Genome Annotation

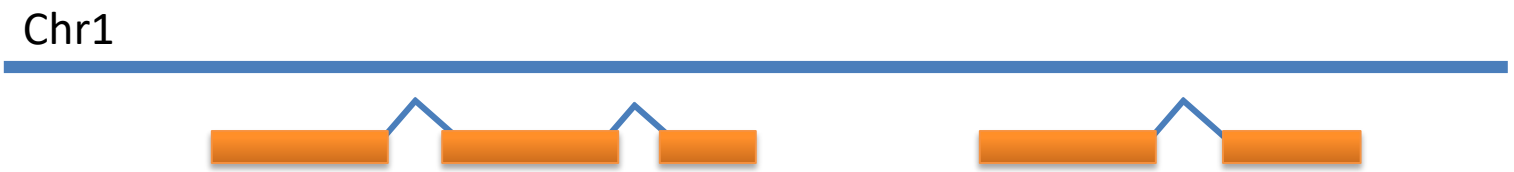
Qi Sun

Bioinformatics Facility

Cornell University

Two Steps in Genome Annotation

1. Predict genes on the genome



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

Two Steps in Genome Annotation

2. Predict functions of each gene

Gene ID	Gene description
GRMZM2G002950	Putative leucine-rich repeat receptor-like protein kinase family
GRMZM2G006470	Uncharacterized protein
GRMZM2G014376	Shikimate dehydrogenase; Uncharacterized protein
GRMZM2G015238	Prolyl endopeptidase
GRMZM2G022283	Uncharacterized protein

Week 1. Gene prediction

Genome Assembly

Evidence based

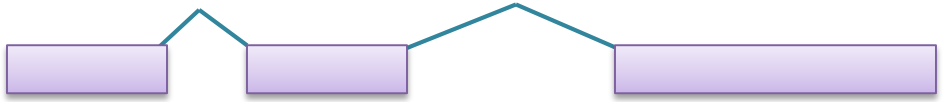
RNA-seq

Known Proteins

Ab initio gene prediction

Species-specific model

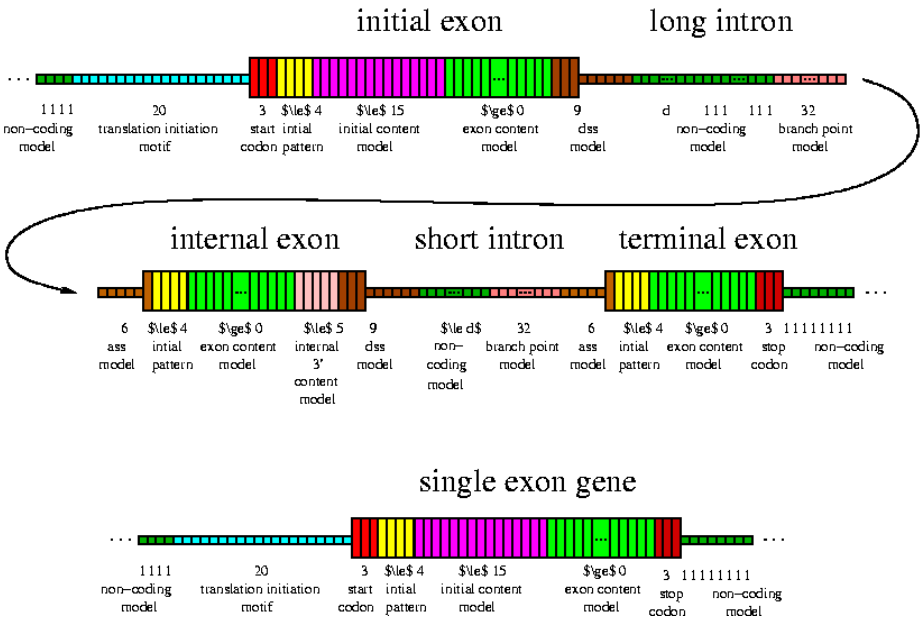
prediction



Ab initio gene prediction

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



Evidence vs *ab initio* gene prediction

Evidence based gene

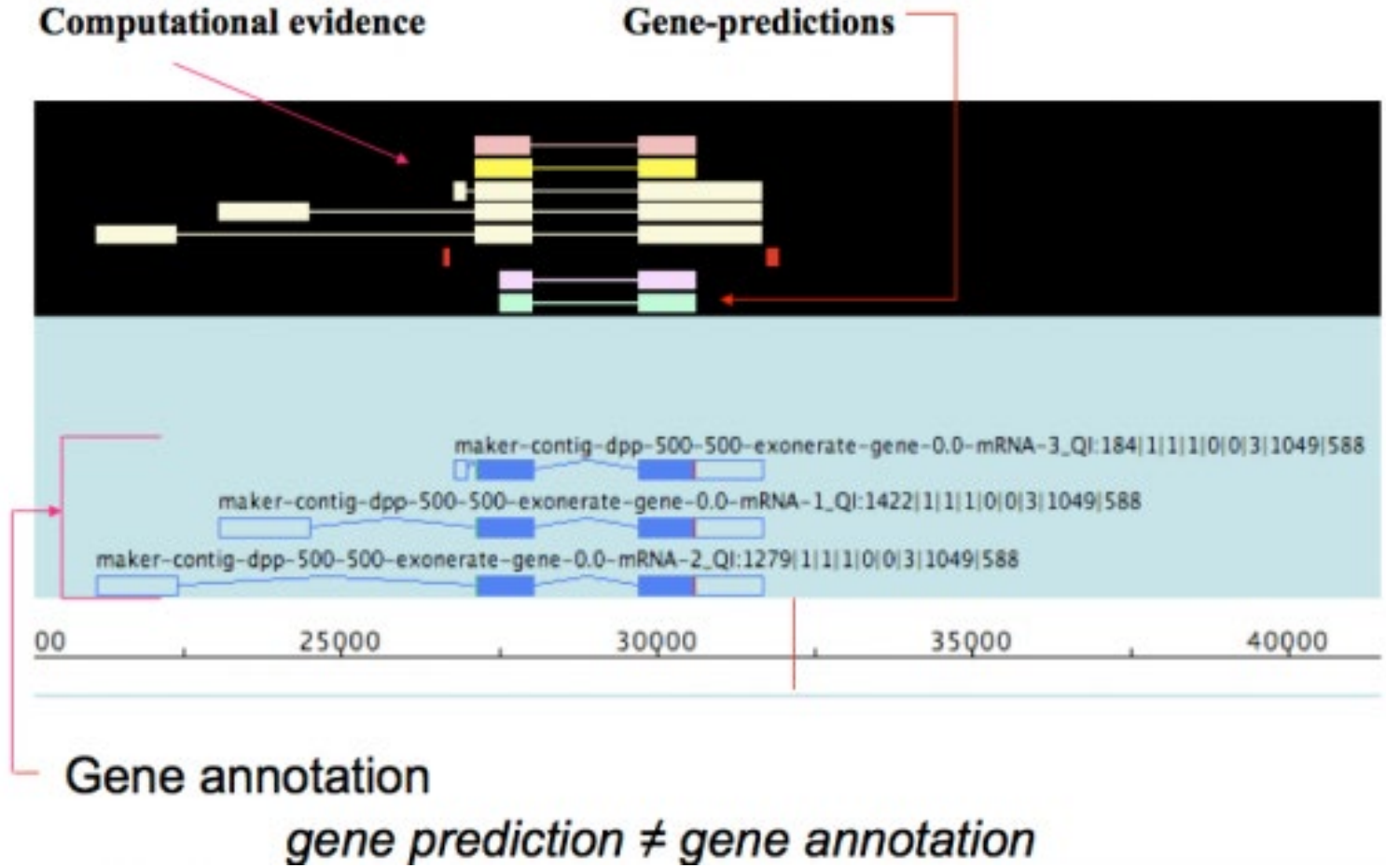
- More reliable;
- Not full length or missing for low-abundant genes;
- Might be contaminated with pre-mRNA

***Ab initio* gene prediction**

- Less reliable;
- Transcription start and end not accurate;

The MAKER Pipeline

Evidence supported gene + *Ab initio* predictions



Genome Annotation Tools

Eukaryotic

Popular pipelines

(Evidence + *Ab Initio*)

- Maker
- Braker

Ab initio

- Augustus
- GeneMark
- SNAP

Genome Annotation Tools

Prokaryotic

Online Services

1. RAST
2. NCBI

Standalone pipelines

1. NCBI PGAP
2. Prokka

Input files for MAKER:

Referred to as **EST** in Maker

Genome Sequence

Transcriptome

- **RNA-seq**
- **Relatedspecies**

Protein sequences

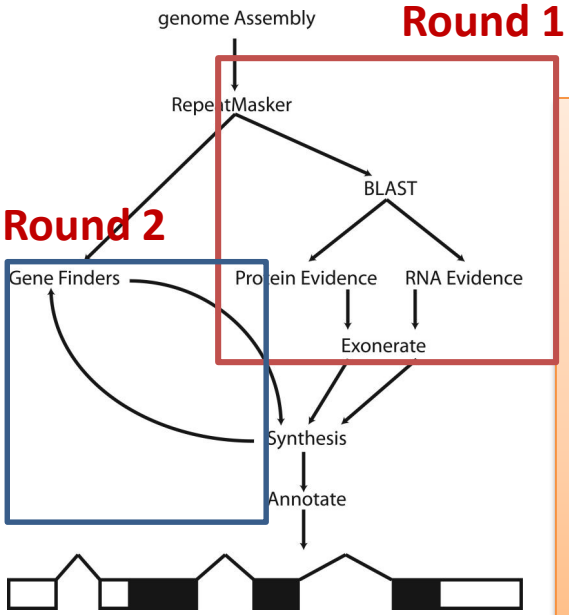
- Related species

Repeat Database

Run Maker

Command: **maker** -base output_rnd1

- Use control files located in the same directory to define what to do;
- Output results in directory output_rnd1



Round 1 Control file

genome=pyu_contig.fasta

est2genome=1
protein2genome=1

est=pyu_est.fasta
protein=sp_protein.fasta

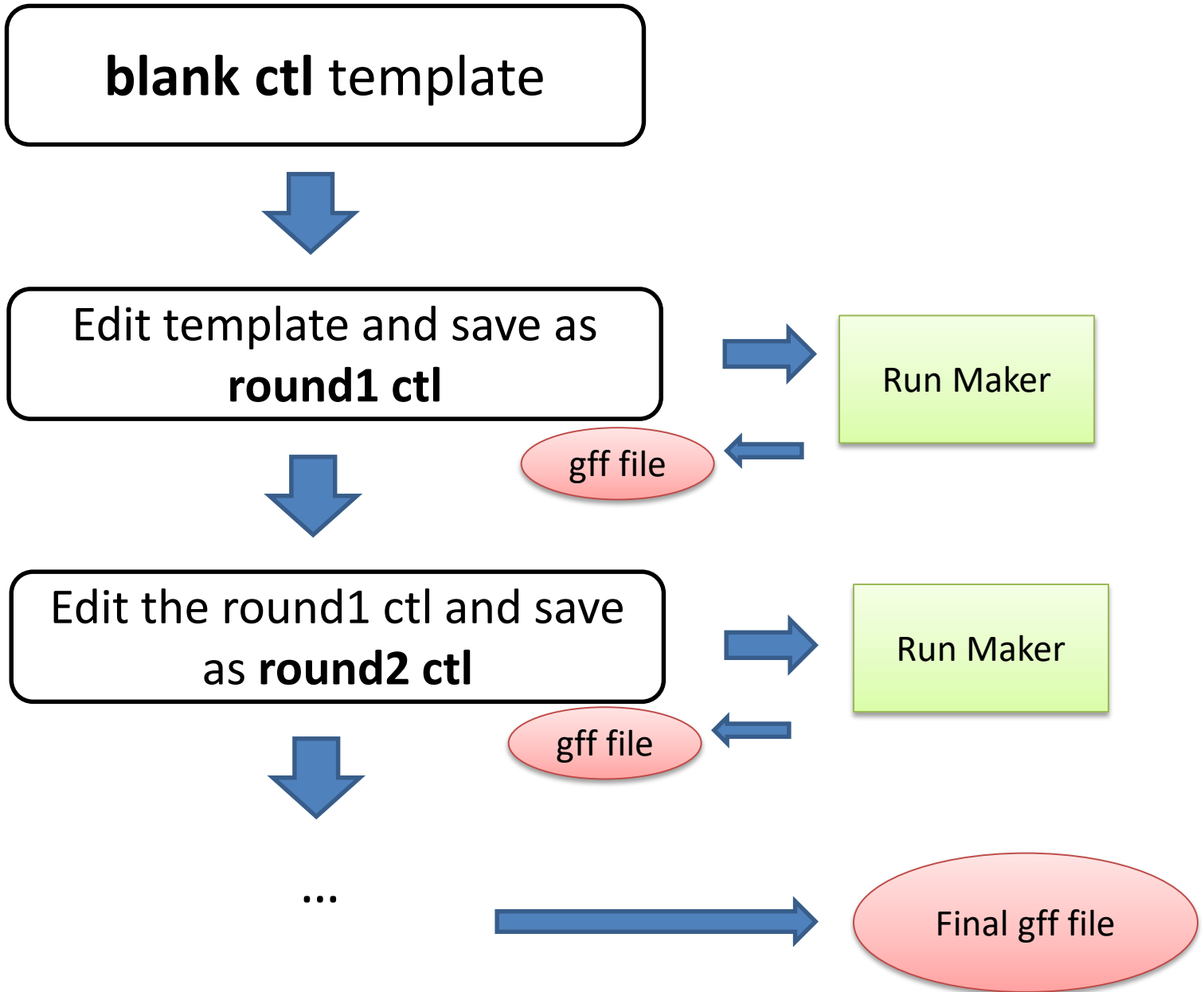
Round 2 control file

maker_gff=pyu_rnd1.all.gff

est2genome=0
protein2genome=0

est_pass=1
protein_pass=1
snaphmm=pyu1.hmm

Control files



Repeat Masking in round 1

Software: RepeatMasker

Simple repeats: E.g. “AAAAA…AAA”

Complex repeats: E.g. retroelements

- **Prebuilt DB:** Dfam

  Repbase (license required)

- **Custom DB:** Build with RepeatModeler

Soft vs Hard Masking

ACCAGAGTACTACGATAC TTTTTTTTTTTTTTTTT ACCAAACGTACAA

Soft masking:
ACCAGAGTACTACGATAC ttttttttttttttttt ACCAAACGTACAA

Hard masking:
ACCAGAGTACTACGATAC NNNNNNNNNNNNNNNNN ACCAAACGTACAA

Maker behavior for masking:

softmask=1

- Soft masking of simple repeats
- BLAST: masked regions not for seeding, only for extension;

rm_gff=repeats.gff

- Hardmask external gff file.

Run **RepeatMasker**

Within Maker pipeline (ctl file):

```
#-----Repeat Masking (leave values blank to skip repeat masking)
model_org=simple
rmlib=repeat_proteinte_proteins.fasta
rm_gff=
prok_rm=0
```

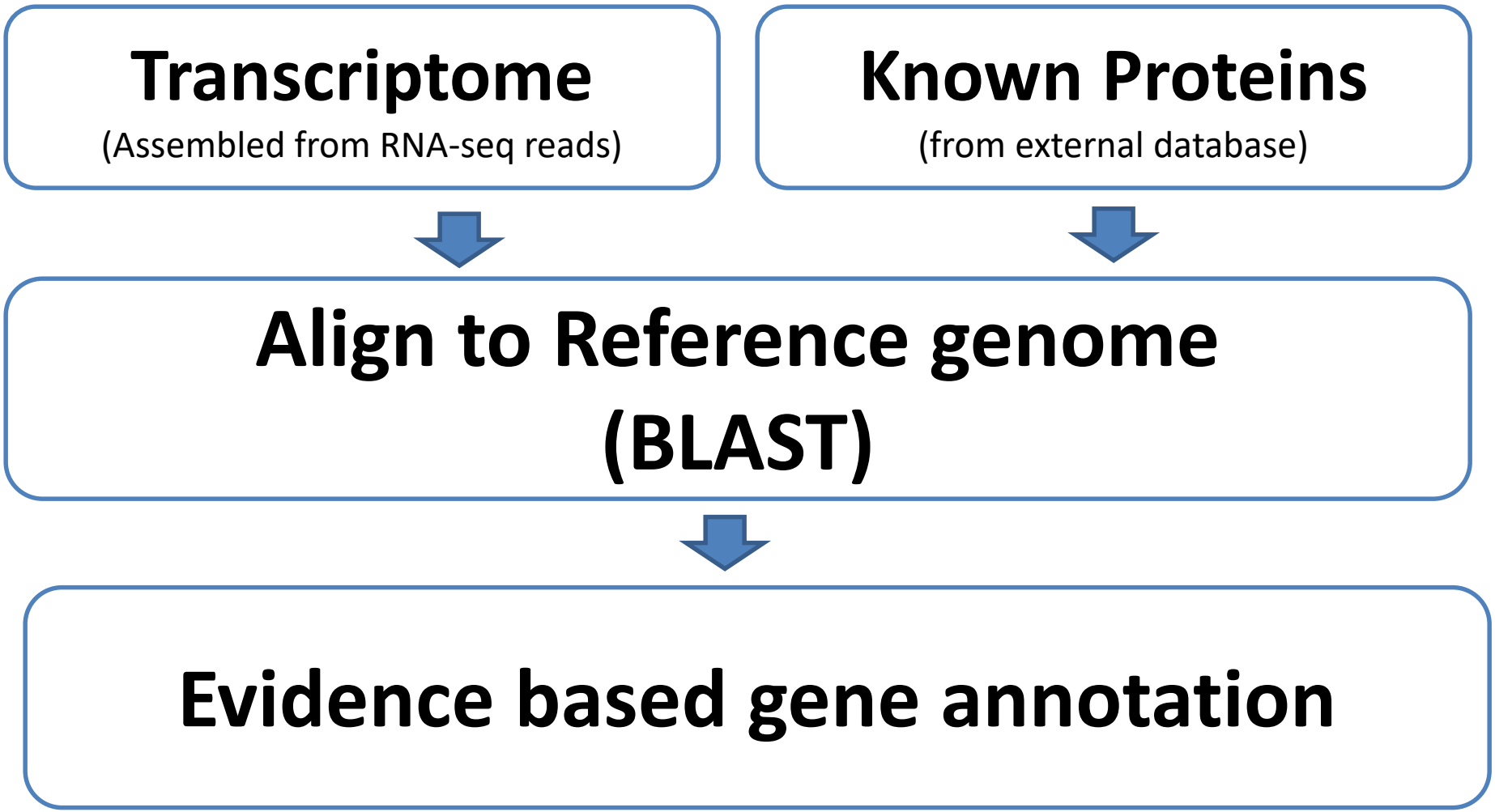
Outside Maker pipeline:

- *De novo* (custom repeat database from Repeat Modeller)
- Dfam (pre-built repeat database)
- Create a gff file to feed into Maker pipeline

https://gist.github.com/darencard/bb1001ac1532dd4225b030cf0cd61ce2

Round 1 of Maker

- Evidence based Gene Annotation

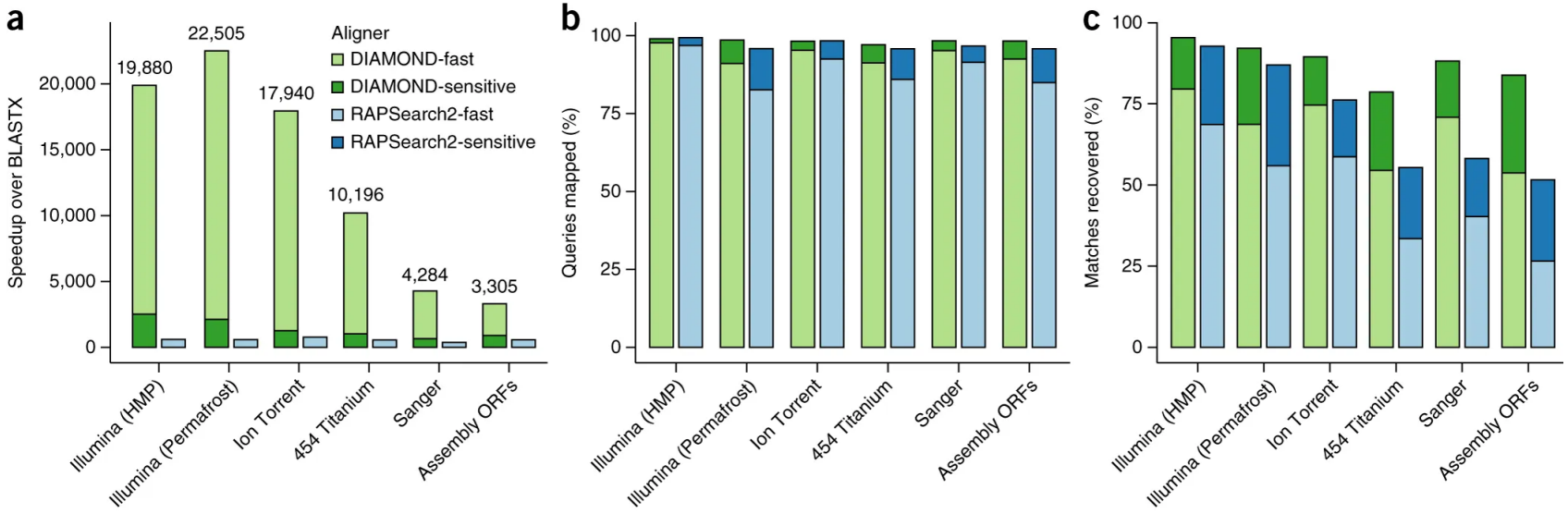


NCBI BLAST programs

- **blastn** nucleotide query vs. nucleotide database
- **blastp** protein query vs. protein database
- **blastx** nucleotide query vs. protein database
- **tblastn** protein query vs. translated nucleotide database
- **tblastx** translated query vs. translated database

DIAMOND

A much faster alternative to BLAST



Diamond-sensitive vs BLASTX

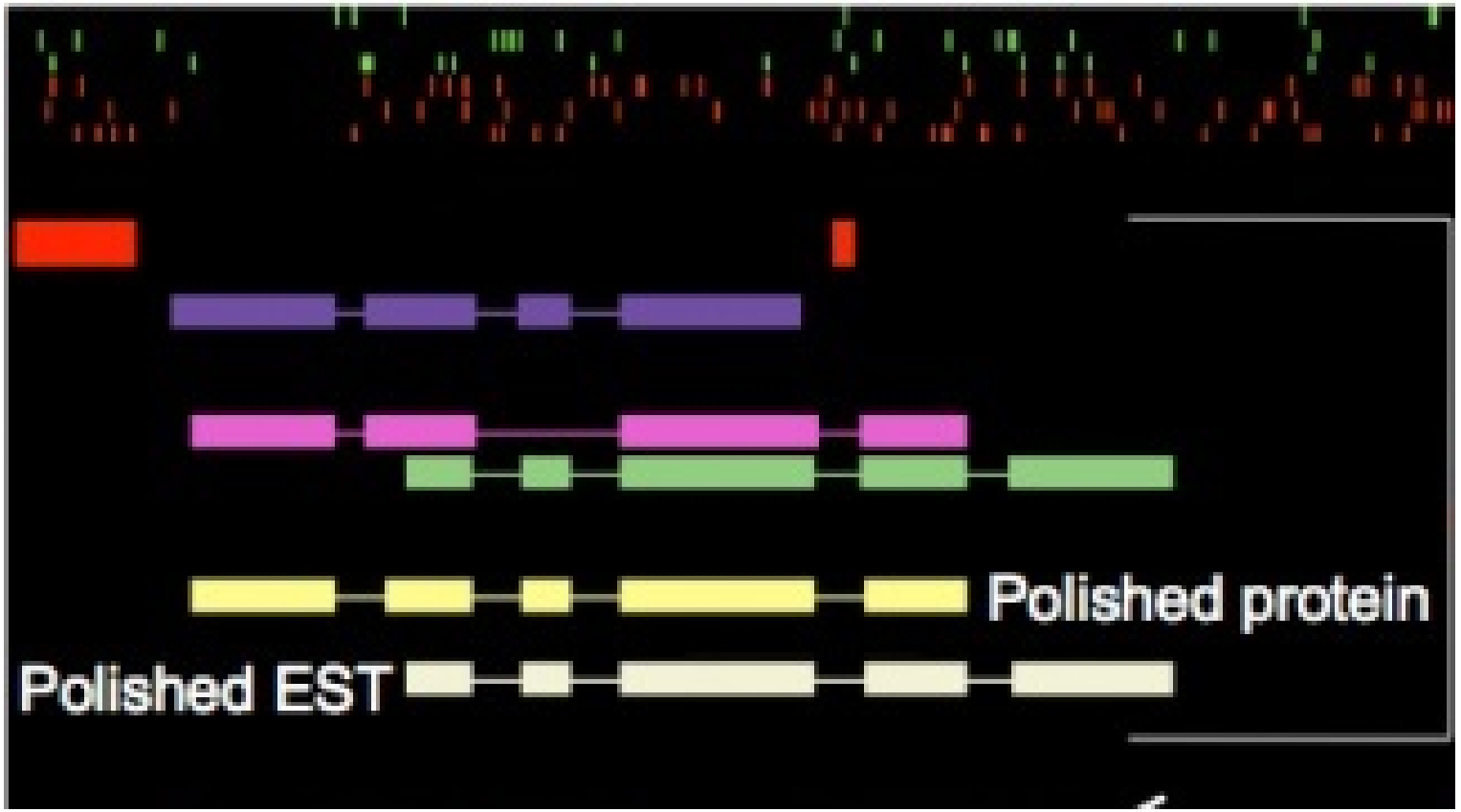
2,000 times faster;

Aligning 99% of reads

Obtaining over 92% of targets

* Supported in BRAKER annotation pipeline

Exonerate to polish BLAST hits



Round 1 control file:

Specify the transcript and protein sequences to be used

#-----EST Evidence (for best results provide a file for at least one)

est=pyu_est.fasta

#-----Protein Homology Evidence (for best results provide a file for at least one)

protein=sp_protein.fasta #protein sequence file in fasta format (i.e. from mutiple oransisms)

#-----Gene Prediction

est2genome=1 #infer gene predictions directly from ESTs, 1 = yes, 0 = no

protein2genome=1 #infer predictions from protein homology, 1 = yes, 0 = no

Round 2 of Maker

- Model construction and prediction

Training set:

- Evidence based gene models

Procedures:

- Build HMM model with AUGUSTUS or SNAP
- Use the HMM for prediction

Round 2 steps

Build model outside of Maker:

Make a GFF file from round 1

```
gff3_merge -d pyu_rnd1_master_datastore_index.log
maker2zff -l 50 -x 0.5 pyu_rnd1.all.gff
```

Build an HMM model

Augustus or SNAP

Back to Maker.

***Ab initio* Prediction using HMM model**

genome=dpp_contig.fasta #genome sequence

…

snaphmm=pyu1.hmm

est2genome=0

protein2genome=0

GFF file with predicted genes

```
f
  es.mod
```

Maker supported prediction software

#AUGUSTUS

```
augustus_species=thomas_1.hmm
```

#SNAP

```
gmhmm=thomas_1.hmm
```

#GMHMM

```
snaphmm=es.mod
```

#FGENESH

```
fgenesh_par_file= #Fgenesh parameter file
```

Include these lines in ctl file (2nd round)

Starting gff file

maker_gff=pyu_rnd1.all.gff # round 1 GFF

Do not run BLAST & Repeatmasking, pass on information from previous run (set **est2genome** and **est2genome** to 0**)**

est2genome=0 #do not run EST alignment
protein2genome=0 #do not run protein alignment

est_pass=1 #pass on EST alignment from round 1 GFF
protein_pass=1 #pass on protein alignment from round 1 GFF
rm_pass=1 #pass on repeat alignment from round 1 GFF

Alternatively, if your gff file is way too big,

Parse out the est2genome annotation from previous Maker gff file

```
awk '{ if ($2 == "est2genome") print $0 }'  pyu_rnd1.all.gff  > est2genome.gff
```

In Ctl file:

```
maker_gff=

est_pass=0
est_gff=est2genome.gff
```

- Do the same for “protein2genome” and “repeat”

Round 3: Repeat round 2

Make a GFF file from round 2

```
gff3_merge -d pyu_rnd1_master_datastore_index.log
maker2zff -l 50 -x 0.5 pyu_rnd1.all.gff
```

Build an HMM model

Augustus or SNAP

Back to Maker. *Ab initio* Prediction using rnd 2 model

genome=dpp_contig.fasta #genome sequence

…

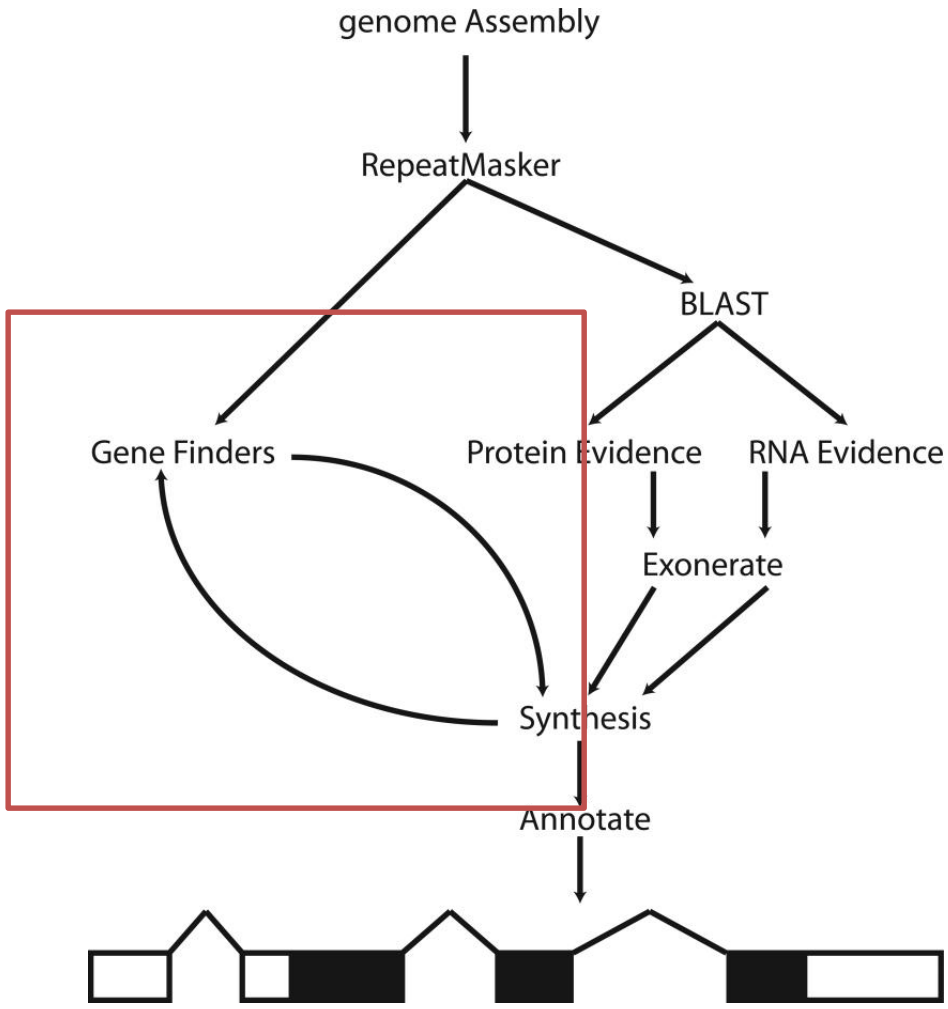
snaphmm=pyu2.hmm

est2genome=0

protein2genome=0

GFF file with predicted genes

Two or more iterations



A few other notes

1. Parallelization

Use machines with >=40 cores on BioHPC, and use all cores

Command:

```
mpiexec -n 40
```

Control File

```
cpus=1
```

2. Tmp directory

Control File

```
TMP=/workdir/$USER/tmp
```

* system default temporary directory /tmp is too small

3. Use **AUGUSTUS** and/or **SNAP** for prediction.

4. Custom gene, transcript and protein names with a script in Maker. (maker_map_ids)

5. On BioHPC, copy maker and repeatMasker to /workdir/$USER

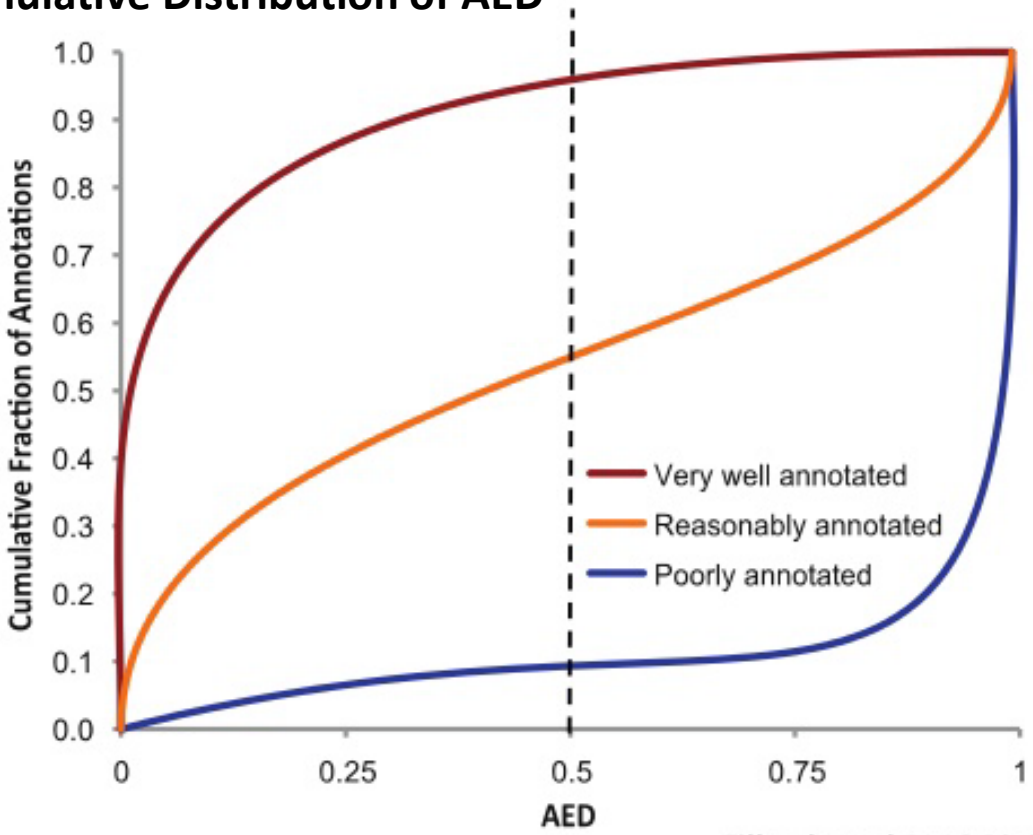
- These two directories contain large data files, better to keep them on /workdir

Avoid under-fitting and over-fitting:
Evaluate results with AED Score

(Annotation Edit Distance)

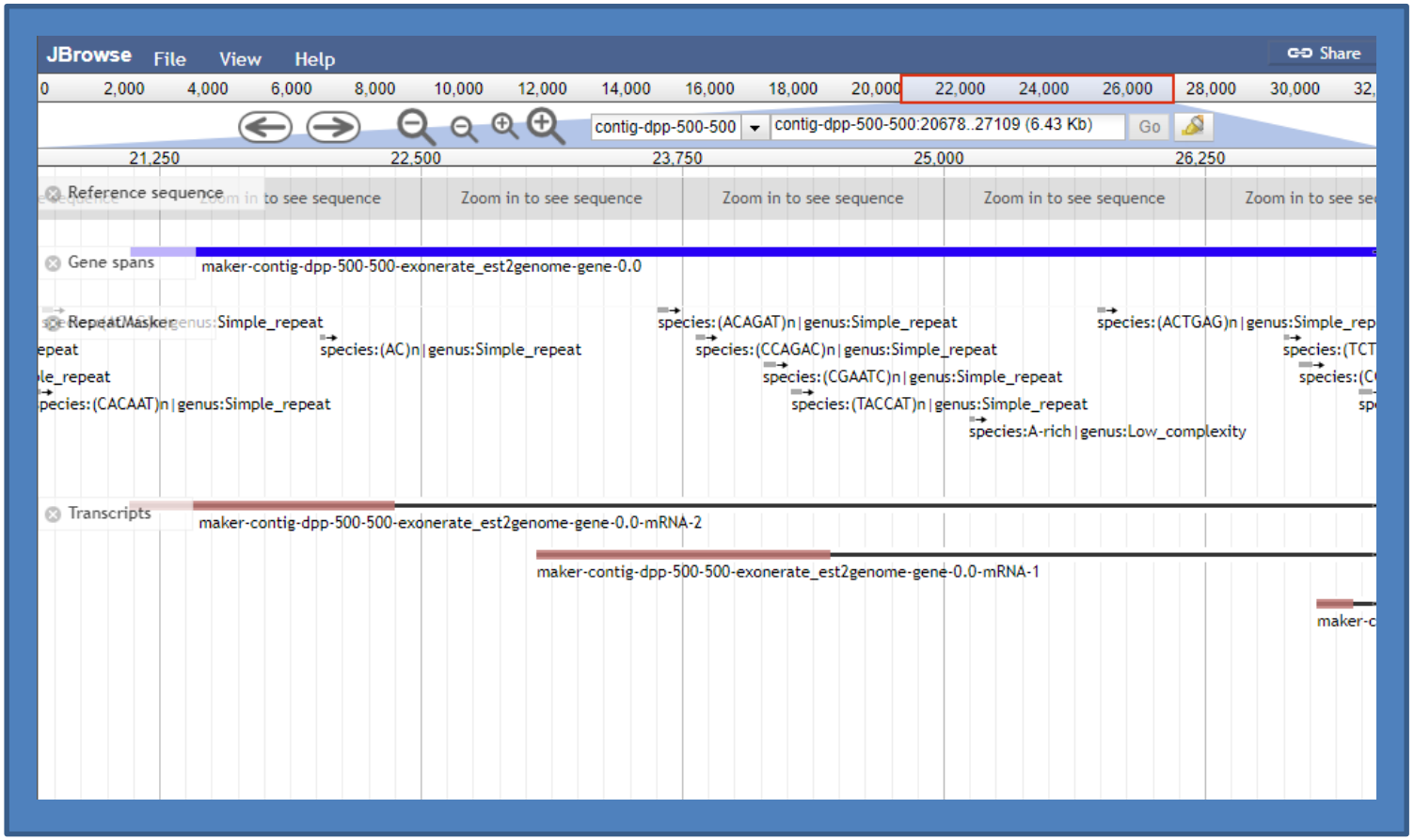
AED=0: Genes models of perfect concordance with the evidence;
AED=1: Genes models with no evidence support

Cumulative Distribution of AED



Eilbeck et al BMC Bioinformatics 2009

Visualization - JBrowse or IGV



JBrowse: https://biohpc.cornell.edu/lab/userguide.aspx?a=software&i=357#c

IGV: http://software.broadinstitute.org/software/igv/UserGuide

Can I trust MAKER annotation? NO

- Genes could be missing;
- Transcription start/end could be wrong;

Then, why do we do it?

- Useful for high-throughput experiments;
- Genome annotation is hard, and it's always a work in progress;

What to do if gene is missing?

Run TBLASTN with a closely related protein:

```
makeblastdb -in myGenome.fa -parse_seqids -dbtype nucl

tblastn -query myProtein.fa -db myGenome.fa -out output_file
```

Run PFAM on all ORF (slow, and exons only)

```
getorf -minsize 100 -sequence myGenome.fa -outseq myorf.fa

pfam_scan.pl -fasta myorf.fa –pfamB mydomain.hmm
```

Check with RNA-seq data