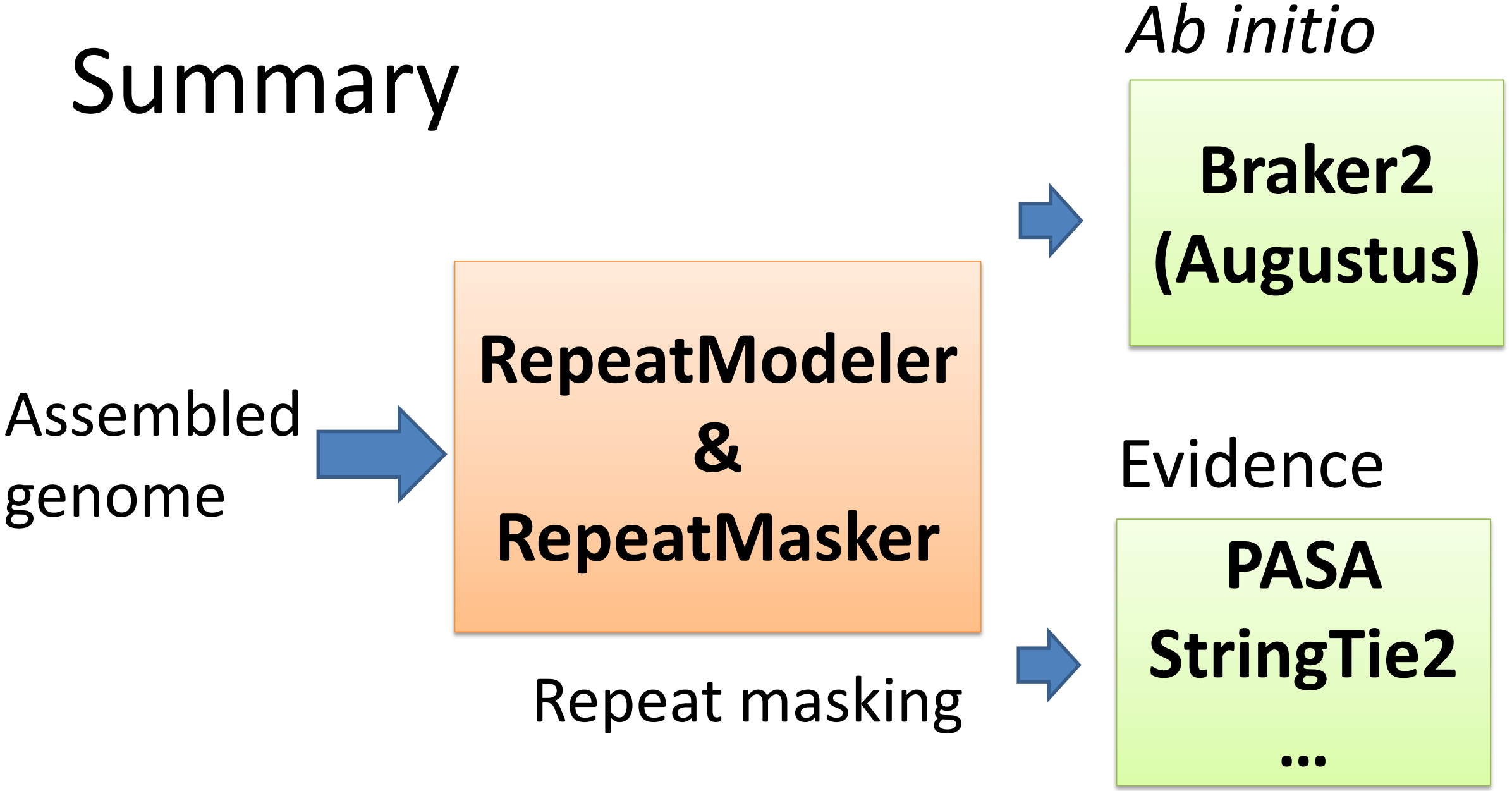
- Genome
- Transcript
- Protein

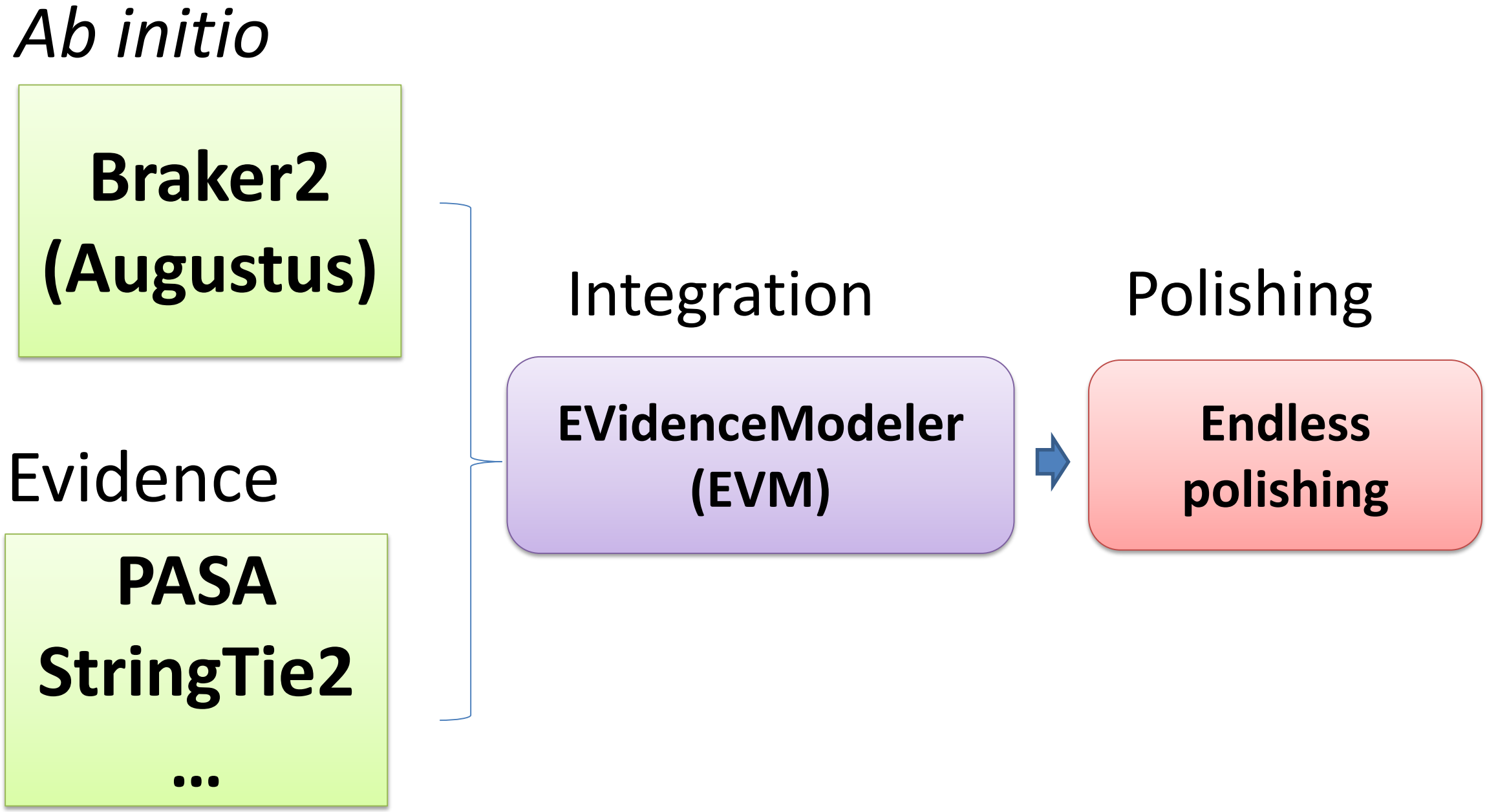
Assess annotation quality

For example:

Prediction using Model 1
C:71.3%[S:68.9%,D:2.4%],F:17.4%,M:11.3%,n:3354
Prediction using Model 2
C:89.6%[S:88.2%,D:1.4%],F:2.9%,M:7.5%,n:3354

Summary


Ab initio
Braker2
(Augustus)
Assembled
genome
RepeatModeler
&
RepeatMasker
Repeat masking
Evidence
PASA
StringTie2
…


Ab initio
Braker2 (Augustus)
Evidence
PASA
StringTie2
…
Integration
EVidenceModeler (EVM)
Polishing
Endless polishing

Ab initio prediction is not always reliable

Prediction accuracy of Augustus (Sn: sensitivity; Sp: specifity)

Nucleotide		Exon		Transcript	
Sn	Sp	Sn	Sp	Sn	Sp
97.0	89.0	86.1	72.6	50.1	28.7

Use the gene prediction results with caution

RNA-seq analysis

Regular RNA-seq: very good;

3’ RNA-seq: not good;

SNP annotation: OK

Make construct for transgenics: not good

Annotation result

GFF3 file format

chr	source	type	start	end	score	strand	frame	attributes
1	AUGUSTUS	gene	1	9276	0.29	+	.	ID=g1;
1	AUGUSTUS	mRNA	1	9276	0.29	+	.	ID=g1.t1;Parent=g1;
1	AUGUSTUS	CDS	7574	8116	0.97	+	1	ID=g1.t1.CDS1;Parent=g1.t1;
1	AUGUSTUS	exon	7574	8116	.	+	.	ID=g1.t1.exon1;Parent=g1.t1;
1	AUGUSTUS	CDS	8229	8589	0.37	+	1	ID=g1.t1.CDS2;Parent=g1.t1;
1	AUGUSTUS	exon	8229	8589	.	+	.	ID=g1.t1.exon2;Parent=g1.t1;
1	AUGUSTUS	CDS	8668	9276	0.84	+	0	ID=g1.t1.CDS3;Parent=g1.t1;
1	AUGUSTUS	exon	8668	9276	.	+	.	ID=g1.t1.exon3;Parent=g1.t1;

Required: ID and Parent

Annotation result

GTF file format

								Atribute
1	AUGUSTUS	exon	7574	8116	.	+	.	transcript_id "g1.t1"; gene_id "g1";
1	AUGUSTUS	intron	8117	8228	0.37	+	.	transcript_id "g1.t1"; gene_id "g1";
1	AUGUSTUS	CDS	8229	8589	0.37	+	1	transcript_id "g1.t1"; gene_id "g1";
1	AUGUSTUS	exon	8229	8589	.	+	.	transcript_id "g1.t1"; gene_id "g1";
1	AUGUSTUS	intron	8590	8667	0.84	+	.	transcript_id "g1.t1"; gene_id "g1";
1	AUGUSTUS	CDS	8668	9276	0.84	+	0	transcript_id "g1.t1"; gene_id "g1";
1	AUGUSTUS	exon	8668	9276	.	+	.	transcript_id "g1.t1"; gene_id "g1";

“gffread” from cufflinks

- **Converting GFF3 to GTF**

```
gffread -T -o  output.gtf  input.gff3
```

GFF3 -> GTF

```
gffread -o  output.gff3  input.gtf
```

GTF -> GFF3

- **Extract transcript/CDS sequence into fasta**

```
gffread -g genome.fa  -w transcript.fasta input.gff3
```

Transcript

```
gffread -g genome.fa  -x CDS.fasta input.gff3
```

CDS

- **Extract protein sequence into fasta**

```
gffread -g genome.fa -y protein.fasta input.gff3
```

Visualization & manual curation (Apollo, IGV)

Braker2 dependencies:

Supported software versions

At the time of release, this BRAKER version was tested with:

- AUGUSTUS 3.3.4 [F2]
- GeneMark-ES/ET/EP 4.59_lic
- BAMTOOLS 2.5.1[R5]
- SAMTOOLS 1.7-4-g93586ed[R6]
- ProtHint 2.5.0
- GenomeThreader 1.7.0[R7]
- Spaln 2.3.3d [R8, R9, R10, F3]
- (Exonerate 2.2.0 [R11])[F3]
- NCBI BLAST+ 2.2.31+ [R12, R13]
- DIAMOND 0.9.24
- cdbfasta 0.99
- cdbyank 0.981
- GUSHR 1.0.0

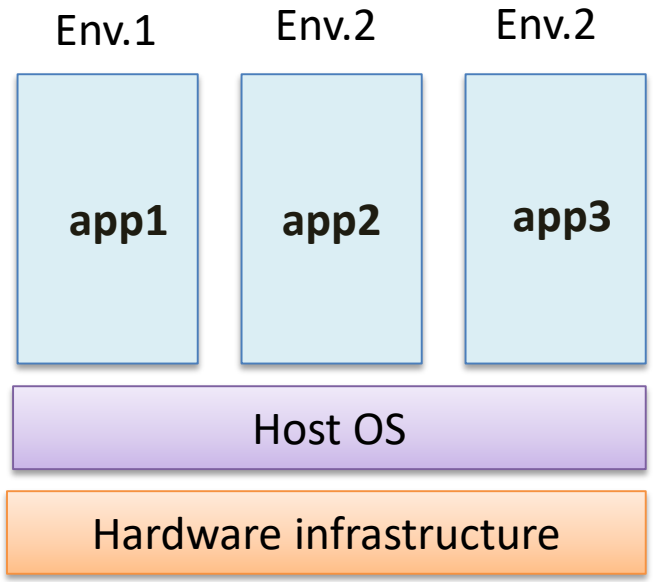
When installing a pipeline

- **Each pipeline should be fully contained in its own environment**
- **Use the same version as described;**

Three ways of isolation

Conda vs **Docker** vs **VM**

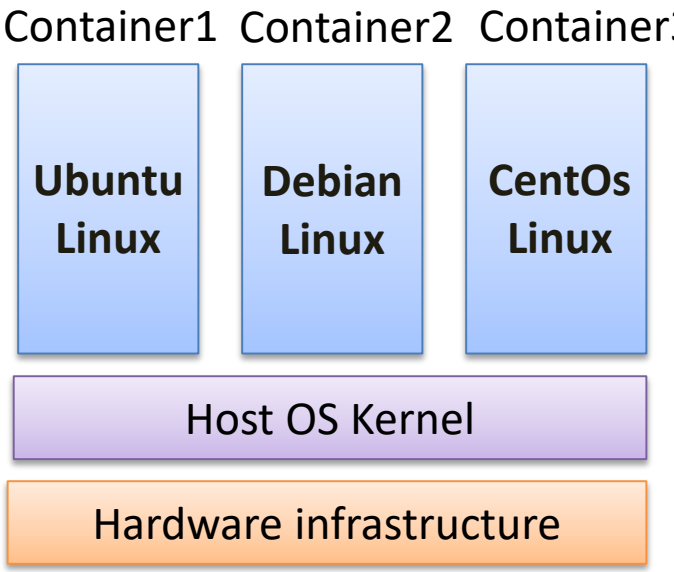
Conda environments



- Modified $PATH;
- File system not isolated;

Drawback: not fully contained

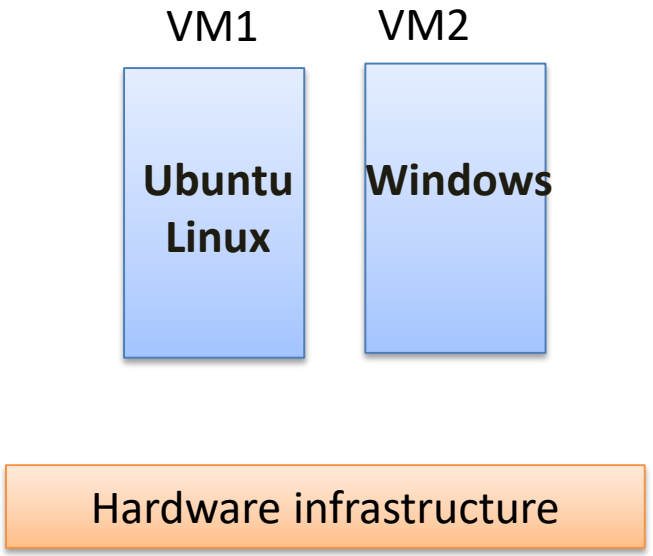
Docker containers



- File system and network port are contained;

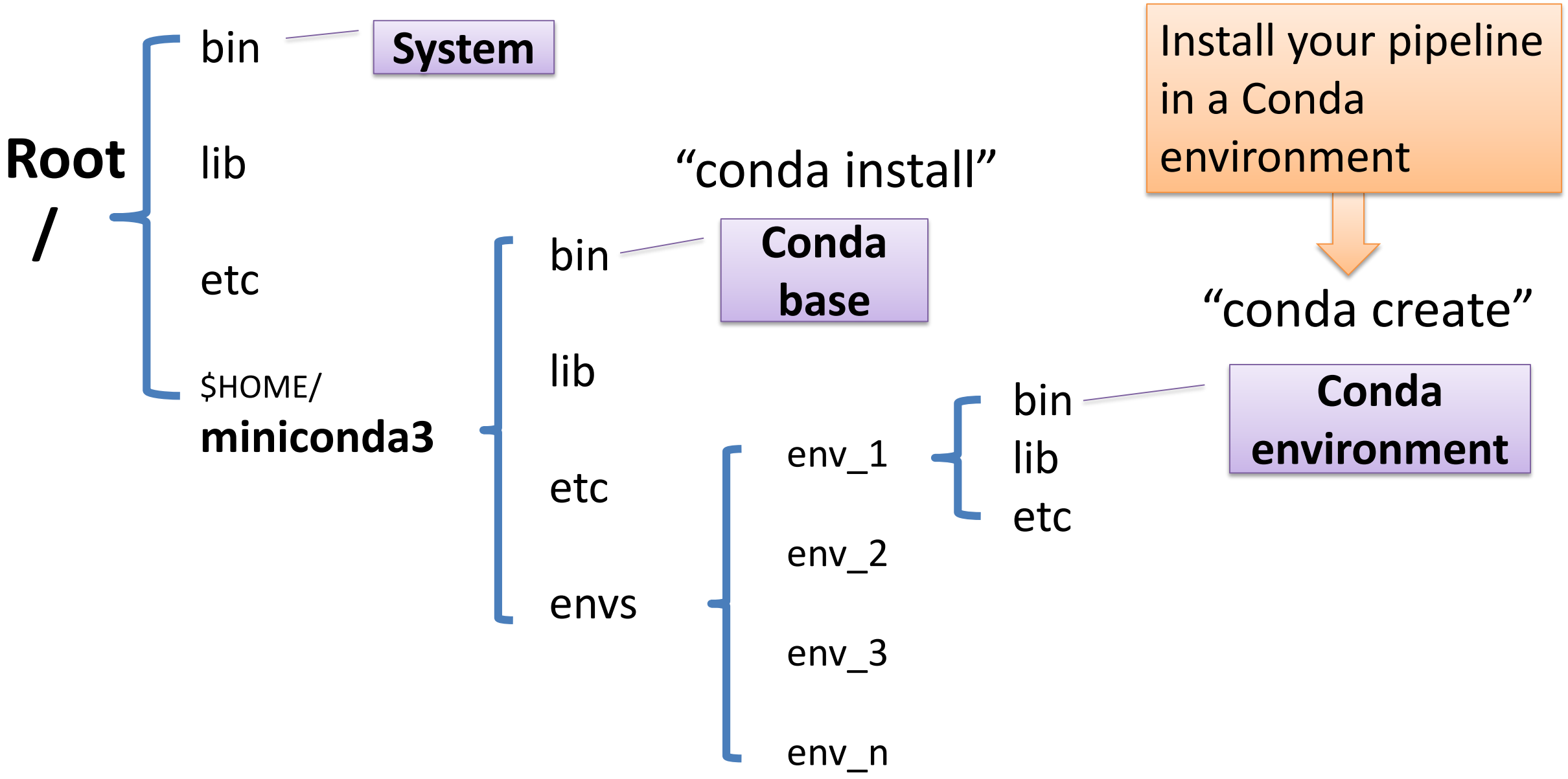
Drawback: require root

Virtual Machine



Drawback: “VMWare tax”

Software can be installed at three different levels, with Conda



When setting up a pipeline for your annotation project, save a Docker image, do not rely on Docker file

- Theoretically, you can re-build the same image with the same Docker file. In practice, you cannot.

Command to save a docker instance to a image file

```
docker1 export -o braker2.tar 104db52110d2
```