

An Introduction to the CPT Galaxy and WebApollo for Phage Whole Genome Annotation

Jason Gill

Department of Animal Science

Center for Phage Technology

Texas A&M University

Genome annotation tools

Fully automated annotation

- RAST/myRAST/RASTtk
 - <http://rast.nmpdr.org/>
- Prokka
 - <http://www.vicbioinformatics.com/software/prokka.shtml>
- NCBI Prokaryotic Pipeline
 - https://www.ncbi.nlm.nih.gov/genome/annotation_prok/

Semi-automated annotation

- DNA Master
 - <http://cobamide2.bio.pitt.edu/>
- CPT Galaxy/Apollo
 - <https://cpt.tamu.edu/galaxy-pub/>

Manual annotation / genome editors

- Sanger Artemis
 - <http://www.sanger.ac.uk/science/tools/artemis>
- Broad Argo
 - <https://archive.broadinstitute.org/annotation/argo/>

What is Galaxy?

- Galaxy is not an analysis tool itself
- Galaxy provides a **platform** for performing reproducible bioinformatics research
- Provides a Web-browser-based **user interface** for other command-line tools
- Provides a **history** of actions performed and tool outputs
- Allows users to chain operations together in **workflows** to perform complex analyses
- Galaxy is **open-source** (free) with an active user community

What is Galaxy?

- Galaxy can interface with any Linux command-line program via a short script called a “wrapper”
 - The wrapper presents input options to the user and passes these back to the invoked program (e.g., BLASTp, Glimmer3, etc.)
- Galaxy then keeps a record of that job, its inputs and outputs in a history
- Ultimately, Galaxy offers the power and flexibility of command-line Linux data analysis to the average biologist
- Maintains a record of work you’ve done, even years later

Why use Galaxy?



Why use Galaxy for phage?

- Galaxy is a popular and well-supported bioinformatic infrastructure
 - >125 Galaxy platforms available worldwide
- Automates tasks, retains inputs and outputs for future reference
- Is customizable for individual use cases, user retains control of the analysis
- **Education:** Students can see and interpret tool outputs, customize analyses
- **Novel or unusual genomes:** Analysis using customizable approaches
- **Beyond annotation:** comparative genomics, mutational analysis, phylogenetics

The Galaxy interface

Left panel: tools

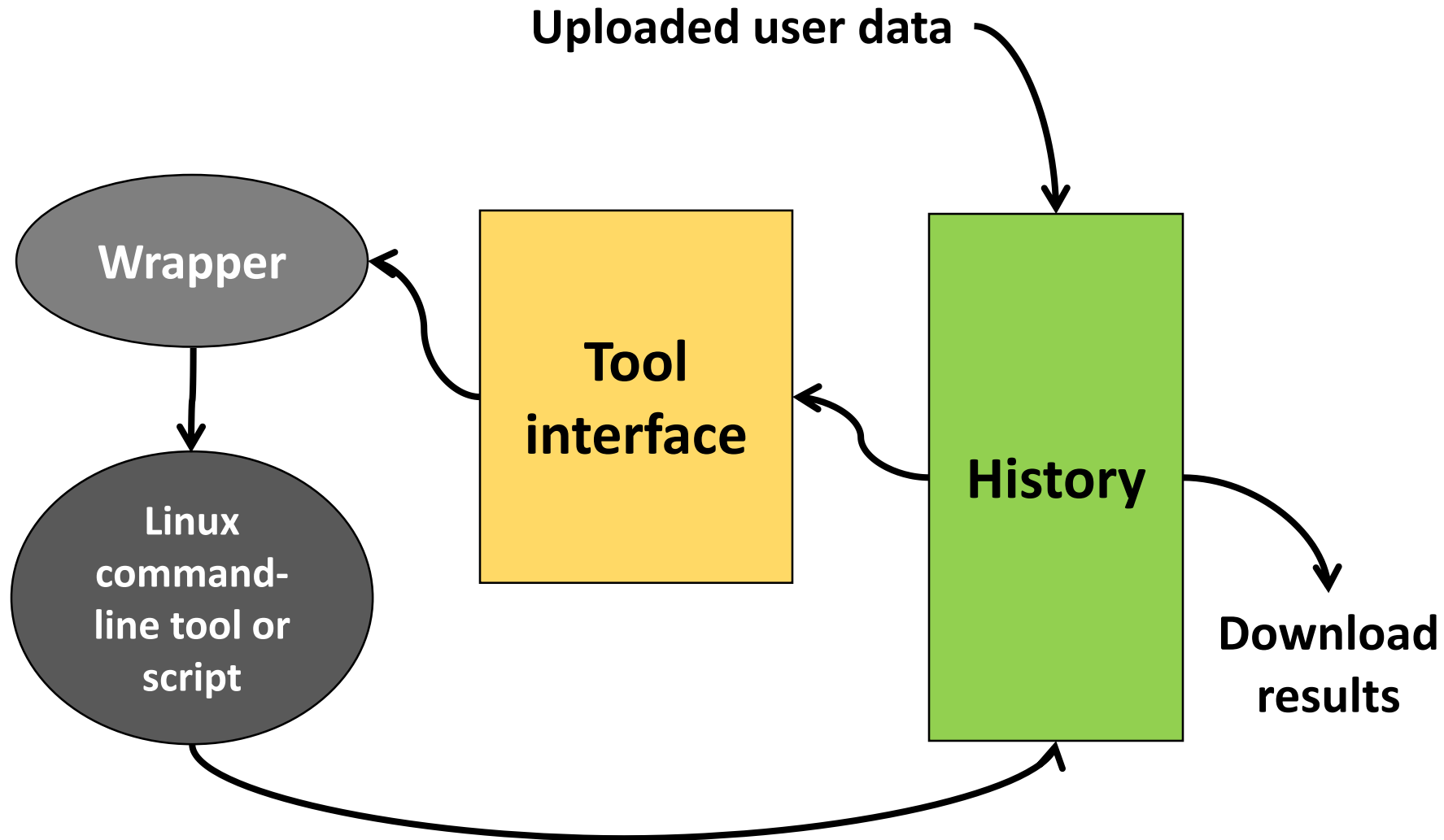
Center panel: analysis
and results

Right panel: history

The screenshot displays the Galaxy CPT web interface, which is organized into three main panels:

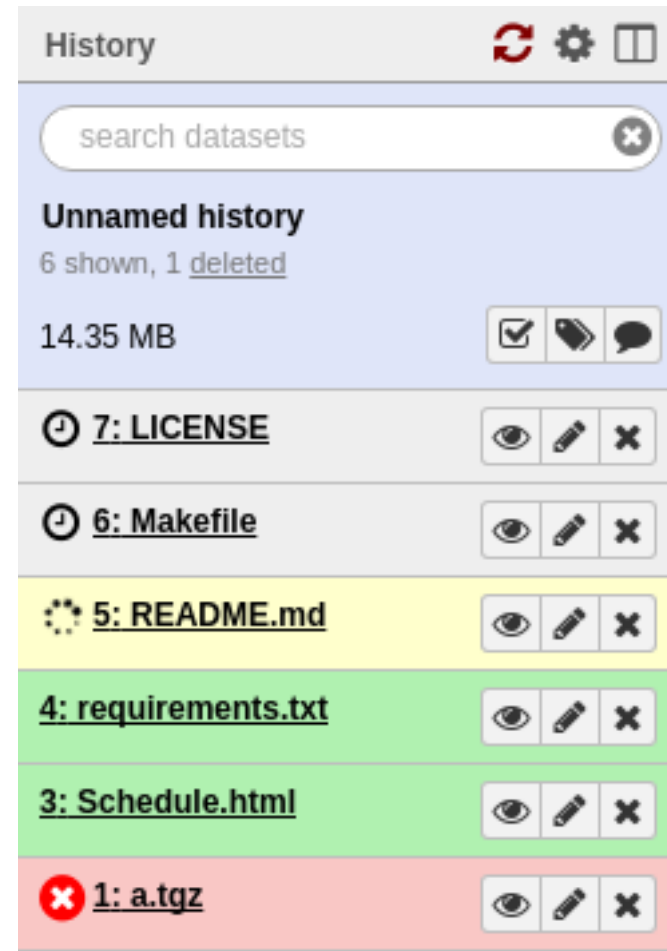
- Left Panel (Tools):** Contains a search bar and a list of available tools categorized under "CPT TOOLS". Tools include CPT2: 464 Tools, CPT2: Utilities, CPT2: ABIF/AB1, CPT2: Blast, CPT2: Fasta Tools, CPT2: GFF3, CPT2: Genbank Tools, CPT2: Comparative Genomics, CPT2: Phage Analysis Tools, CPT2: NGS, CPT2: PAUSE3, CPT2: JBrowse, CPT:Scripts and Analysis, CPT:Genbank, CPT:Blast, CPT:Admin, CPT:External Software, CPT:Oneoff/Custom, CPT:Circos Tools, and CPT:PHANTASM v1.
- Center Panel (Analysis and Results):** Displays a genomic track visualization. The top section shows "Galaxy Updates" and "Available Tracks" (Gene Calls, Reference sequence). The main track area shows a genomic region (NC_005880:9611..13120, 3.51 Kb) with various annotations including GeneMarkS, Glimmer3, MetaGeneAnnotator, and ShineFind from MGA. The track shows gene calls (gene_27, gene_28, gene_29, gene_30, gene_31, gene_32) and reference sequences (cds_orf00028, cds_orf00029, cds_orf00030, cds_orf00031, cds_orf00032).
- Right Panel (History):** Displays a list of datasets and workflows. The top section shows "Copy of 'PAP of Mt0425' shared by 'ryland@tamu.edu' (active items only)". Below this, a list of datasets and workflows is shown, including "61: NCBI EFetch Results", "40: Concatenate datasets on data 39 and data 37", "39: Filter sequences by length on data 38", "38: Genbank Genome Sequence Export on data 36", "37: JBrowse on Mt0425DNA.fasta", "36: NCBI Entrez EFetch on data 33", "35: Rebase GFF3 features on data 32 and data 14", "34: Report on top blast hits", and "33: Top accession".

General order of operations



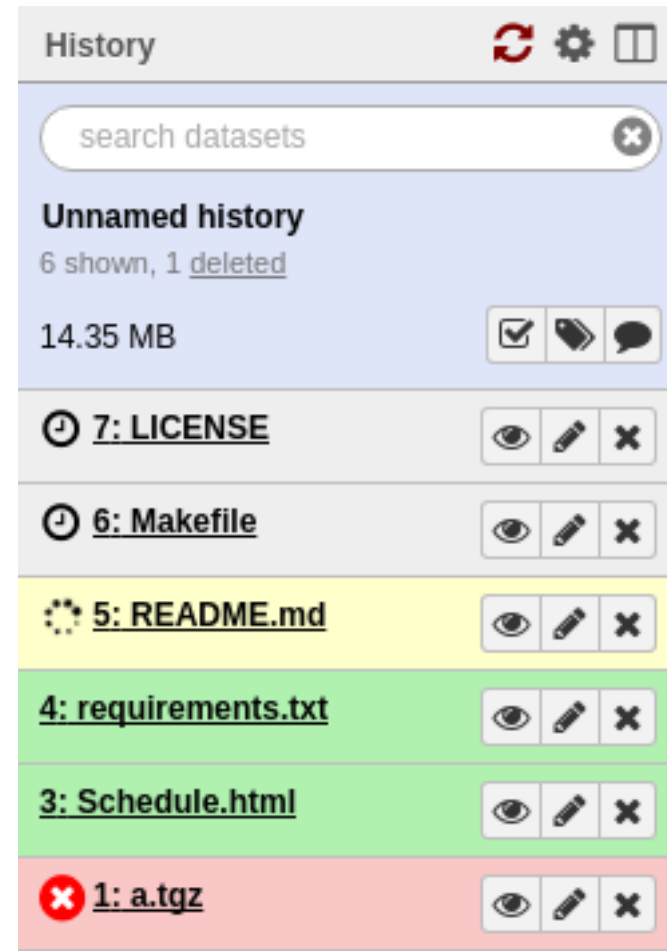
The history

- The history panel contains all input and output data for your analysis
- All input data for tools (sequences, Genbank files, etc.) must be uploaded to the history to be passed through to the tool
- All running jobs will appear in the history in the order they were entered
- All tool output data will appear in the history



The history

- Each item in the history is a **dataset** and appears in the order it was entered
 - Each item is numbered
 - Numbers can't be changed but names can be edited by the user
- **Grey** items are queued to run
- **Yellow** items are running
- **Green** items are jobs that are completed and ready for viewing, download or input into the next tool
- **Red** items are jobs that failed or returned an error



The history

- The user can create an unlimited number of new histories to keep track of related analyses
- Datasets can be copied or moved between histories
- Histories can be copied or shared between users

The screenshot displays the Galaxy / CPT web interface. The top navigation bar includes links for Analyze Data, Workflow, Shared Data, Visualization, Admin, Help, and User, along with a 'Using 33.7 GB' status indicator. Below the navigation bar, there are search bars for 'search histories' and 'search all datasets'. The main content area is divided into four panels, each representing a different history:

- Margaery stitching**: 27 shown, 6 deleted, 17 hidden. 16.8 MB. Contains datasets like '50: JBrowse on Margaery 150805' and '49: Convert XMFA to gapped GFF3 on data 13, data 43, and data 45'.
- Papaya PAUSE**: 8 shown, 9 deleted, 4 hidden. 1.7 GB. Contains datasets like '21: Fix Gene Boundaries on Percy', '19: Caulobacter phage Percy.gbk', '16: Analyse TerL Sequences on data 15', and '15: Pasted Entry'.
- Pierogi PAUSE**: 11 shown, 8 deleted, 36 hidden. 2.0 GB. Contains datasets like '55: JBrowse on Pierogi.fa', '53: MIST v3 on data 49', '36: Phage QC on data 14 and data 31', and '34: Start Codon Usage'.
- Barrett PAUSE**: 4 shown, 22 deleted, 19 hidden. 1.9 GB. Contains datasets like '8: JBrowse on NODE2 Barrett150730 143268 cov 23.5.fa', '3: NODE2 Barrett150730 143268 cov 23.5.fa', and '2: GSAF Download (Sample2 S3 L001 R2 001.fastq)'.

Each panel includes a 'Switch to' button and a 'search datasets' input field. The datasets are listed with their names, sizes, and status (shown, deleted, hidden). Each dataset entry has a small icon (eye, pencil, and X) for viewing, editing, or deleting the dataset.

Datasets in the history

29: Start Codon Usage

4 lines, 1 comments

format: **tabular**, database: ?

1 2 3

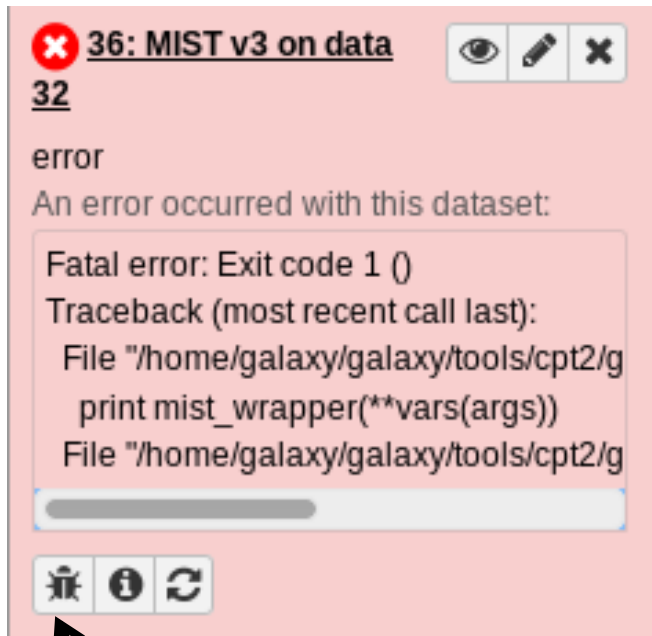
4 5 6 7

8 9

1	2
# Codon	Count
atg	244
gtg	5
tta	1
ttg	3

1. **Eyeball** views the dataset in the main panel
2. **Pencil** modifies metadata: name, data type, etc
3. **X** sends a dataset to the trash. You can recover deleted datasets (see below)
4. **Save** downloads the dataset to your hard-drive. You don't *need* to do this, as Galaxy will always have a copy for you
5. **Information** views details about the tool that was run and how it was configured
6. **Rerun** is a very commonly used button. This lets you re-run the tool, with the same parameters configured
 - Need to run the same tool with slightly different parameters? Don't waste time filling out the tool form; re-run it and tweak those.
 - Job failed? Try modifying the tool inputs and re-running it.
7. **Visualize** lets you visualize compatible datasets
8. **Tags** let you annotate datasets with tags
9. **Comments** let you comment on a dataset to remind yourself why you did it, or maybe to annotate some interesting results you found in the output

Failed jobs






Bug report button

- Not failures but learning opportunities!
- First: did you use the tool correctly?
 - Correct input file
 - Correct parameters set
- If the tool has truly failed, click the “bug” icon to submit a bug report
- Submitting bug reports will help us improve the service

Analyzing data

NCBI BLAST+ blastp Search protein database with protein query sequence(s) (Galaxy Version 0.1.01) Options

Protein query sequence(s)

 40: Phage K NO INTRONS all CDS

Subject database/sequences
Locally installed BLAST database


Protein BLAST database
NR 2017-9

Type of BLAST
☒ blastp - Traditional BLASTP to compare a protein query to a protein database
☐ blastp-short - BLASTP optimized for queries shorter than 30 residues

Set expectation value cutoff
0.001

Output format
Tabular (extended 25 columns)




Advanced Options
Hide Advanced Options

 **Note.** Database searches may take a substantial amount of time. For large input datasets it is advisable to allow overnight processing.

Analyzing data

NCBI BLAST+ blastp Search protein database with protein query sequence(s) Options

Protein query sequence(s)

   40: Phage K NO INTRONS all CDS

Subject database/sequences

Locally installed BLAST database

Protein BLAST database

NR 2017-9

Type of BLAST

☒ blastp - Traditional BLASTP to compare a protein query to a protein database

☐ blastp-short - BLASTP optimized for queries shorter than 30 residues

Set expectation value cutoff


0.001


Output format

Tabular (extended 25 columns)

Advanced Options

Hide Advanced Options

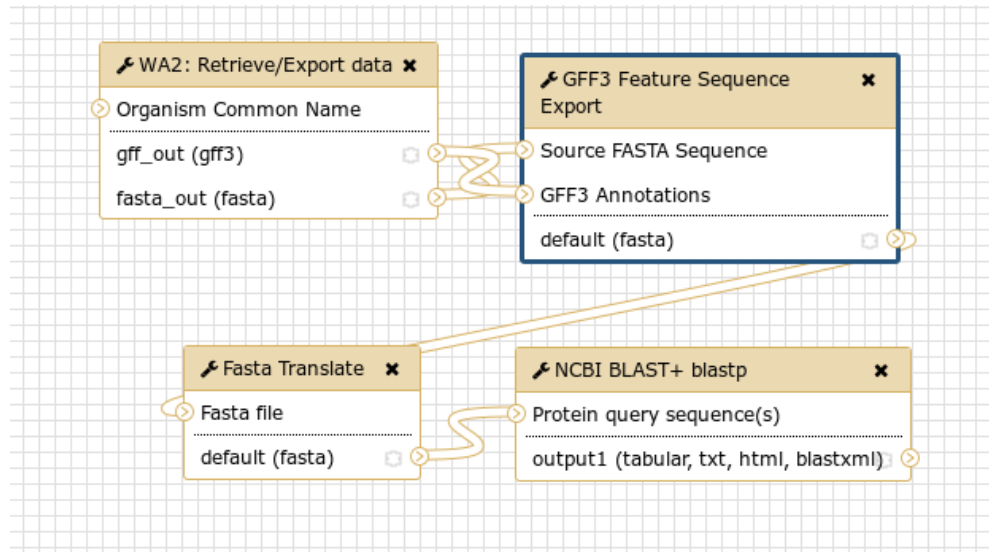
 Execute

 **Note.** Database searches may take a substantial amount of time. For large input queries, you may want to allow overnight processing.

- The tool options presented are determined by the wrapper, but these are all switches that could be entered at the command line
- Input data must come from the history
- Output will appear in the history when the job is launched

Workflows

- One of Galaxy's most powerful features is the ability to connect jobs in workflows
- Some analyses take a long time; output from one job will automatically be handed off to the next when it finishes



Galaxy training resources

- Galaxy home: <https://usegalaxy.org/>
- Galaxy 101: <https://galaxyproject.org/tutorials/g101/>
- CPT Galaxy training: <https://cpt.tamu.edu/training-material>

Welcome to CPT Galaxy Training

Collection of tutorials for CPT Galaxy users and BICH464 students. Further tutorials developed and maintained by the worldwide Galaxy community are available here.

CPT Galaxy for Students

Topic	Tutorials
Introduction to Galaxy and Apollo	6
Additional Analyses	5
Phage Annotation Pipeline in CPT Galaxy	4

CPT Galaxy for Scientists

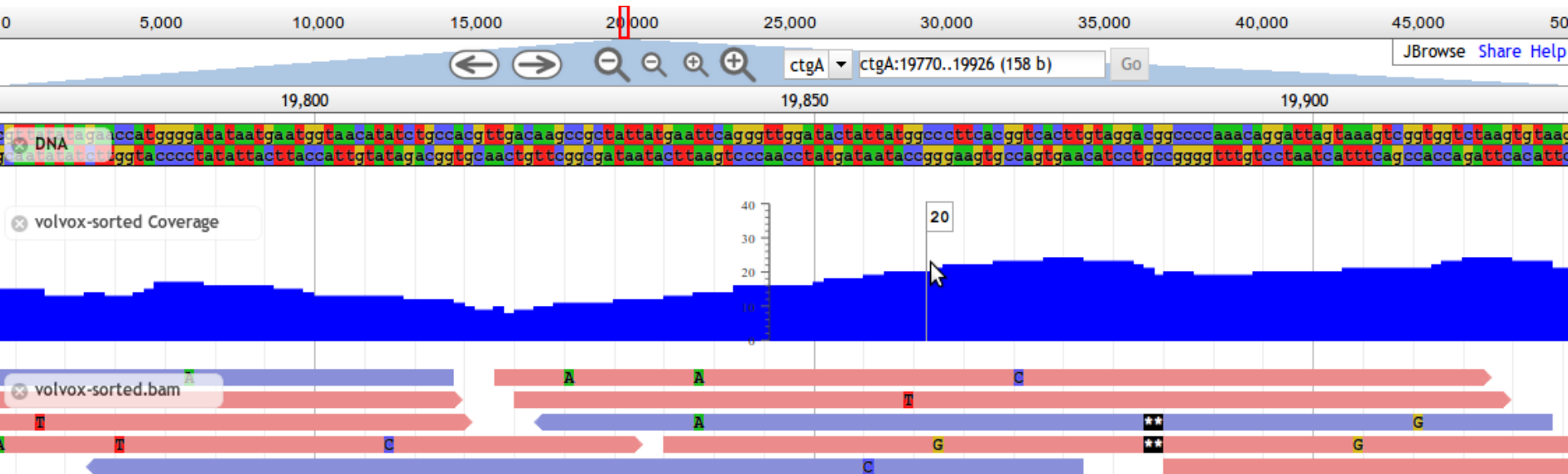
Topic	Tutorials
De Novo Assembly	2

What is WebApollo?

- WebApollo is an interactive genome visualizer that supports collaborative genome annotation
 - An extension of the popular JBrowse genome viewer that allows editing
 - “Google Docs, but for genomes”
- Still in development, Apollo is less robust than Galaxy and new features continue to be added
- Maintains genome annotations and multiple evidence “tracks” to guide annotations
- The CPT has developed tools that bridge Galaxy ↔ Apollo

JBrowse

- JBrowse is a genome viewer implemented in many online tools (including RAST and PATRIC)
- Viewer only, no editing function
- Apollo is an addition to JBrowse
 - To work with data in Apollo, it is first used to generate a JBrowse instance that is then loaded into Apollo



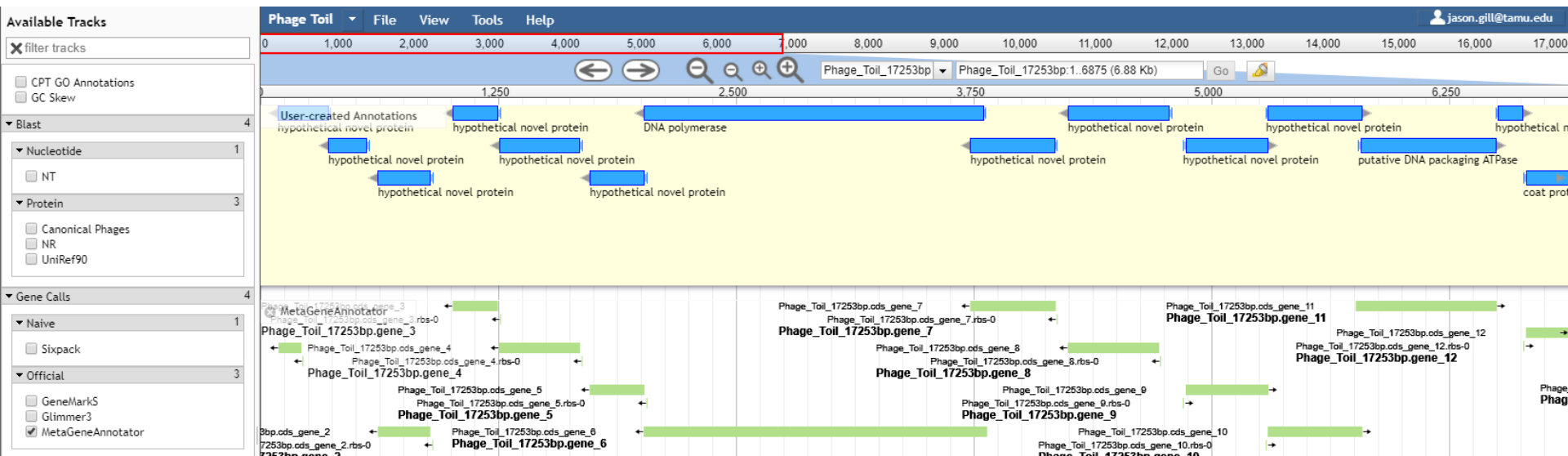
Annotations in Apollo

- User annotations appear in the topmost track of the display
- Tracks below this are generated by various tools in Galaxy and imported to Apollo

The screenshot displays the Apollo genome browser interface. On the left, the 'Available Tracks' panel is visible, showing a search bar for 'filter tracks' and several track categories: 'Blast' (4 tracks), 'Nucleotide' (1 track), 'Protein' (3 tracks), 'Gene Calls' (4 tracks), and 'Naive' (1 track). The 'Protein' section is expanded, showing options for 'Canonical Phages', 'NR', and 'UniRef90'. The 'Gene Calls' section is also expanded, showing options for 'GeneMarkS', 'Glimmer3', and 'MetaGeneAnnotator'. The main display area shows a genomic track for 'Phage Toil' with a scale from 0 to 11,000. The track is divided into several sections: 'User-created Annotations' (topmost), 'Nucleotide' (1 track), 'Protein' (3 tracks), 'Gene Calls' (4 tracks), and 'Naive' (1 track). The 'User-created Annotations' track shows several blue bars representing protein annotations, including 'hypothetical novel protein', 'putative DNA packaging ATPase', 'coat protein', and 'LysM domain protein'. The 'Gene Calls' track shows a blue bar for 'DNA polymerase'. The 'Naive' track shows a blue bar for 'Sixpack'.

Evidence tracks in Apollo

- Features in Apollo can **only** be created from evidence tracks
- Unlike purely manual editors like Artemis, the user cannot select sequence and create a feature *de novo*
- This is part of a philosophical decision by Apollo, that features are **only created with evidence**



File formats

- File formats are important!
- Sequence analysis is computational, and each program has an expected input and output file format
- Different formats are used by different tools
- A strength of Galaxy is the ability to link analysis and file format conversion in workflows
- Note that most programs that deal with DNA or protein sequence are expecting data in *plain text* (ASCII, UTF8) format

General Feature Format, version 3

```
##gff-version 3
ctg123 . mRNA          1300  9000  .  +  .  ID=mrna0001;Name=sonichedgehog
ctg123 . exon          1300  1500  .  +  .  ID=exon00001;Parent=mrna0001
ctg123 . exon          1050  1500  .  +  .  ID=exon00002;Parent=mrna0001
ctg123 . exon          3000  3902  .  +  .  ID=exon00003;Parent=mrna0001
ctg123 . exon          5000  5500  .  +  .  ID=exon00004;Parent=mrna0001
ctg123 . exon          7000  9000  .  +  .  ID=exon00005;Parent=mrna0001
```

- GFF3 is becoming a dominant format for storing sequence data
- One line per feature: compact, easier to search, parse, and process
- Can be used to store data other than genome annotations: BLAST alignments, conserved domains, etc.
 - Jbrowse/Apollo natively recognizes features in the GFF3 format
- The DNA sequence can be stored as part of the GFF3 file as a FASTA sequence, or can exist as a separate FASTA file

General Feature Format, version 3

```
##gff-version 3
ctg123 . mRNA      1300  9000  .  +  .  ID=mrna0001;Name=sonichedgehog
ctg123 . exon      1300  1500  .  +  .  ID=exon00001;Parent=mrna0001
ctg123 . exon      1050  1500  .  +  .  ID=exon00002;Parent=mrna0001
ctg123 . exon      3000  3902  .  +  .  ID=exon00003;Parent=mrna0001
ctg123 . exon      5000  5500  .  +  .  ID=exon00004;Parent=mrna0001
ctg123 . exon      7000  9000  .  +  .  ID=exon00005;Parent=mrna0001
```


type: Type of feature
(gene, exon, CDS,
etc.)

source: Name
of the program
that generated
the feature

seqid: Name of the
DNA sequence the
annotation refers to

General Feature Format, version 3

```
##gff-version 3
ctg123 . mRNA          1300  9000  .  +  .  ID=mrna0001;Name=sonichedgehog
ctg123 . exon         1300  1500  .  +  .  ID=exon00001;Parent=mrna0001
ctg123 . exon         1050  1500  .  +  .  ID=exon00002;Parent=mrna0001
ctg123 . exon         3000  3902  .  +  .  ID=exon00003;Parent=mrna0001
ctg123 . exon         5000  5500  .  +  .  ID=exon00004;Parent=mrna0001
ctg123 . exon         7000  9000  .  +  .  ID=exon00005;Parent=mrna0001
```



start, end : Coordinates of the start and end of the feature, as base position of the sequence specified by **seqid**

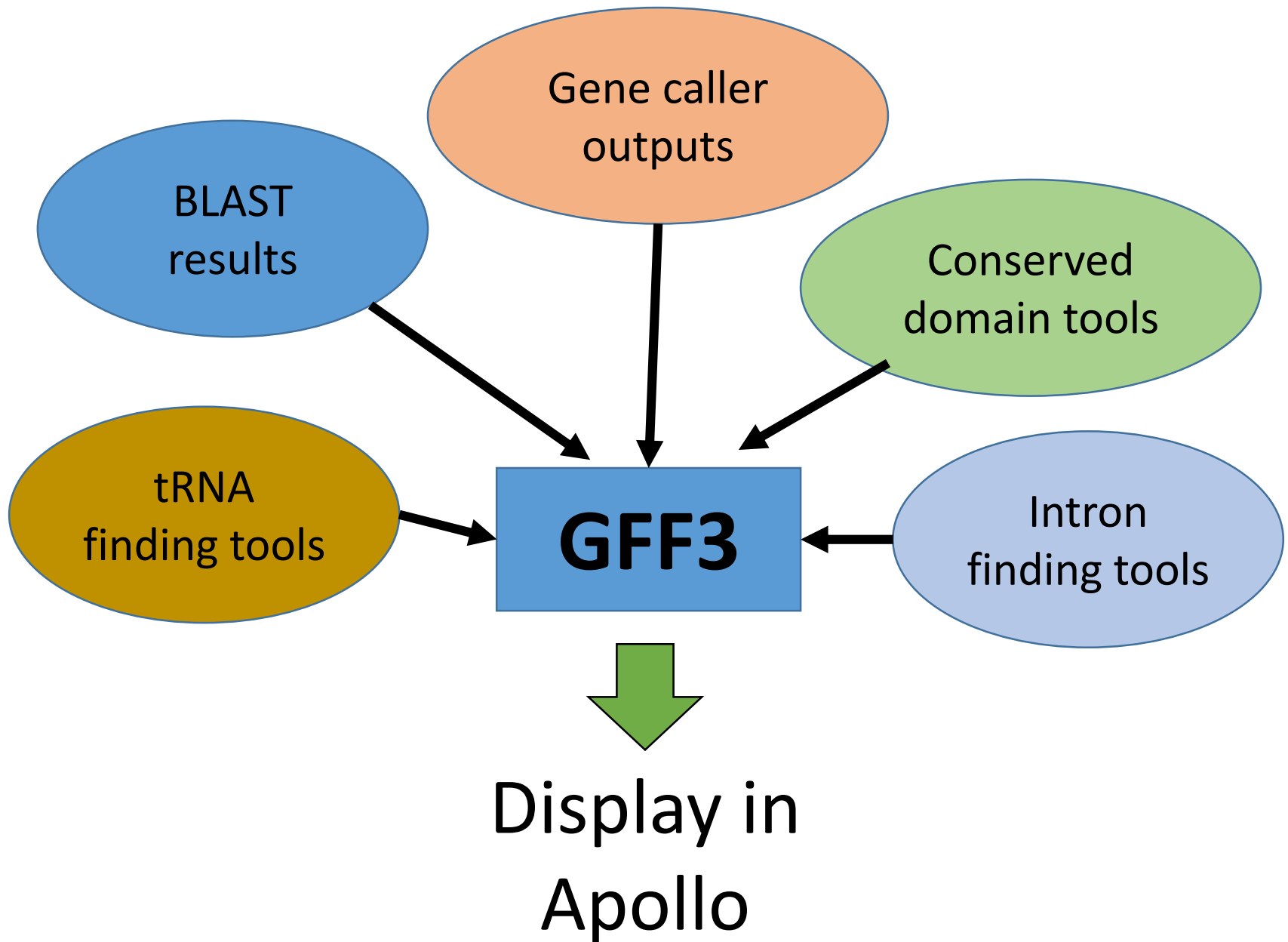
score: Feature score (e.g., E-value, P value)

strand: Plus (+) or minus (-) DNA strand

phase: where the feature begins relative to the reading frame; 0, 1, or 2 base offset. CDS features must have a phase.

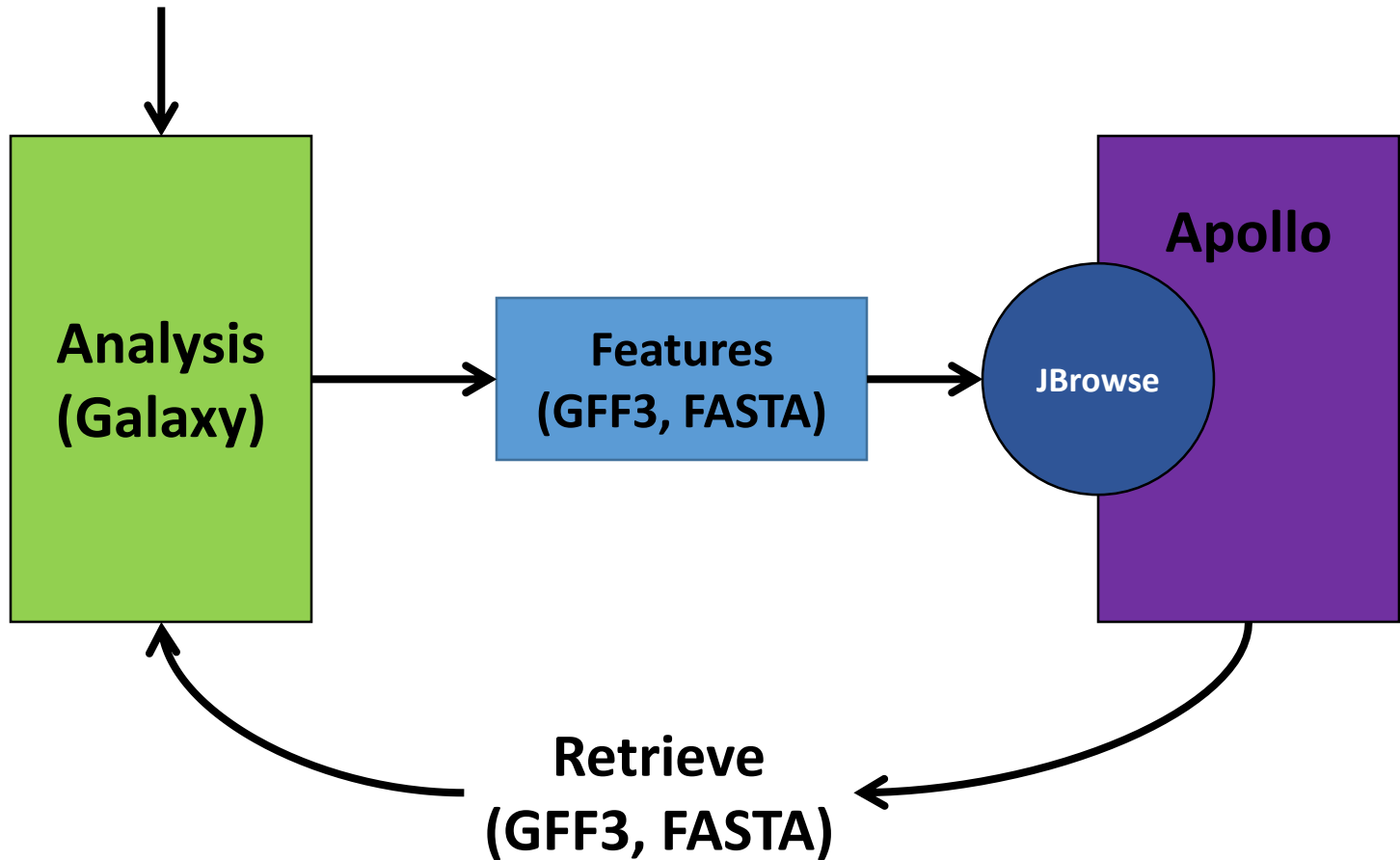
attributes:
Feature attributes listed as tag=value

parent-child relations: Features are nested in a hierarchy of mRNA>gene>CDS; makes sense for eukaryotic genomes but is largely redundant for bacteria



Galaxy/Apollo order of operations

User uploaded data



Different phage strategies for DNA packaging



cos: Complementary overhangs, DNA packaging is *site-specific*; each packaged DNA molecule is identical and 100% of the genome. Overhangs can be 3' or 5', depending on the phage.

Paradigm phage: λ

ABCD XYZA



BCDE YZAB



CDEF ZABC



pac: DNA molecules are *terminally redundant* and *permuted*; each DNA molecule is >100% of the genome and starts and stops at a different location. Packaging can initiate from a specific *pac* site or randomly, depending on the phage.

Paradigm phage: T4



Terminal repeats (TR): DNA molecules are *terminally redundant* but *not permuted*; each DNA molecule is >100% of the genome but starts and stops at the same place (each molecule is identical). TR's can be ~100 bp to >10 kb.

Paradigm phage: T7



Terminal proteins: DNA molecules have a protein covalently linked to the 5' end of each strand. Each packaged DNA molecule is identical and is 100% of the genome.

Paradigm phage: phi29

Assembly of cos or terminal protein phage DNA

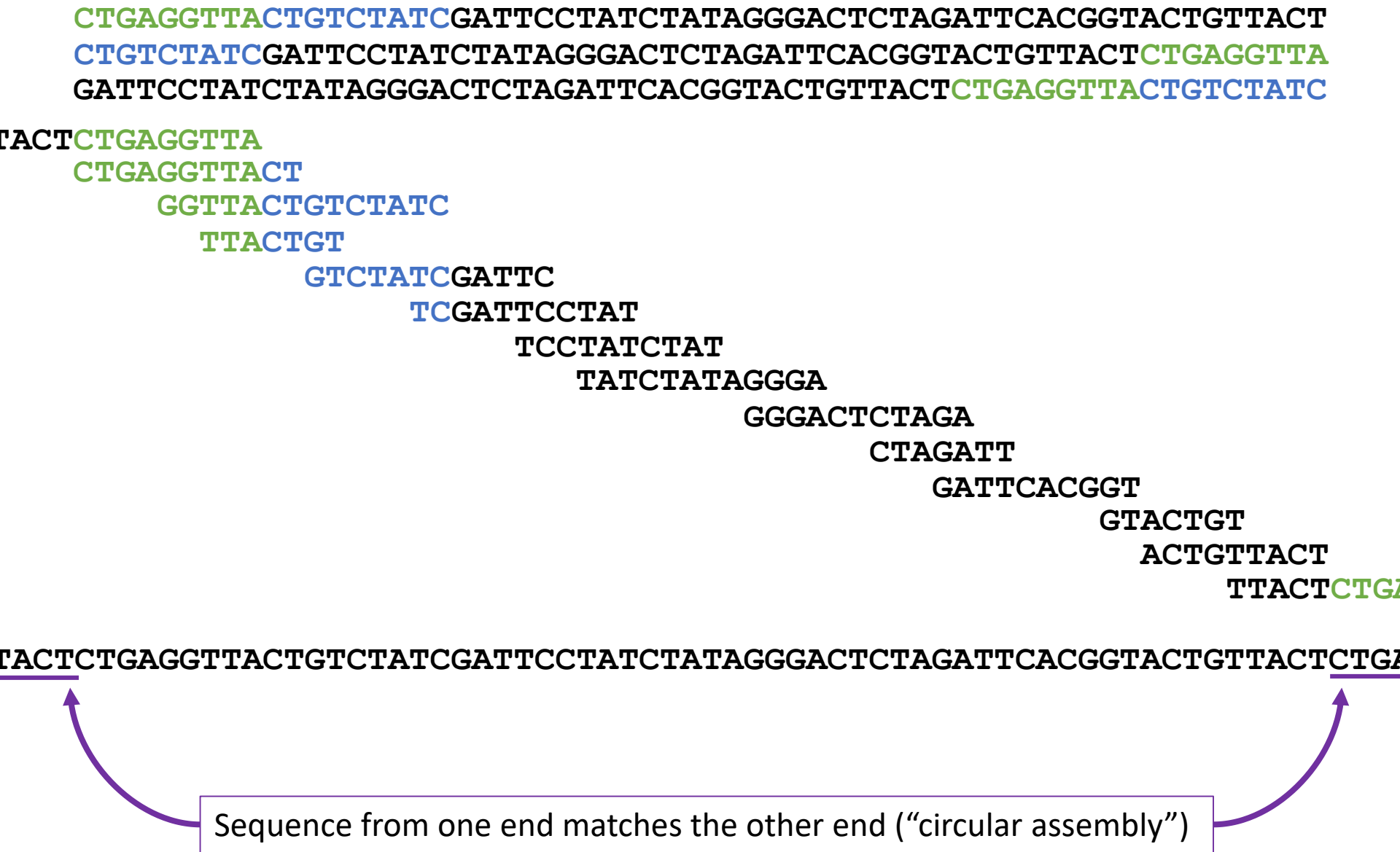
CTGAGGTTACTGTCTATCGATTCCTATCTATAGGGACTCTAGATTCACGGTACTGTTACT
CTGAGGTTACTGTCTATCGATTCCTATCTATAGGGACTCTAGATTCACGGTACTGTTACT
CTGAGGTTACTGTCTATCGATTCCTATCTATAGGGACTCTAGATTCACGGTACTGTTACT

CTGAGGTTACT
 GGTTACTGTCTATC
 TTACTGT
 GTCTATCGATTC
 TCGATTCCTAT
 TCCTATCTAT
 TATCTATAGGGA
 GGGACTCTAGA
 CTAGATT
 GATTCACGGT
 GTACTGT
 ACTGTTACT

CTGAGGTTACTGTCTATCGATTCCTATCTATAGGGACTCTAGATTCACGGTACTGTTACT

- Each DNA molecule packaged into the phage head is identical
- *In theory*, the assembly should reflect the original DNA molecule

Assembly of *pac* phage DNA



Assembly of DTR phage DNA

GTACTGTTACTGTCTATCGATTCCTATCTATAGGGACTCTAGATTCACG
GTACTGTTACTGTCTATCGATTCCTATCTATAGGGACTCTAGATTCACG
GTACTGTTACTGTCTATCGATTCCTATCTATAGGGACTCTAGATTCACG

TCTATAGGGACT
GACTCTAGAT
GATTCACG
TCACG
ACTGTTACT
TACTGTTAC
GTACTGTT
CTGTTACT
ACTGTTA
TTACTGTCT
CTGTCTATCGA
CGATTCCTATC
TATCTATAGGGA

TCTATAGGGACTCTAGATTCACG GTACTGTTACT GTCTATCGATTCCTATCTATAGGGA

High coverage region

Sequence from one end matches the other end ("circular assembly")

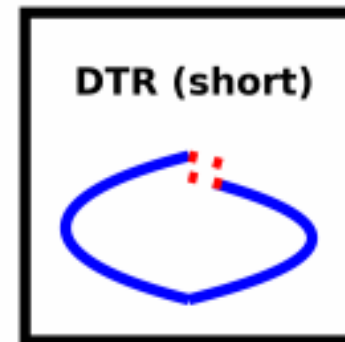
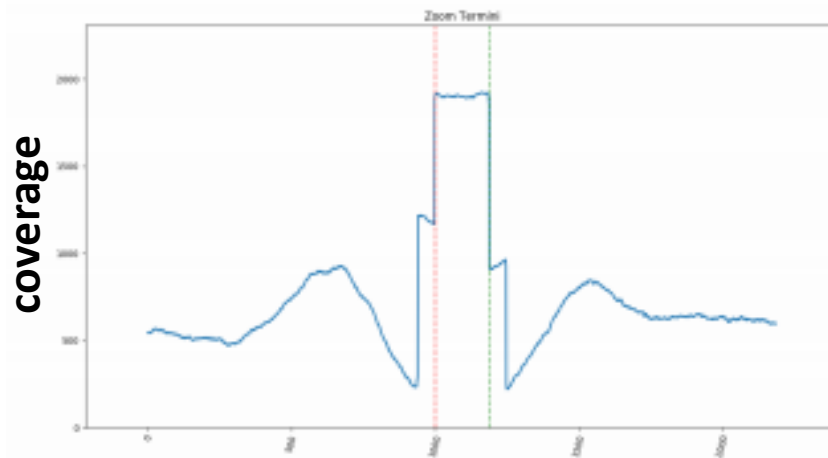
PhageTerm

- PhageTerm attempts to detect genomic termini type and location by mapping Illumina sequence reads back to the assembled contig
- Uses two algorithms
 - Coverage discontinuity: DTR regions have higher coverage
 - Read ends: physical ends in phage DNA lead to over-representation of these ends in the read pool
- Requires the phage sequence and the original reads
- Illumina library prep method affects results: Nextera tends to lose genome ends

Assembly of DTR phage DNA

GTACTGTTACTGTCTATCGATTCCCTATCTATAGGGACTCTAGATTCACGGTACTGTTACT
GTACTGTTACTGTCTATCGATTCCCTATCTATAGGGACTCTAGATTCACGGTACTGTTACT
GTACTGTTACTGTCTATCGATTCCCTATCTATAGGGACTCTAGATTCACGGTACTGTTACT

4s-2 PhageTerm Analysis



TCTATAGGGACTCTAGATTCACGGTACTGTTACTGTCTATCGATTCCCTATCTATAGGGA

High coverage region

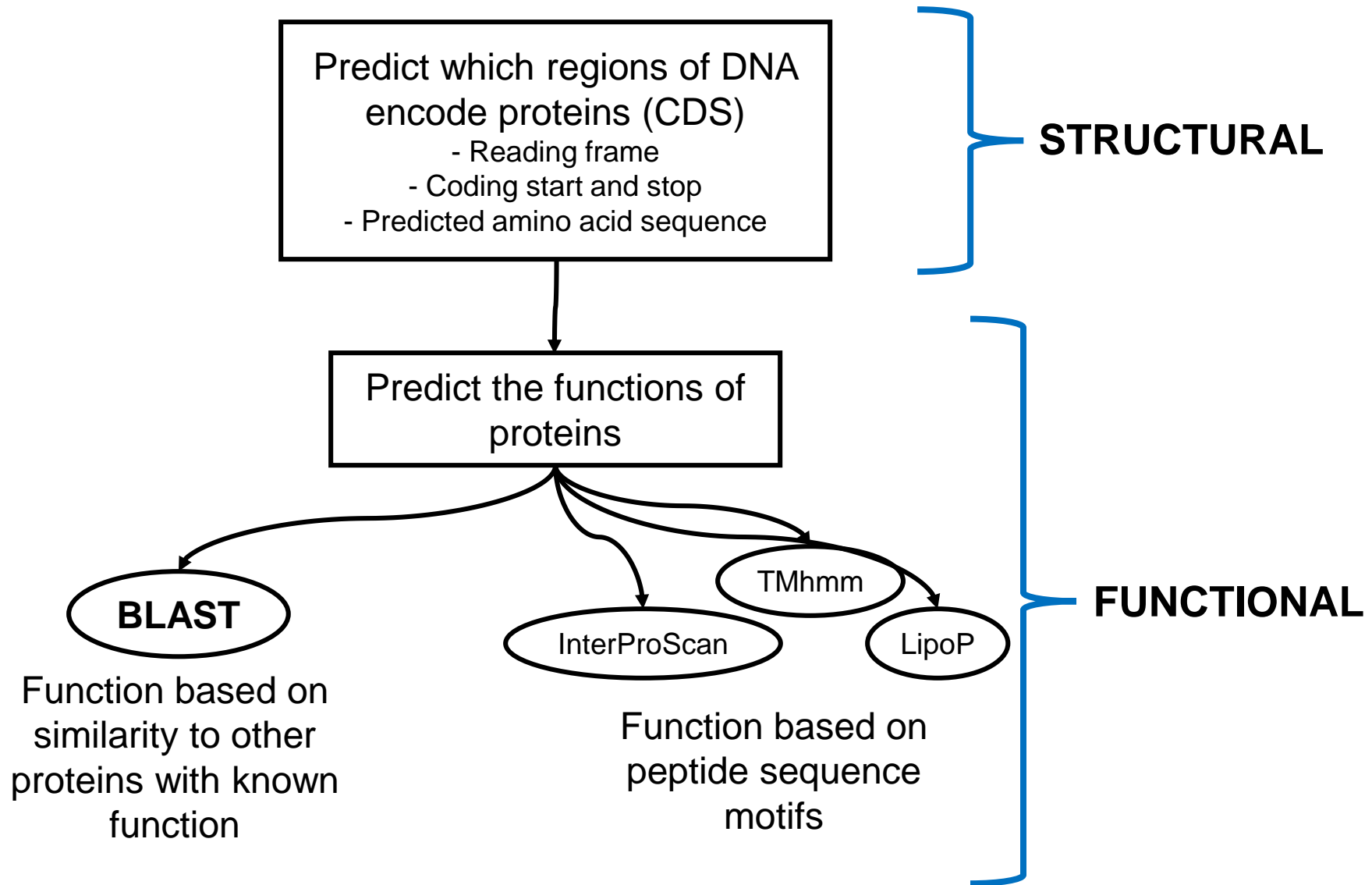
Sequence from one end matches the other end ("circular assembly")

Using PhageTerm to reopen phage contigs

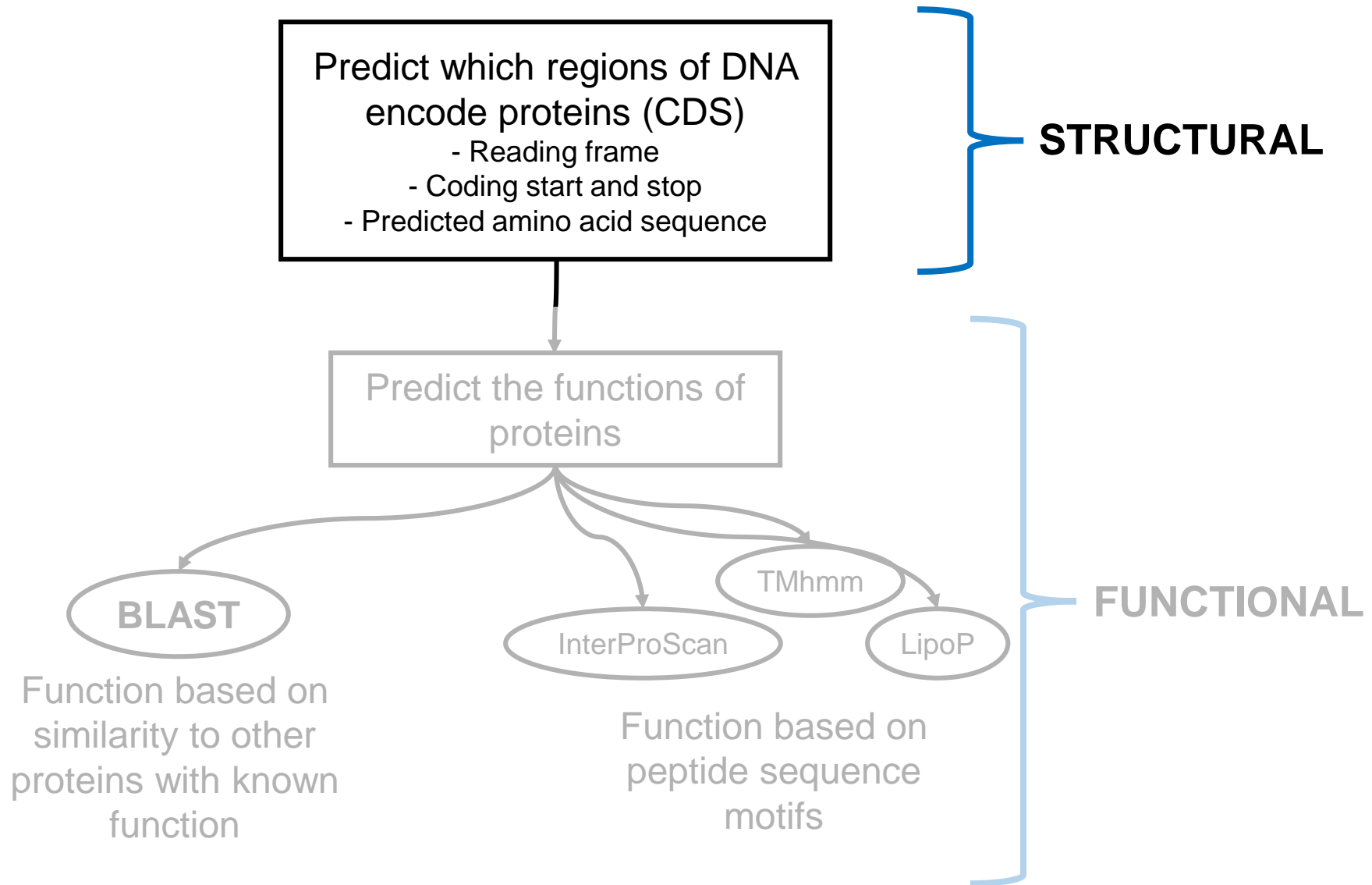
- Phages with *cos* or DTR termini are conventionally opened at the physical termini as present in the packaged phage chromosome
 - Cos: opened at cos sites
 - DTR: opened at the DTR boundary
- Reopening the phage genome *before* annotation simplifies downstream analyses
- If detected, phage contigs should be reopened at cos or DTR boundaries

Structural Annotation

Genome annotation



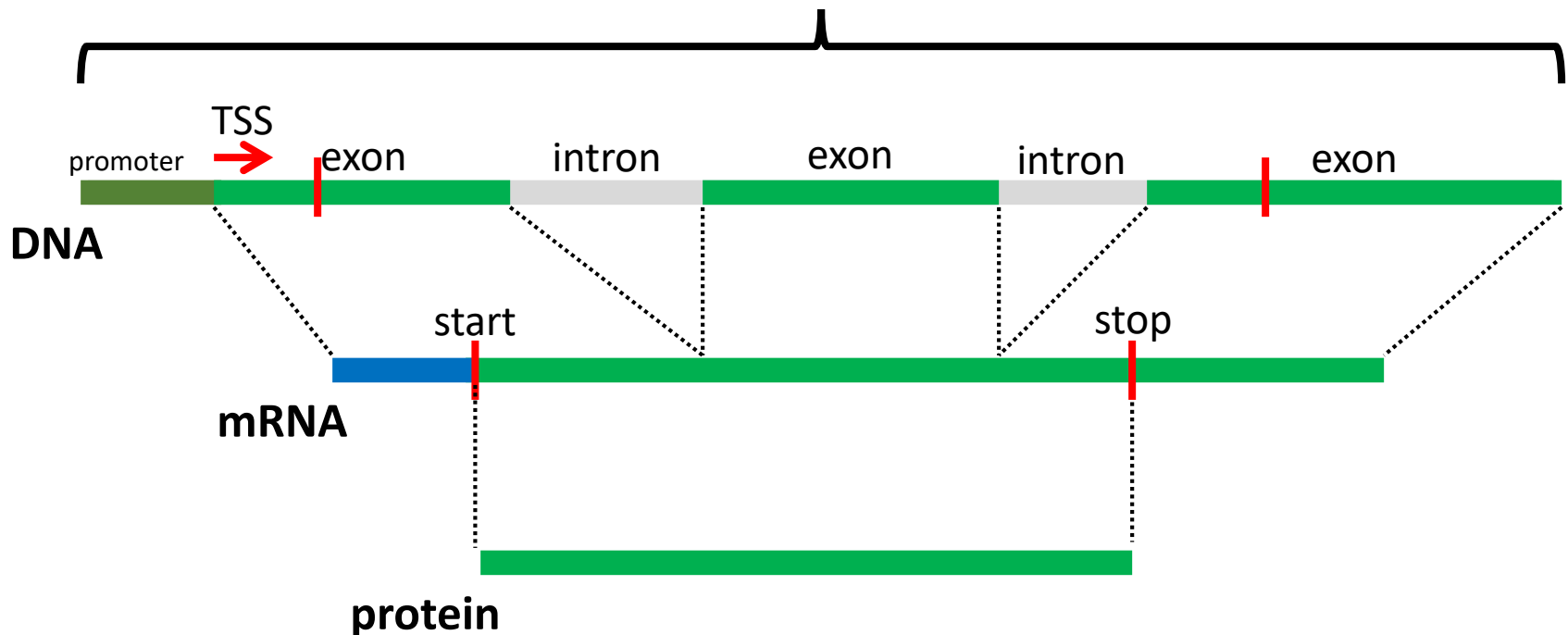
Genome annotation



Eukaryotic gene

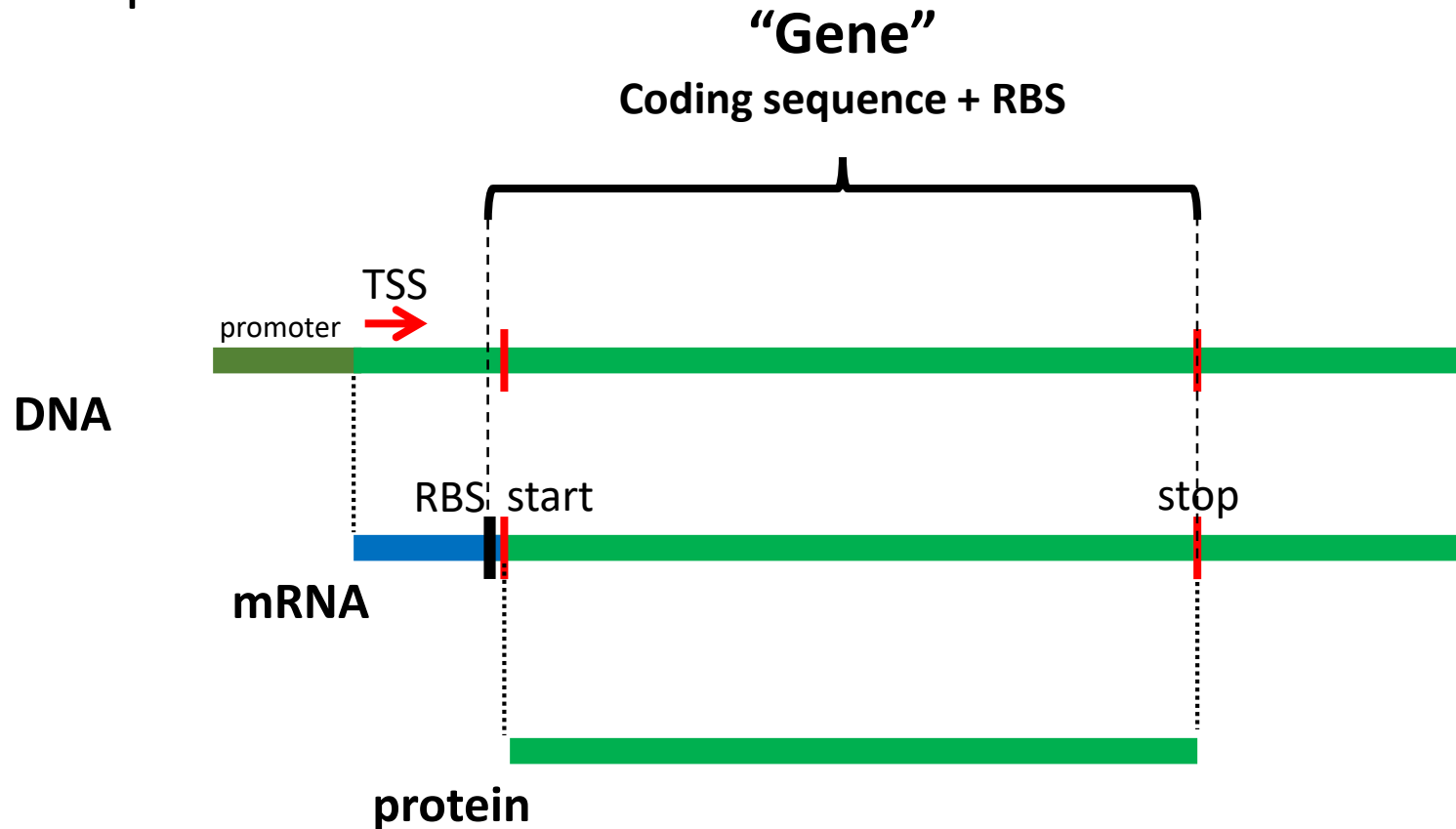
- Extensive mRNA processing for intron splicing, 5' and 3' modification
- Difficult to infer protein sequence directly from DNA sequence

“Gene”

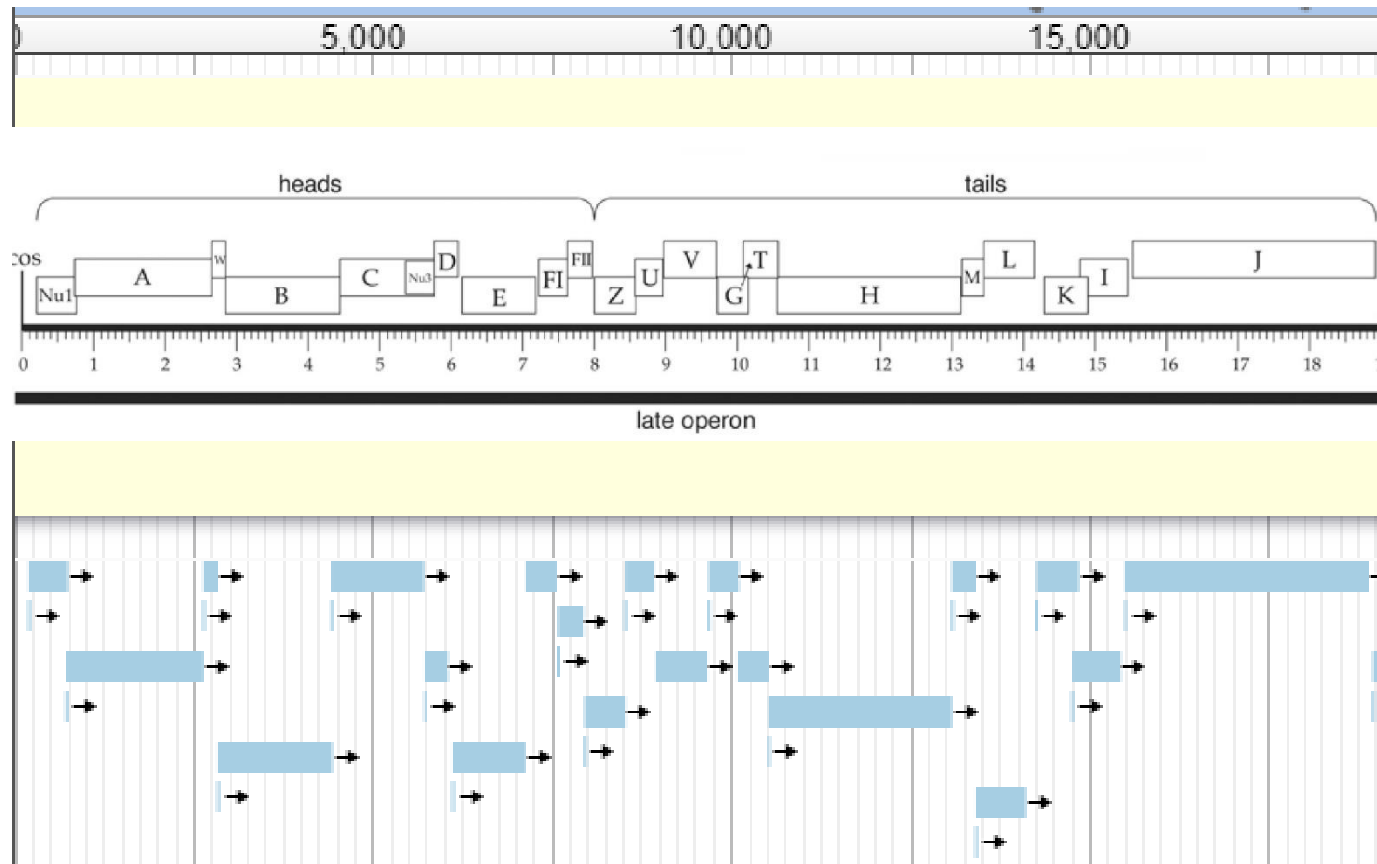


Prokaryotic gene

- Introns rare, little mRNA processing
- Easy to infer protein sequence directly from DNA sequence

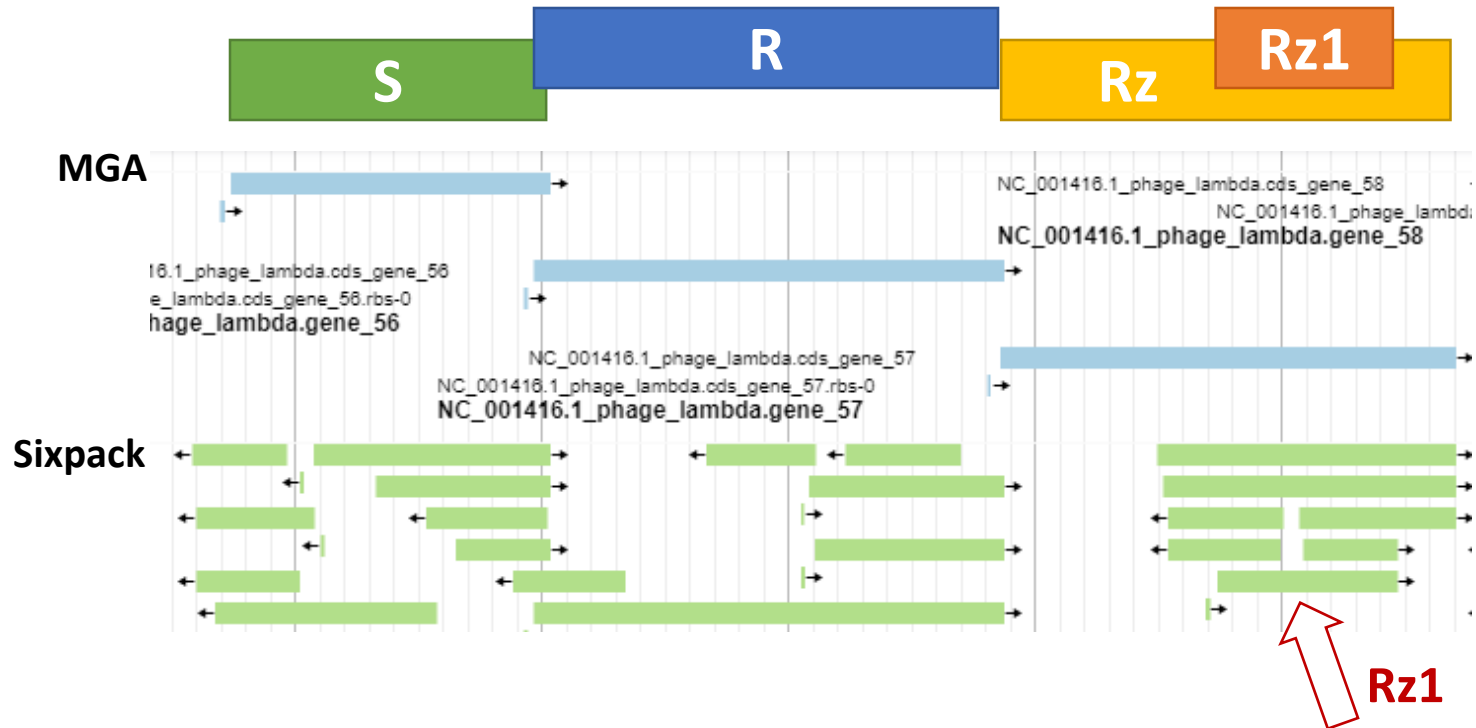


Gene prediction tools



- Gene prediction tools are generally accurate
- Above: gene prediction on the leftmost 19 kb of the lambda genome by MGA identifies all of the protein-coding sequences from Nu1 - J
- Note that prediction tools are looking for “normal” protein-coding genes; cannot find *programmed frameshifts* or *embedded genes*

Gene prediction tools



- MGA identifies *S*, *R* and *Rz* of the lambda lysis cassette but does not find *Rz1*, embedded in *Rz*
 - To MGA, there is no *evidence* that *Rz1* should be called as a gene
- The *Rz1* reading frame is found by Sixpack, but if you were annotating this phage *de novo* you would have no reason to call it at this time
- The evidence for annotating “unusual” protein-coding genes found in phage genomes will come later in separate analyses

Gene prediction and translation initiation

- All protein-coding genes must be translated into protein from an mRNA, which requires **initiation**
- A Translation Initiation Site (TIR) consists of a **Shine-Dalgarno (S-D)** sequence, a **4-12 bp spacer**, and a **start codon**
 - The S-D sequence must base-pair with the complementary sequence at the 3' end of the 16S rRNA to initiate translation of a protein
- The **strength** of translation initiation is affected by how close a gene's RBS is to the consensus S-D sequence **AGGAGGT**, and if the spacer is the right length
- **Any 3-base subset** of the canonical S-D can be used in a TIR
 - Must have appropriate spacing
 - Wobble base-pairing rules apply

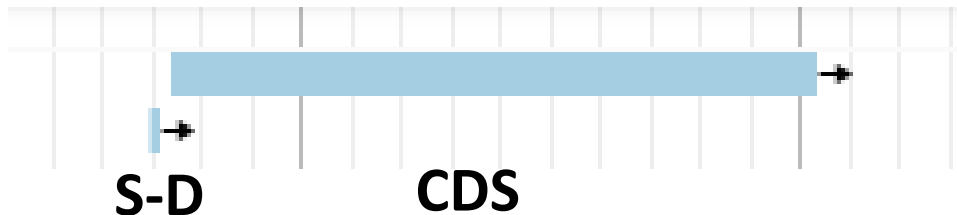
Valid Shine-Dalgarno sequences

<u>Watson-Crick</u>	<u>Wobble (G-U)</u>
---------------------	---------------------

AGGAGGT	AGGAGGT
AGGAGG	GGGGGG
GGAGGT	GGGGGT
AGGAG	AGGGG
GGAGG	GGGGG
GAGGT	GGGGT
AGGA	GGGA
GGAG	GGGG
GAGG	GGGT
AGGT	GGG
AGG	
GGA	
GAG	
GGT	

Shine Find

- The CPT developed the tool *ShineFind* to help with structural annotation
- The tool looks upstream of each ORF called by MGA, Glimmer3 or SixPack and identifies the S-D sequence, and makes this a part of the gene feature



Valid Shine-Dalgarno sequences

Watson-Crick

AGGAGGT

AGGAGG

GGAGGT

AGGAG

GGAGG

GAGGT

AGGA

GGAG

GAGG

AGGT

AGG

GGA

GAG

GGT

Wobble (G-U)

AGGAGGT

GGGGGG

GGGGGT

AGGGG

GGGGG

GGGGT

GGGA

GGGG

GGGT

GGG

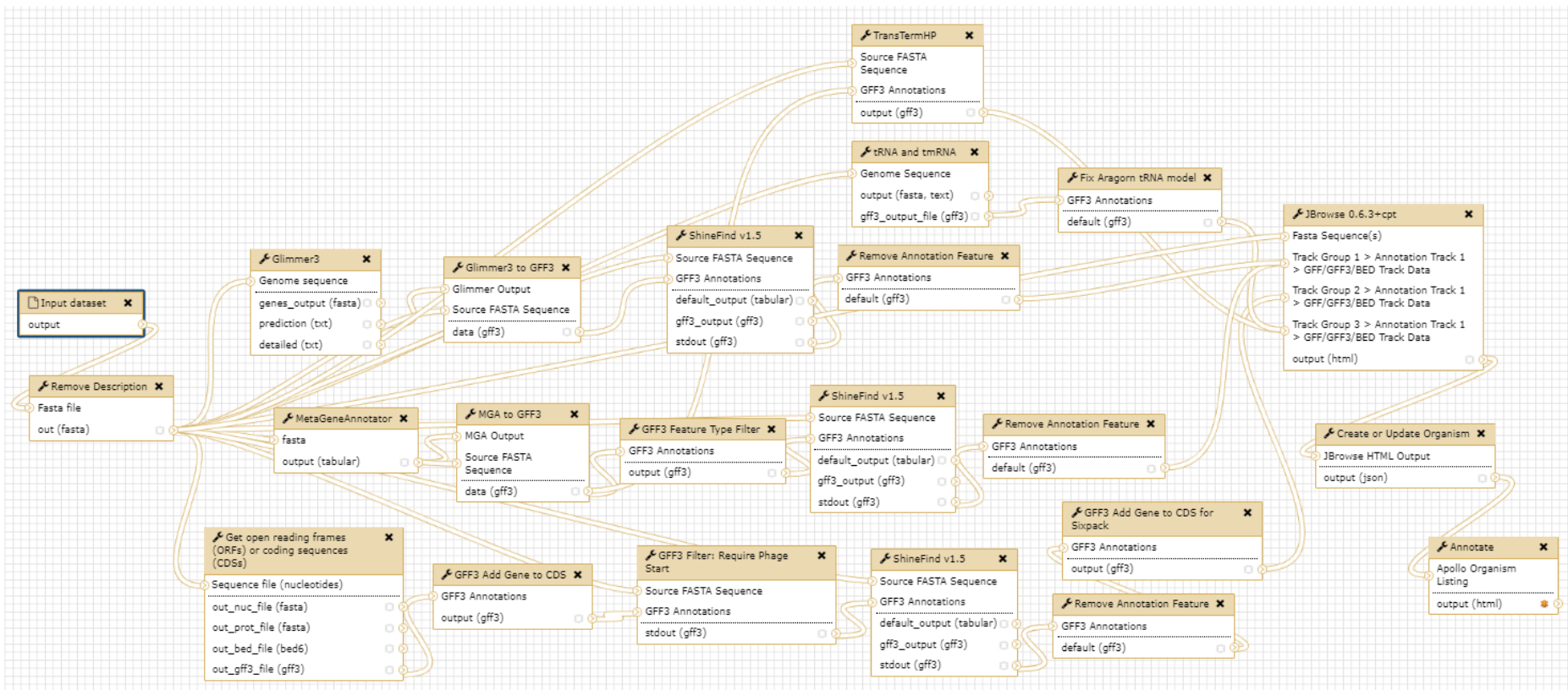
Basic gene structure

- A protein-coding gene **must**:
 - Have a valid start codon: ATG > GTG >> TTG
 - Encode a protein in an **open reading frame (ORF)** determined by the start codon (also called the **coding segment**, or CDS)
 - Be terminated by a stop codon
- A protein-coding gene **should**:
 - Be preceded by a Shine-Dalgarno sequence

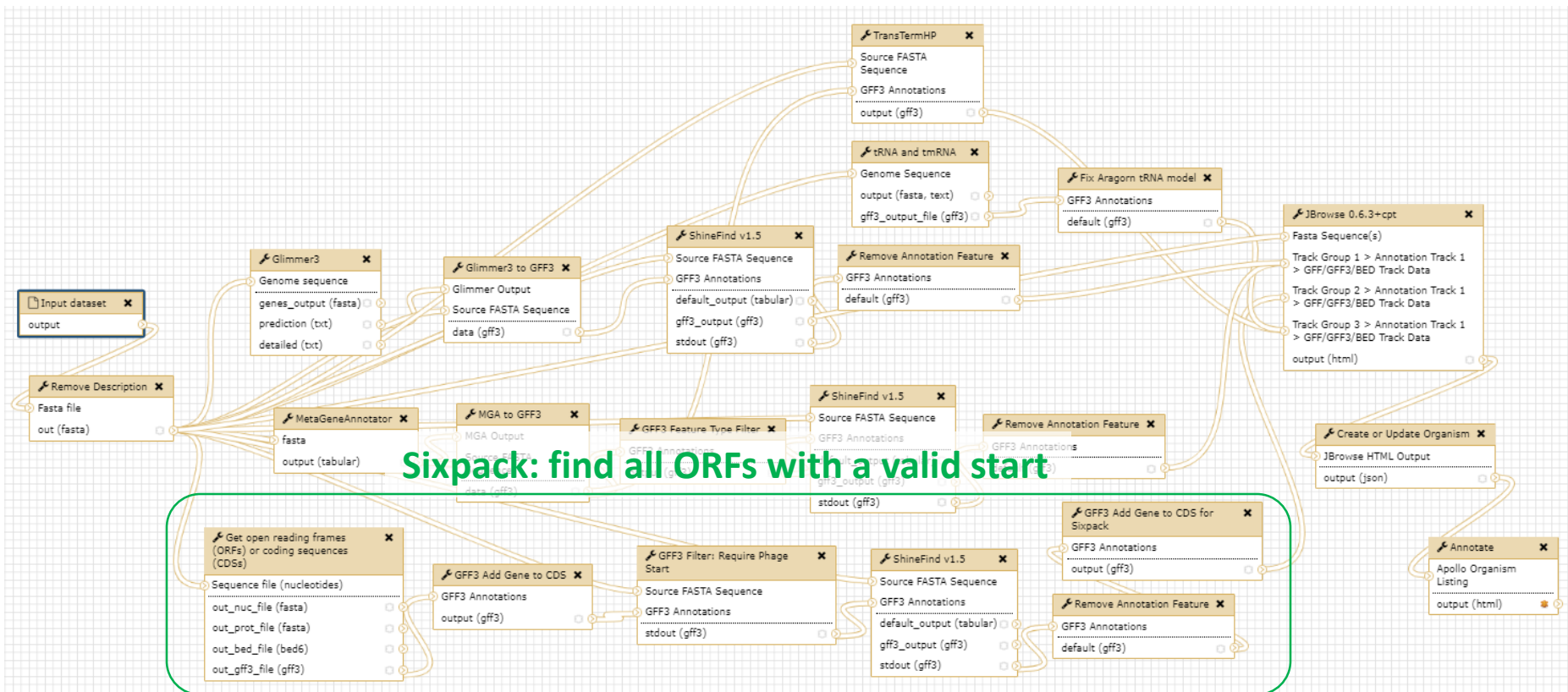


AGT**AGGT**ACCTGATT**ATG**CAGCATGTG...TCGGAT**TAA**GCTT
 M R H V S D *

The CPT Galaxy Structural Annotation workflow



The CPT Galaxy Structural Annotation workflow



Find all ORFs

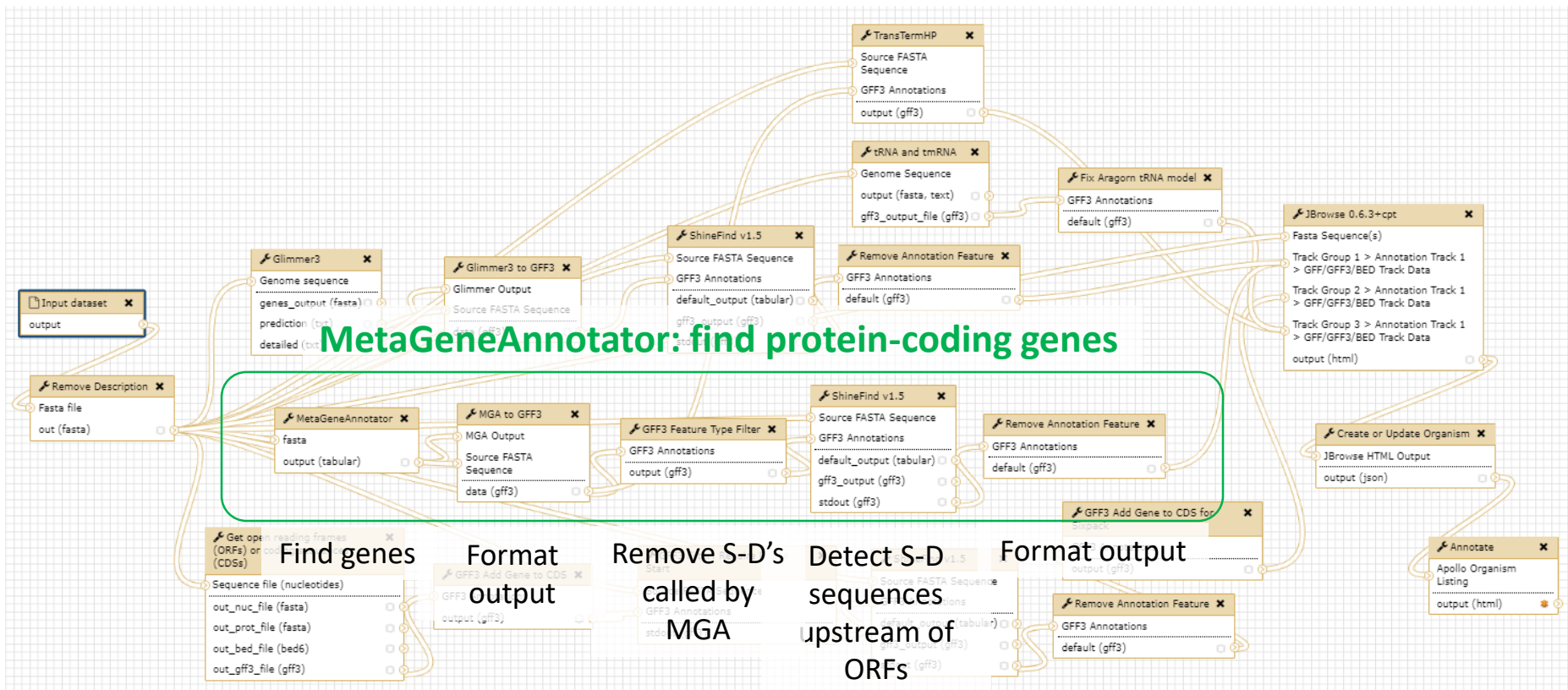
Format
output

Remove all
ORFs without
ATG, GTG or
TTG starts

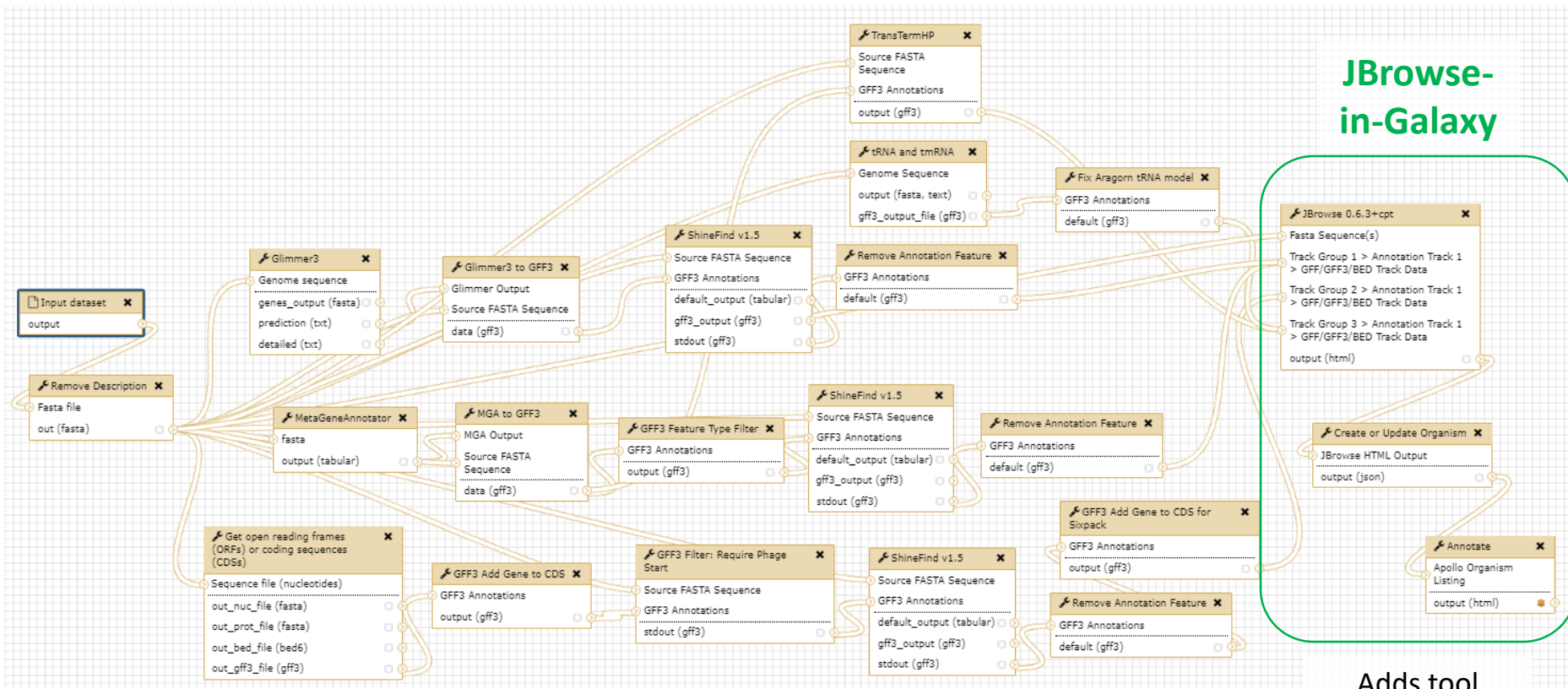
Detect S-D
sequences
upstream of
ORFs

Format output

The CPT Galaxy Structural Annotation workflow



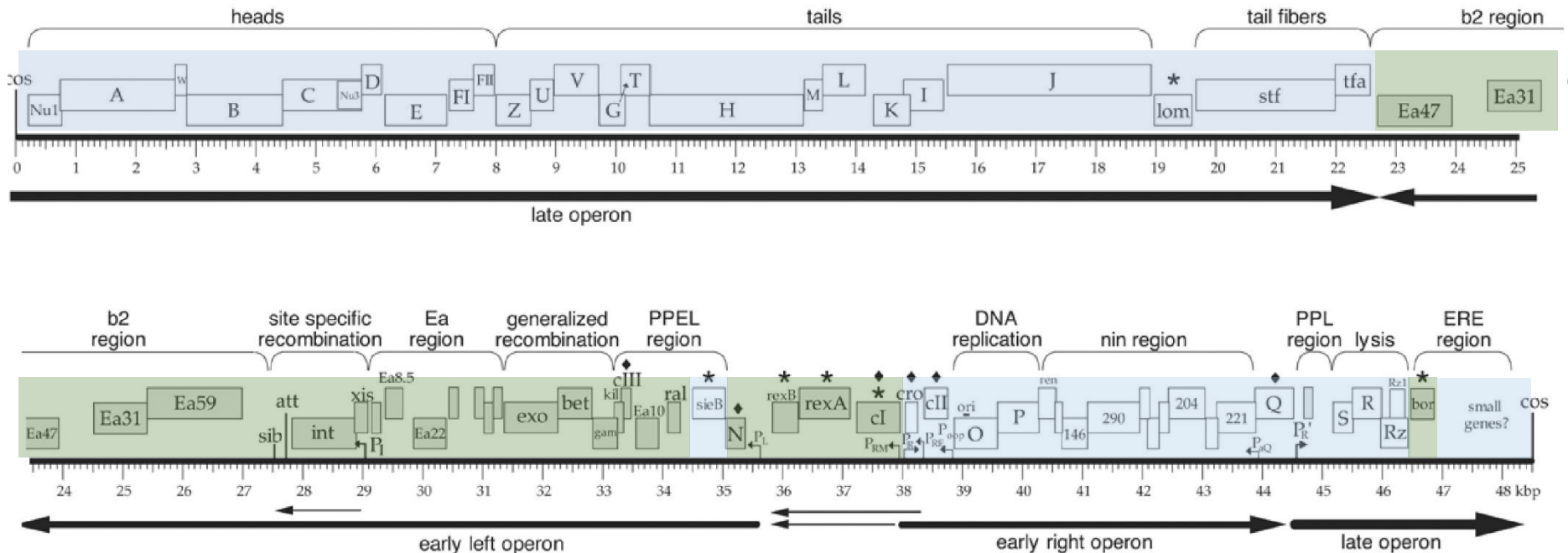
The CPT Galaxy Structural Annotation workflow



**JBrowse-
in-Galaxy**

Adds tool
outputs to
Jbrowse for
use in Apollo

Coding density and organization



- **Density**

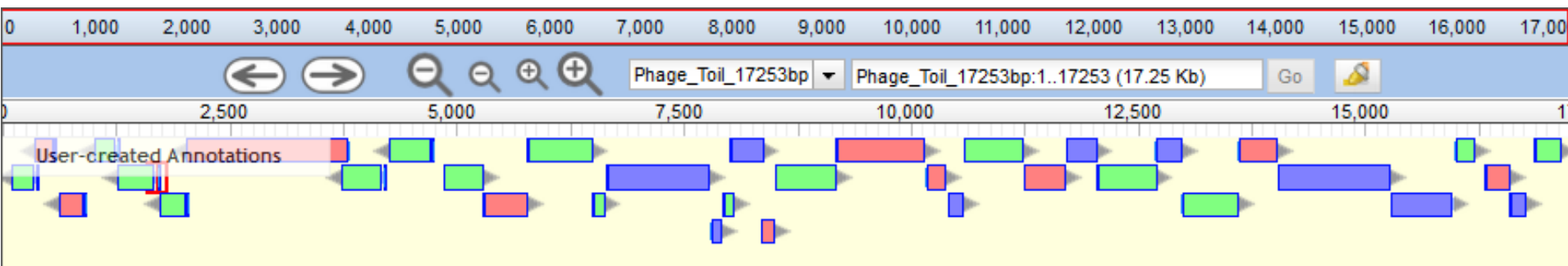
- Most phages have coding densities of >90%
- Most of the DNA contains some kind of feature: protein coding gene, tRNA, terminator, regulatory element, etc.
- These features are **tightly packed** and may even **overlap** if biologically possible

- **Transcriptional units**

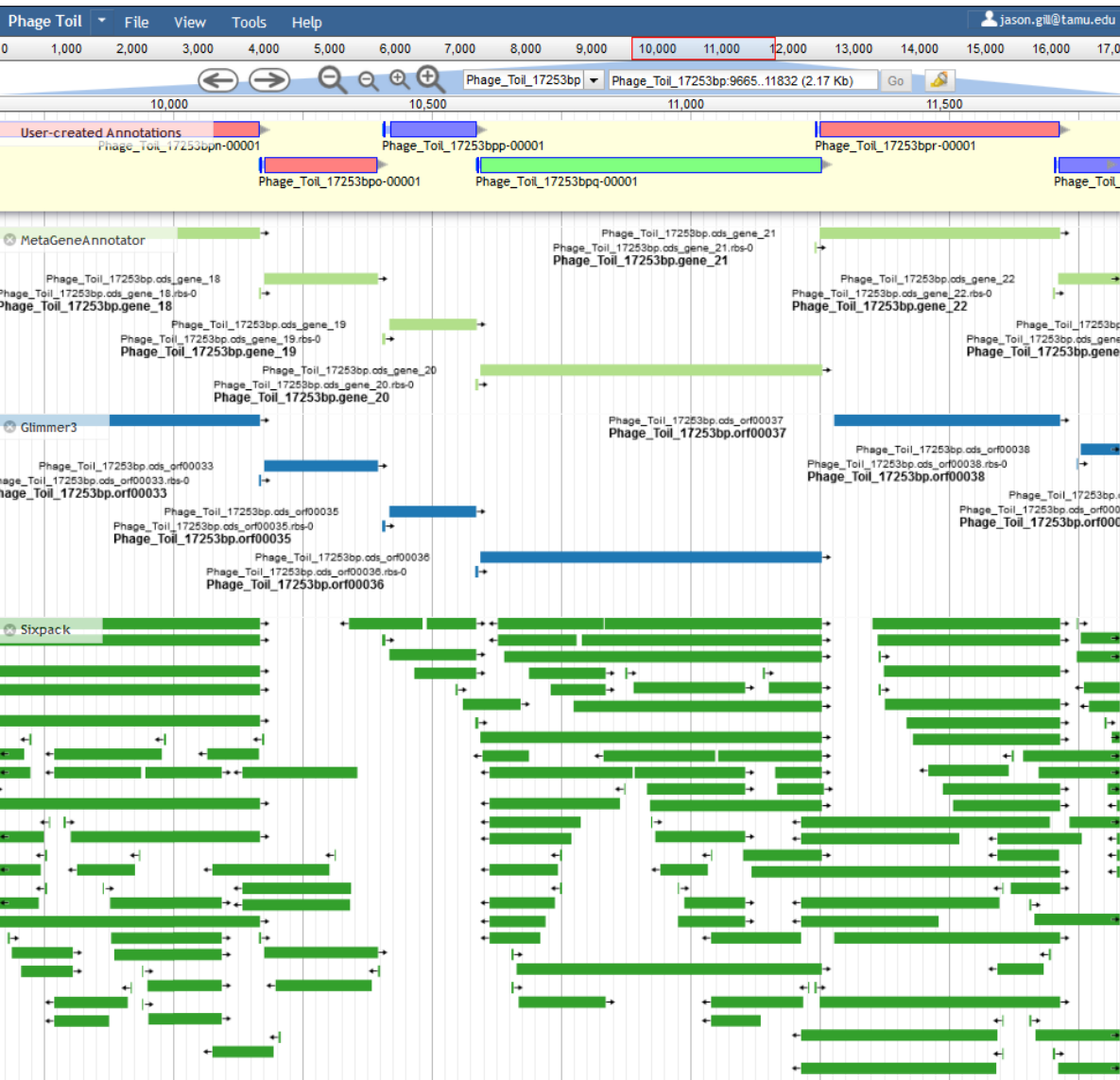
- Phage genes are translated from **polycistronic mRNA's**
- Genes tend to be arranged in groups on the plus or minus strand

General gene finding rules for phage

- Phages have high coding density
 - Genes tend to have minimal gaps between them or overlap slightly (up to ~5-8 aa)
 - Genes should **never** be embedded in each other on opposite strands
- Genes tend to be arranged into transcriptional units: blocks of genes on one strand or the other
- Start codons: ATG > GTG >> TTG
- Have recognizable TIR's, but only a few will have the full consensus S-D sequence AGGAGGT
- Most genes will encode proteins > 30 aa
- Sometimes there is no good-looking gene for a given DNA region and **that is OK**
 - There may be a regulatory element or some other function for that sequence



Calling genes from evidence tracks



- The programs **MetaGeneAnnotator** (MGA) and **Glimmer3** will detect most protein-coding genes
 - Use the outputs from these tools **first** before resorting to the naïve ORF caller (Sixpack)
- We use our own tool, **Shine Find**, to automatically detect S-D sequences upstream of predicted genes as a means of quality control
- The vast majority of “real” genes will have a valid Shine-Dalgarno sequence associated with them

Functional annotation: BLAST, Conserved Domains, and Annotations

Genome annotation

Predict which regions of DNA
encode proteins (CDS)

- Reading frame
- Coding start and stop
- Predicted amino acid sequence

STRUCTURAL

Predict the functions of
proteins

BLAST

Function based on
similarity to other
proteins with known
function

InterProScan

Function based on
peptide sequence
motifs

TMhmm

LipoP

FUNCTIONAL

Determining gene function

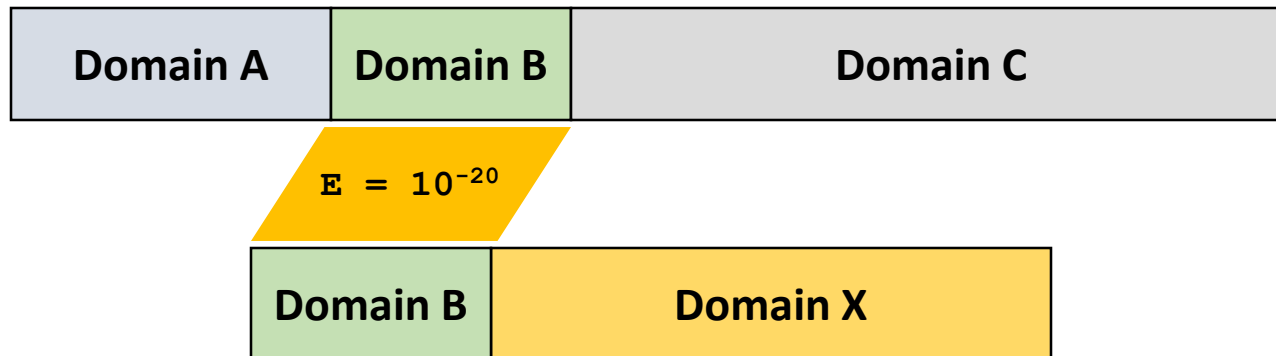
- Most functional prediction is conducted on protein sequence
 - DNA sequence similarity degenerates rapidly
 - Amino acid sequences are more constrained by the needs of maintaining protein function
- The CPT Galaxy Functional Workflow uses BLASTp and InterProScan for main annotation uses

BLASTp

- The CPT Galaxy Functional Workflow uses BLASTp against four databases
 - Canonical phage: Curated RefSeq records of well-studied phages (T4, T5, T7, lambda, N4, etc.)
 - SwissProt: The curated EMBL SwissProt database
 - TrEMBL: The total EMBL protein database
 - nr: The NCBI nr protein database, uncurated













Partial protein similarity can lead to misleading results

- Two different proteins can share a region of similarity if they share a functional domain
- E.g., both proteins may hydrolyze ATP but otherwise have different functions
- BLAST similarity or E-value can be misleading if there is a good match over part of a protein
- Alignment of the BLASTp result to the gene in Apollo shows partial matches

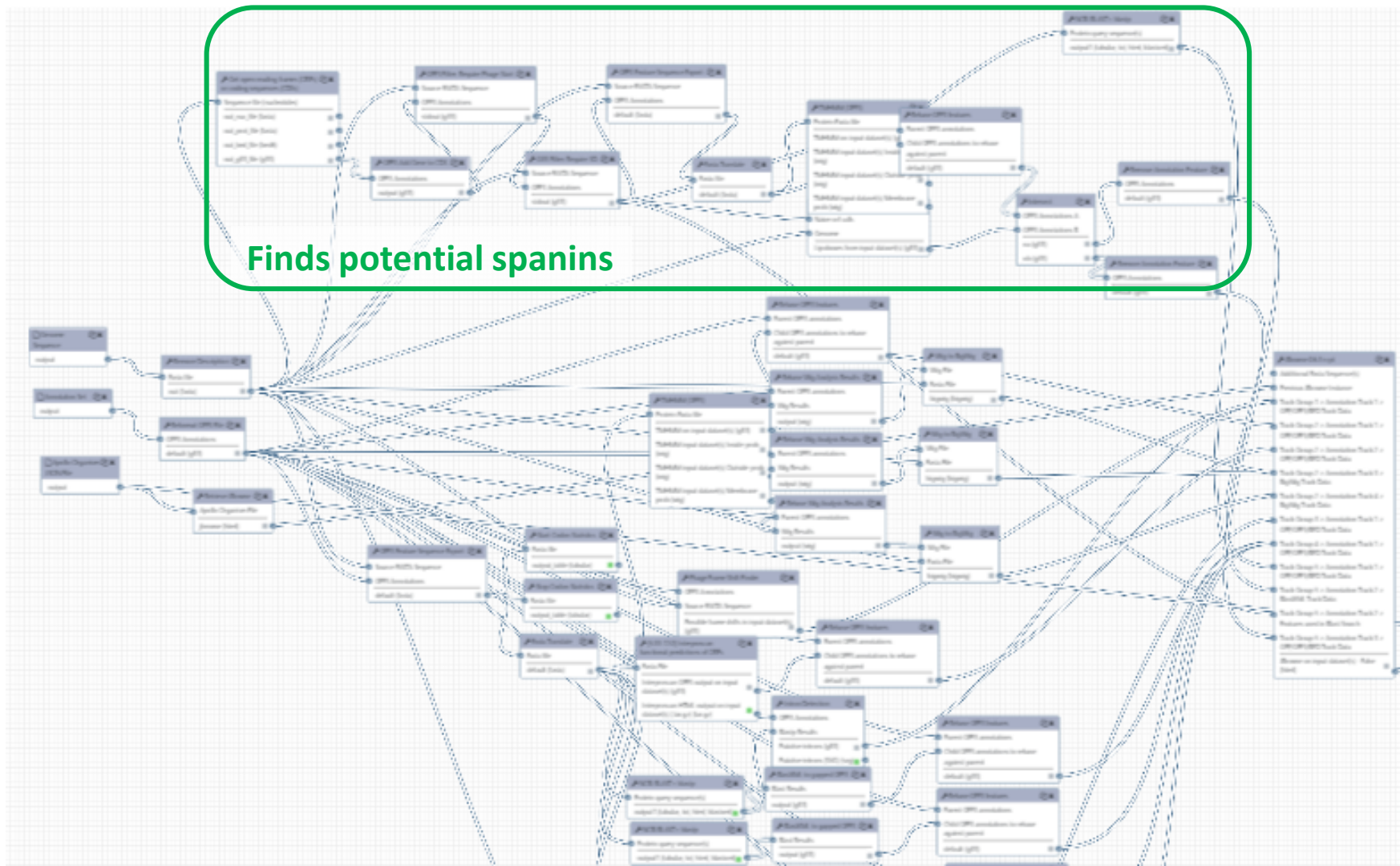




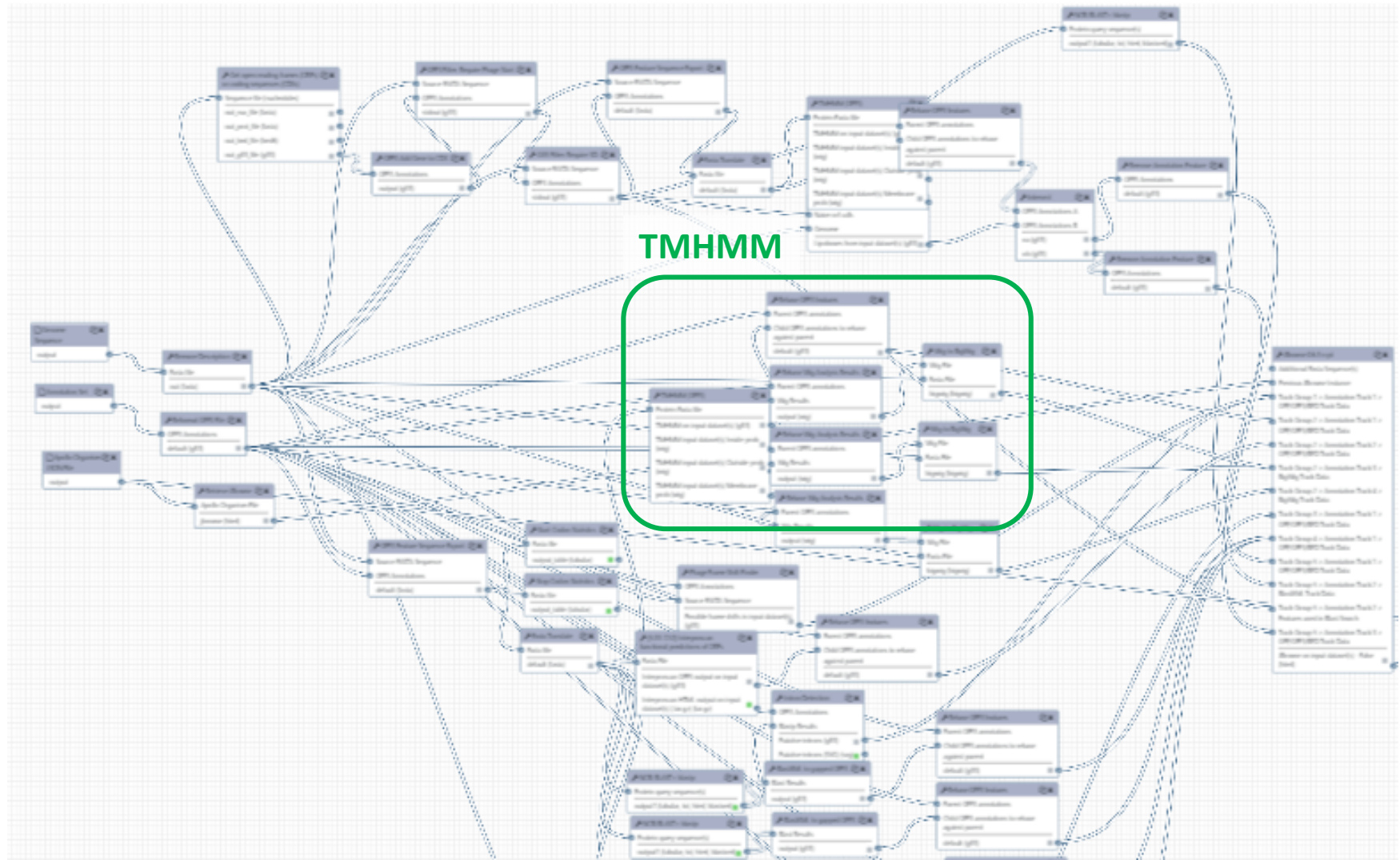
Conserved domains: InterProScan

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- 
- InterPro is hosted by EMBL-EBI and integrated into UniProt
 - InterPro integrates multiple conserved domain databases and assigns a single InterPro ID to related domains
 - InterProScan is the tool that searches a protein sequence against the member databases and detects similarity to conserved domains
 - Online tool only allows searching a single sequence
 - All UniProt records are automatically processed through InterProScan and InterPro domains are part of the protein record

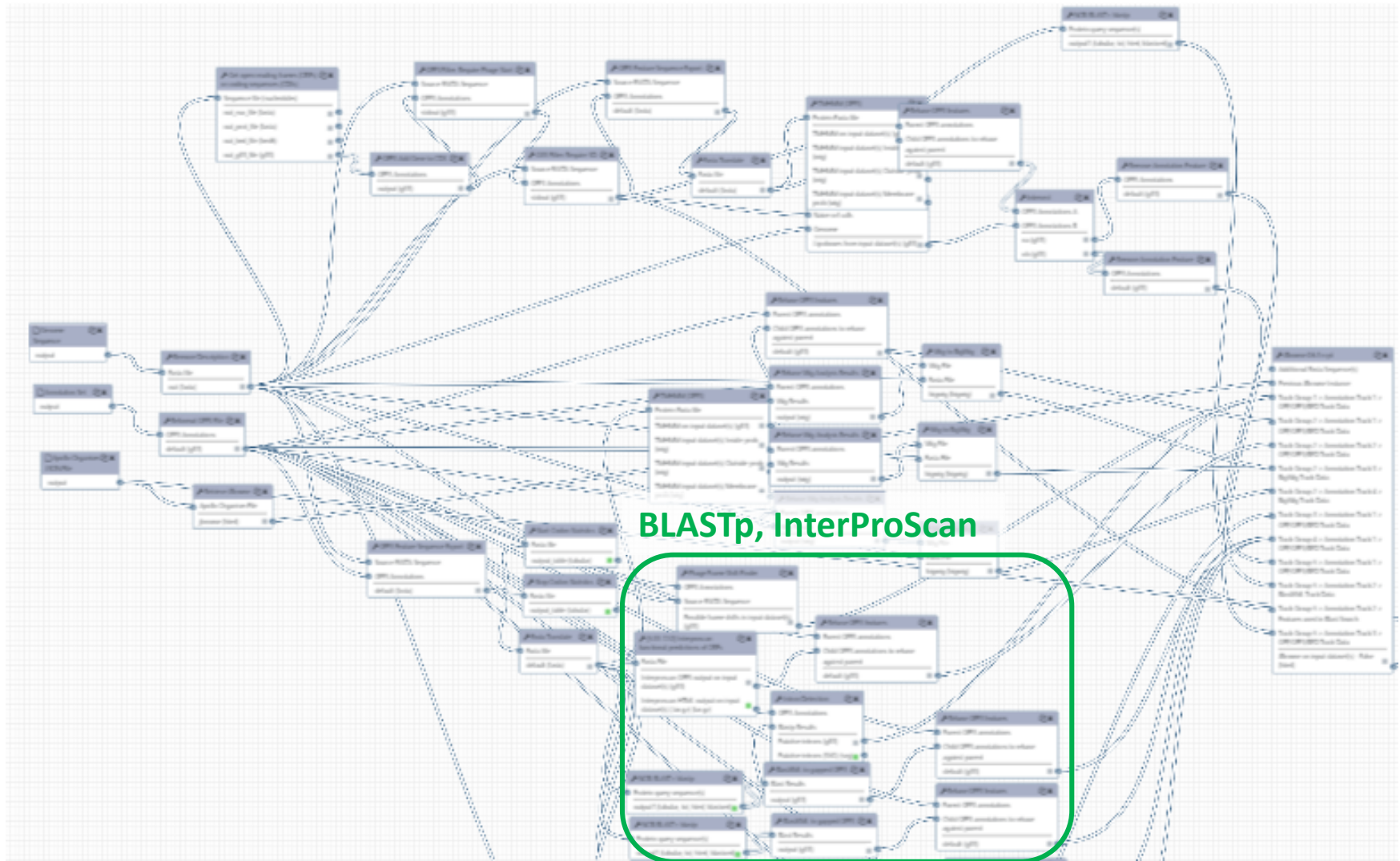
The CPT Galaxy Functional Annotation workflow



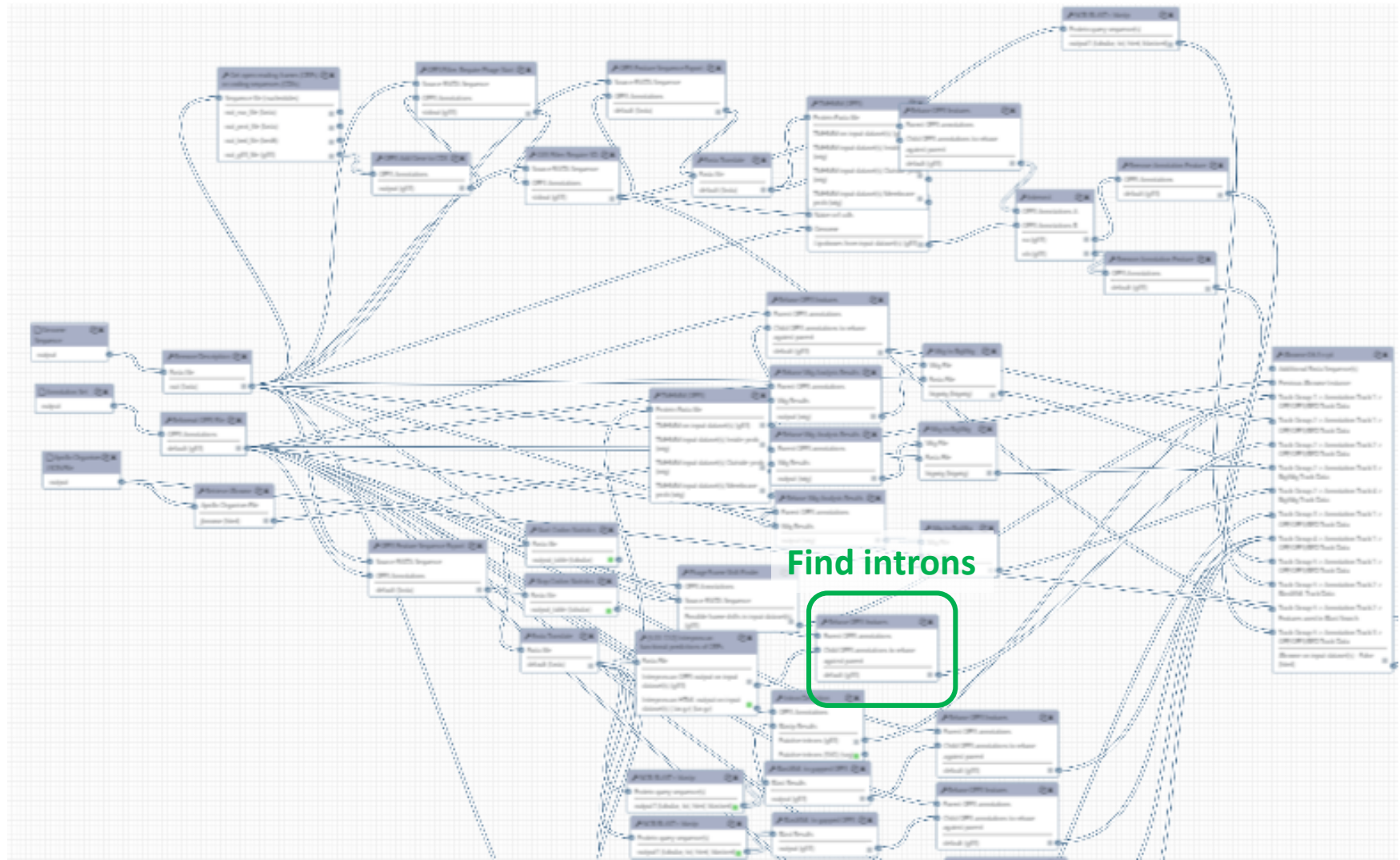
The CPT Galaxy Functional Annotation workflow



The CPT Galaxy Functional Annotation workflow

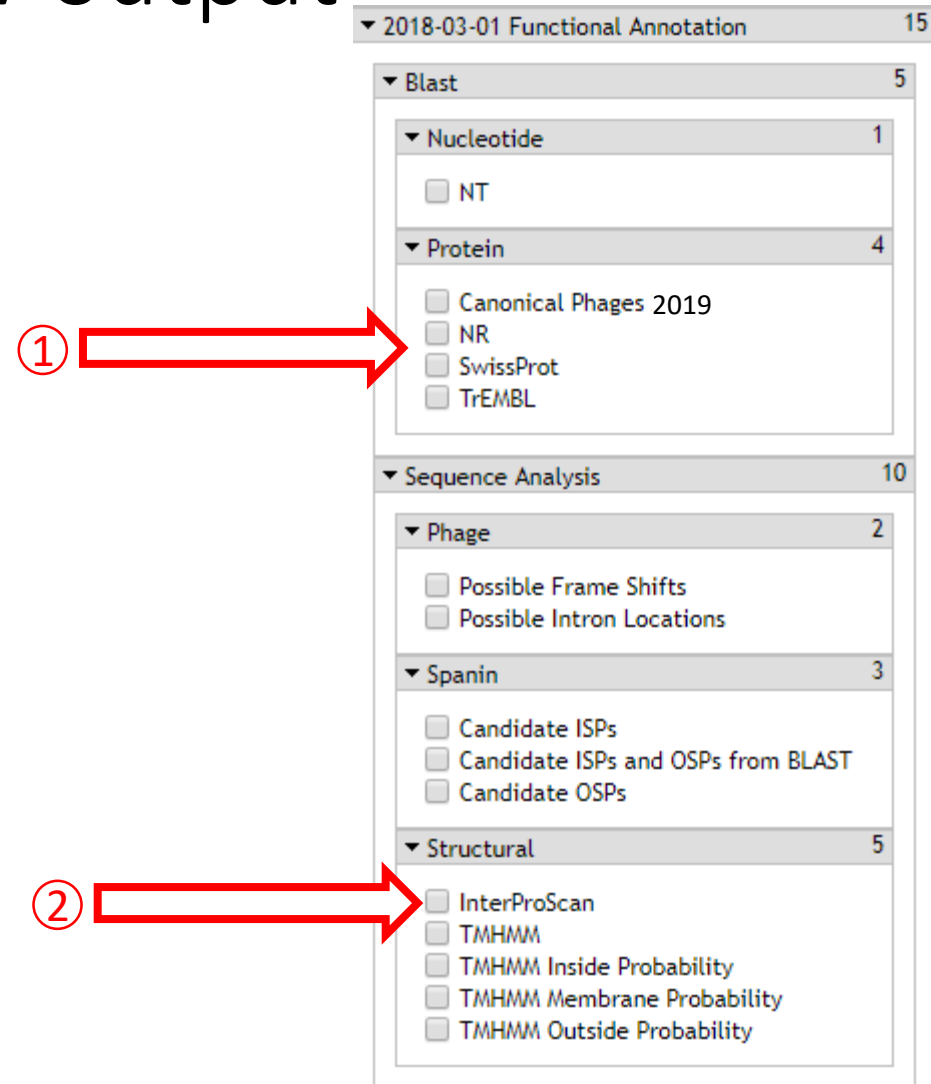


The CPT Galaxy Functional Annotation workflow

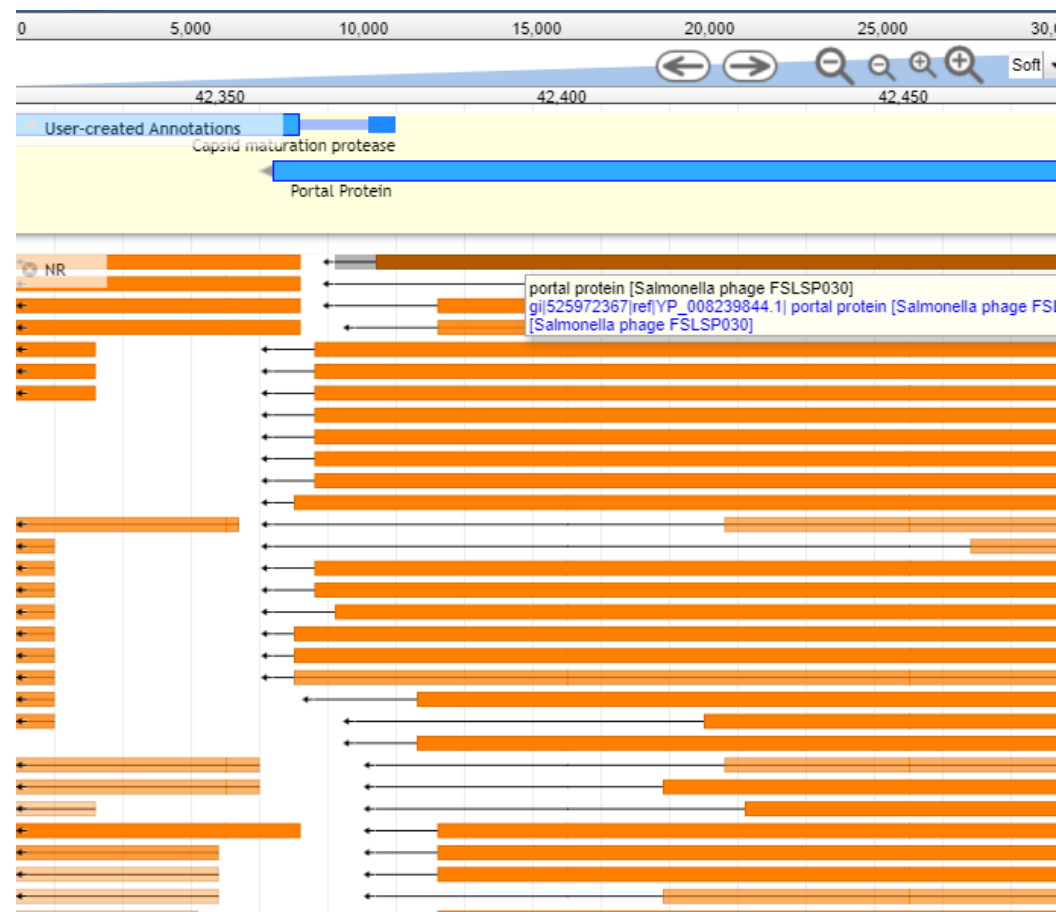


Functional workflow output

- The **functional workflow** runs many different analyses
- We will review these further in the hands-on session
- We will look at the most powerful general analyses, BLASTp and InterProScan



BLAST results in Apollo



- Like other evidence tracks, BLAST results will be aligned under each gene
- The **length** of the alignment is visualized by the length of the bar below the gene
 - Note that if the subject sequence is *longer* than your gene, it will be truncated so it doesn't overlap other genes
- The **E value** of the match is roughly visualized by the *intensity* of the color
- Hovering over a bar will preview the match info

Tools Help

protein_match

Primary Data

Name
lcl|NC_015296.1_prot_YP_004327457.1_86 [gene=phage-peptidoglycan binding] [protein=phage-encoded peptidoglycan binding protein] [protein_id=YP_004327457.1]
[location=complement(56259..57053)]

Type protein_match

Score 1.12276e-170

Description
gnl|BL_ORD_ID|2180 lcl|NC_015296.1_prot_YP_004327457.1_86 [gene=phage-peptidoglycan binding] [protein=phage-encoded peptidoglycan binding protein] [protein_id=YP_004327457.1]
[location=complement(56259..57053)]

Position ISA:107034..107825 (+ strand)

Length 792 bp

Attributes

Accession 2180

Blast_align_length 263

Blast_bits 473.011

Blast_frame 0

Blast_gaps 0

Blast_identity None

Blast_positives 249

Blast_query_end 263

Blast_query_start 1

Blast_sbjct_end 263

Blast_sbjct_start 1

Blast_score 1216.0

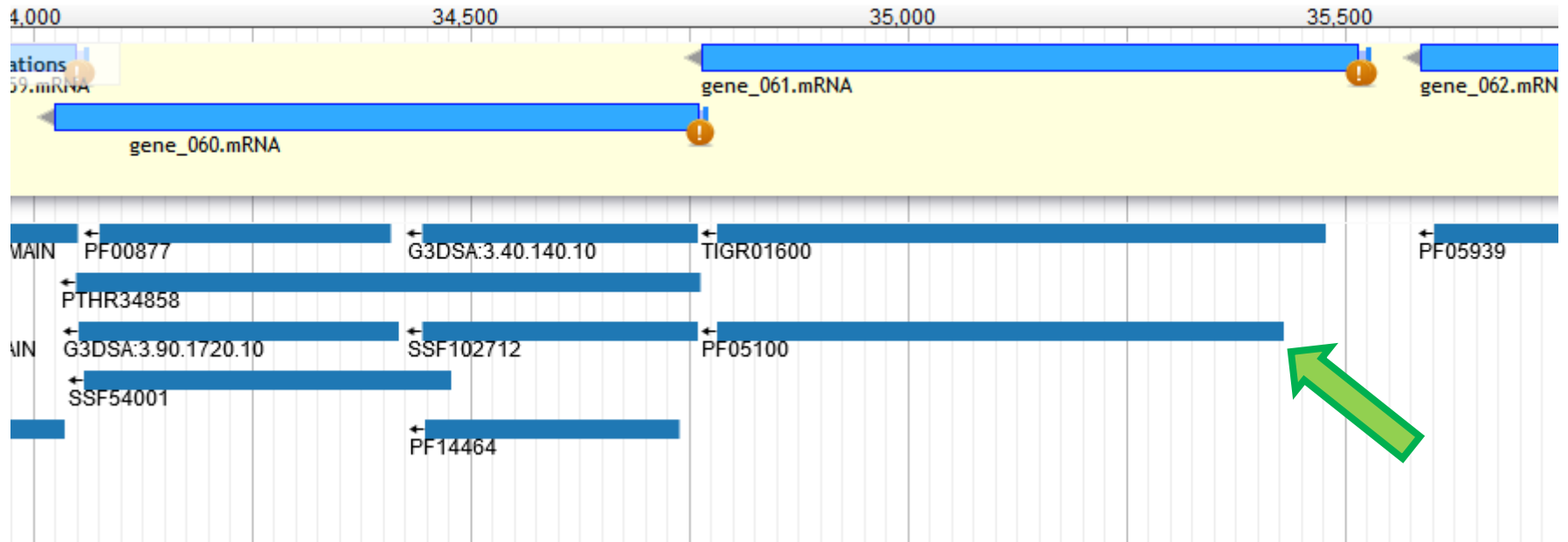
Blast_strand None

Description
lcl|NC_015296.1_prot_YP_004327457.1_86 [gene=phage-peptidoglycan binding] [protein=phage-encoded peptidoglycan binding protein] [protein_id=YP_004327457.1]
[location=complement(56259..57053)]

BLAST hit information

- E-value: lower is better, also reflected in shading of the feature in the track
- Description: may be informative, depends on the quality of the annotation in the protein record used to make the database
- The accession number (underlined) can be used to find the protein record in the database

InterProScan results in Apollo



- InterProScan searches for conserved domains in member databases
- These hits predict protein function based on conserved domains, beware of domain swapping!

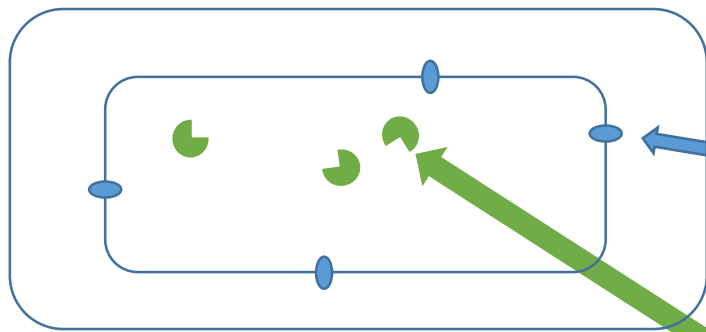
Specialty analyses

Lysis genes

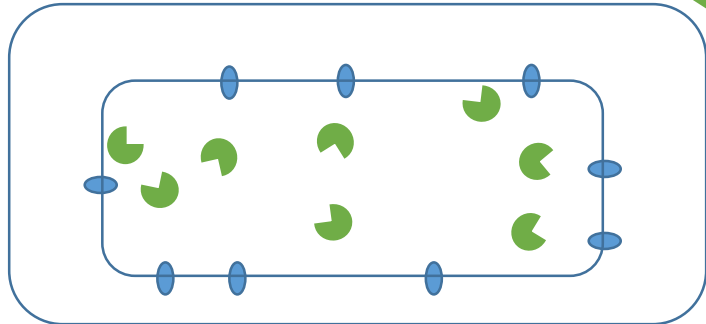
Introns

Frameshifts

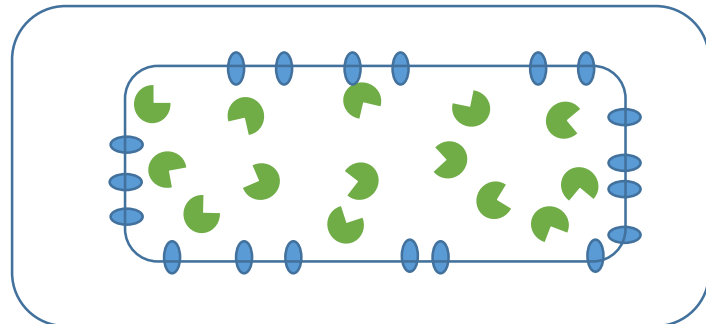
Canonical holin /endolysin lysis



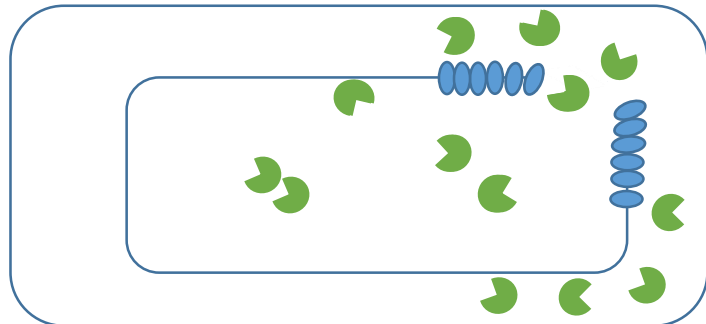
Holin accumulates in IM,
mobile and harmless



Endolysin accumulates in
cytoplasm, fully active



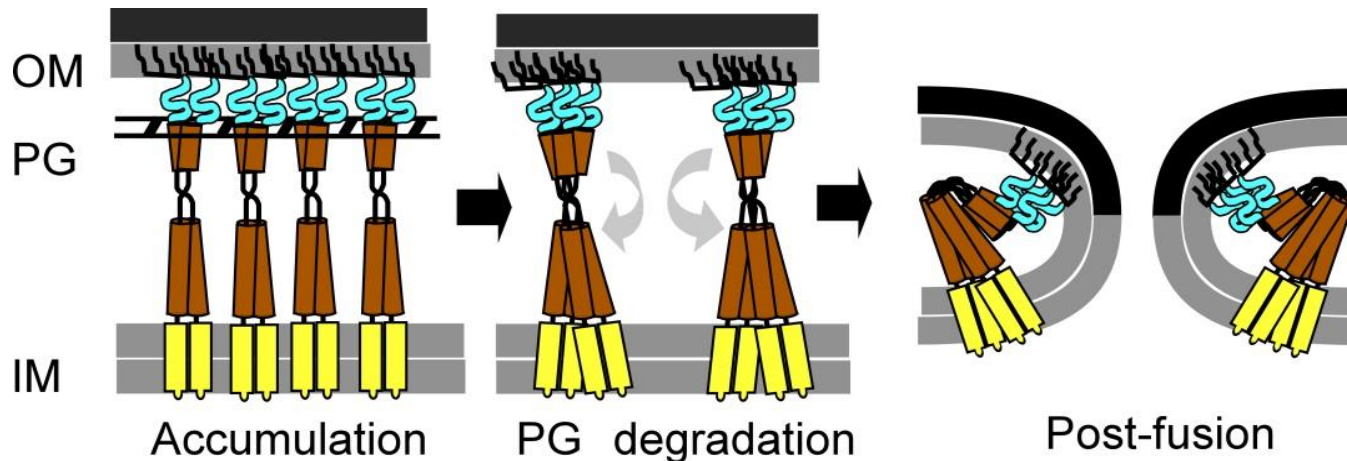
At a the programmed time, holin
triggers to form massive **“holes”**
(average 350 nm for lambda)



Endolysin escapes through holes
& attacks peptidoglycan

Spanin complex

- After holin triggering, endolysin is released to degrade peptidoglycan
- In Gram –ve hosts, a third component, the spanin complex, disrupts the outer membrane
- The canonical spanin is 2 components: an **inner membrane** protein with an N-terminal TMD, and an **outer membrane** lipoprotein tethered to the inner leaflet of the OM by a lipid anchor

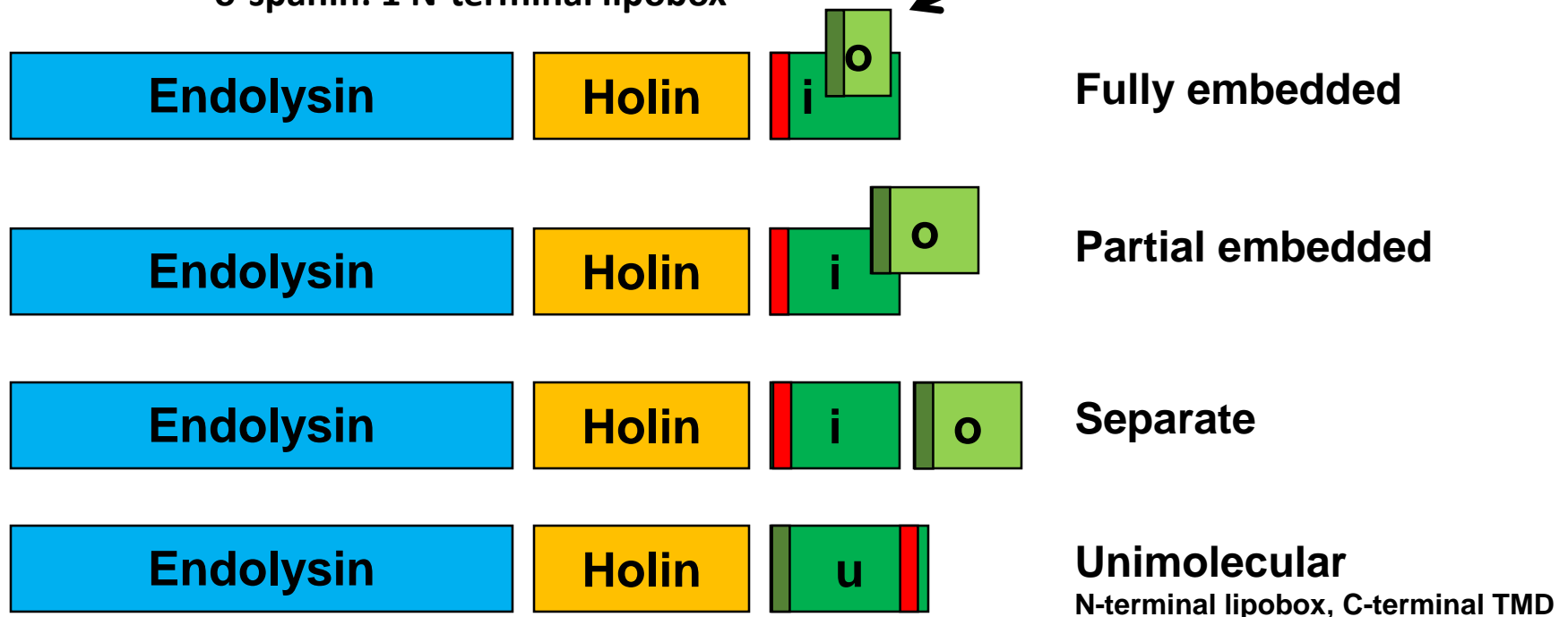


Lysis genes

- Lysis genes are often found co-localized in **lysis cassettes**

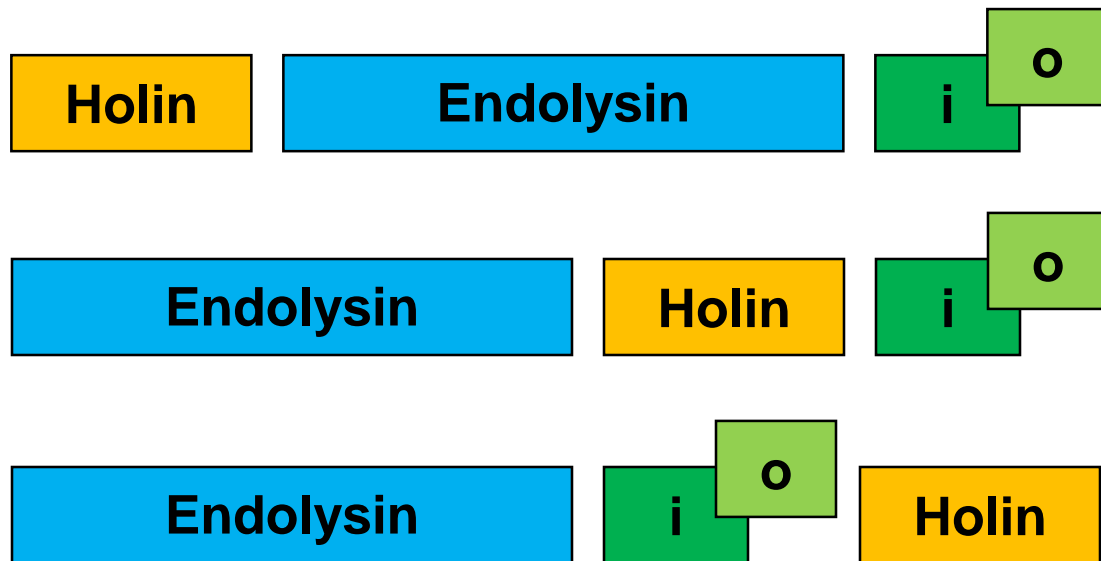
- **Endolysin**: conserved domains or BLAST homology
- **Holins**: small, 1 or more TMD's
- **Spanins**: Adjacent, partially or fully embedded
 - i-spanin: 1 N-terminal TMD
 - o-spanin: 1 N-terminal lipobox

Embedded genes not found in normal structural annotation!



Lysis cassettes

- Gene order is not always conserved within the lysis cassette

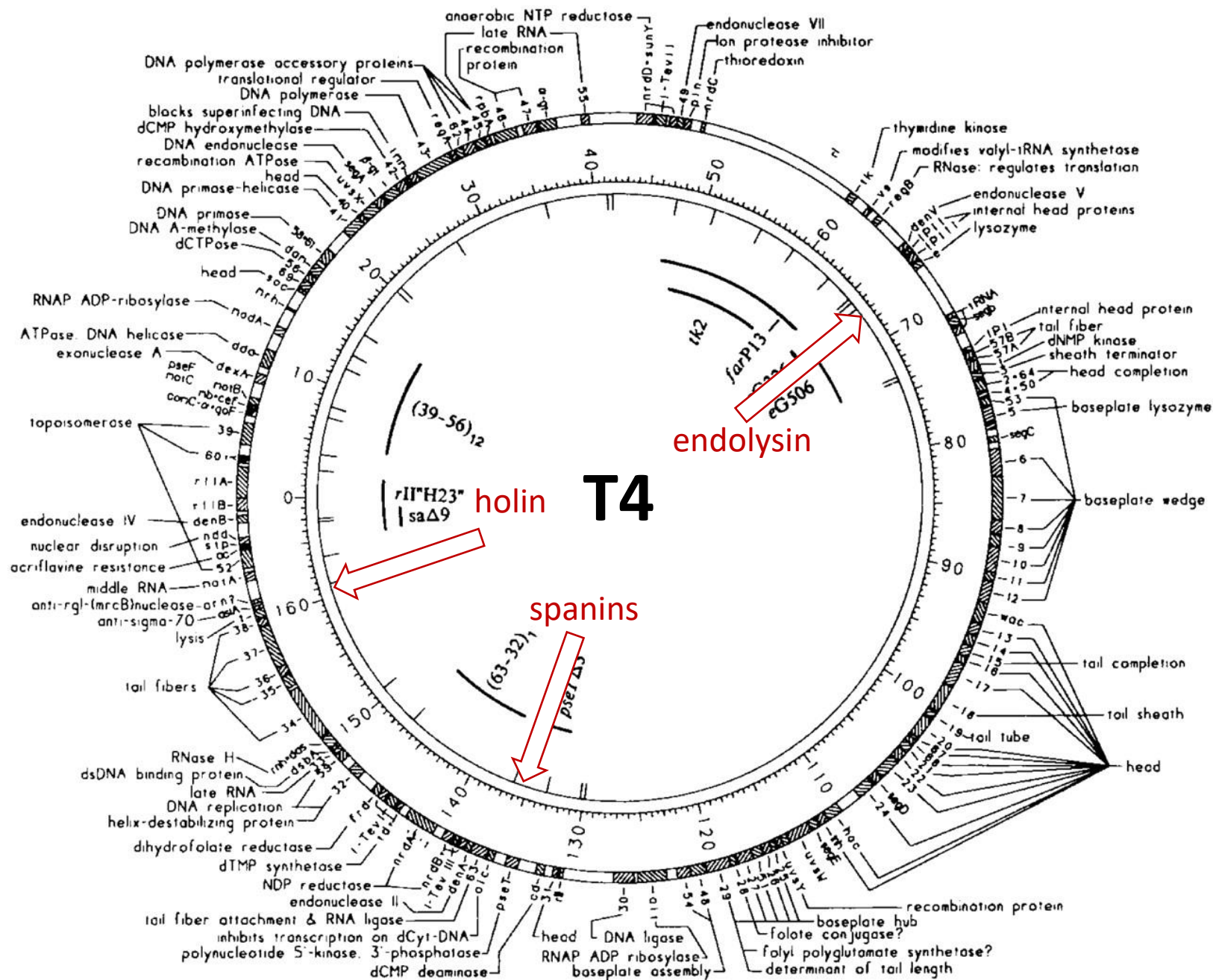


Etc., etc.

Distributed lysis genes

- Lysis genes are not always in discrete cassettes
- Prevalent in larger genomes like T4





Holin finding

▼ 2017-03-29 Functional Annotation 14

▼ Blast 4

▼ Nucleotide 1

☐ NT

▼ Protein 3

☐ Canonical Phages

☐ NR

☐ UniRef90

▼ Sequence Analysis 10

▼ Phage 2

☐ Possible Frame Shifts

☐ Possible Intron Locations

▼ Spanin 3

☐ Candidate ISPs

☐ Candidate ISPs and OSPs from BLAST

☐ Candidate OSPs

▼ Structural 5

☐ InterProScan

☐ TMHMM

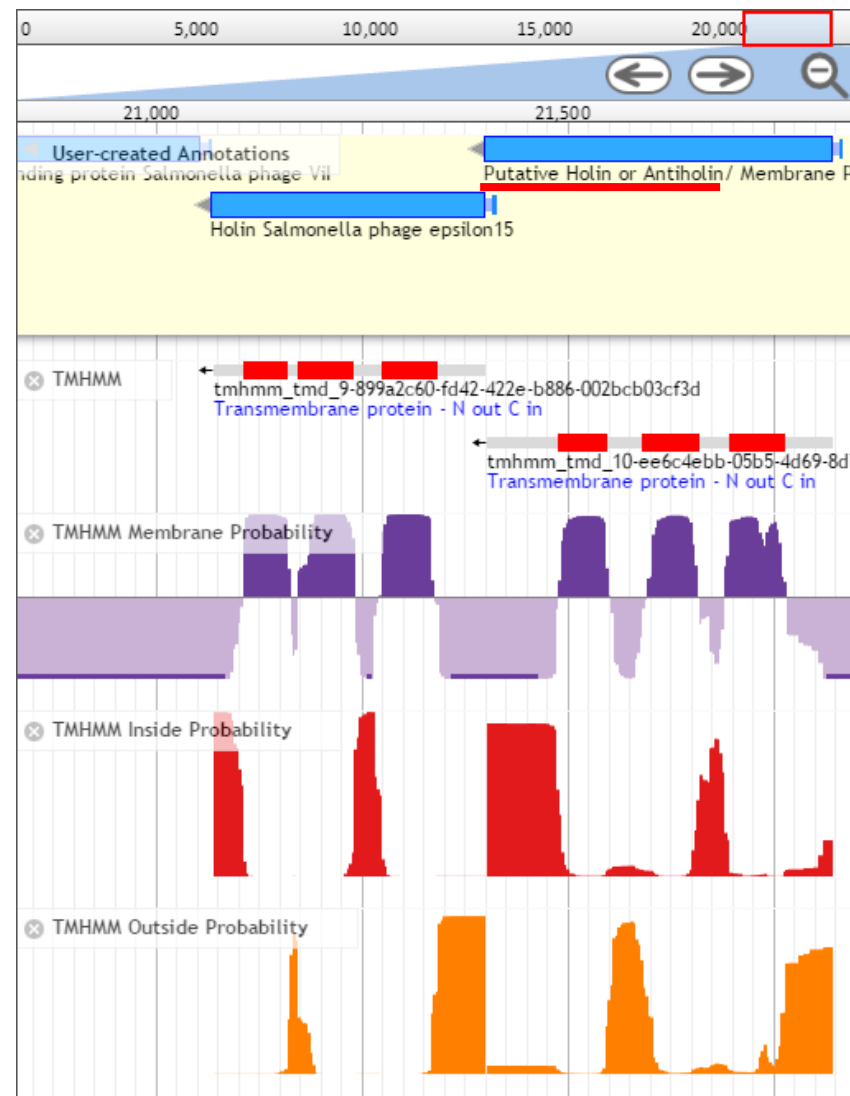
☐ TMHMM Inside Probability

☐ TMHMM Membrane Probability

☐ TMHMM Outside Probability

- Small, TMD-containing proteins
- Look for holins first next to the endolysin gene
- Must have 1 or more TMDs
- May have homology to another holin by BLAST but this is not common

Holin finding



- **Display tracks for TMHMM**

- TMHMM: number and location of predicted TMD's
- Membrane: actual scores from TMHMM plotted to genome
- Inside: Probability this region is in the cytoplasm
- Outside: Probability this region is in the periplasm/extracellular

- **Most likely** location is adjacent to the endolysin

- If no holin candidate near endolysin, lysis genes may be distributed
- Probably will not be identifiable unless there is *only one* small TMD-containing protein in the whole genome (unlikely), or BLAST homology to a known holin (also unlikely)

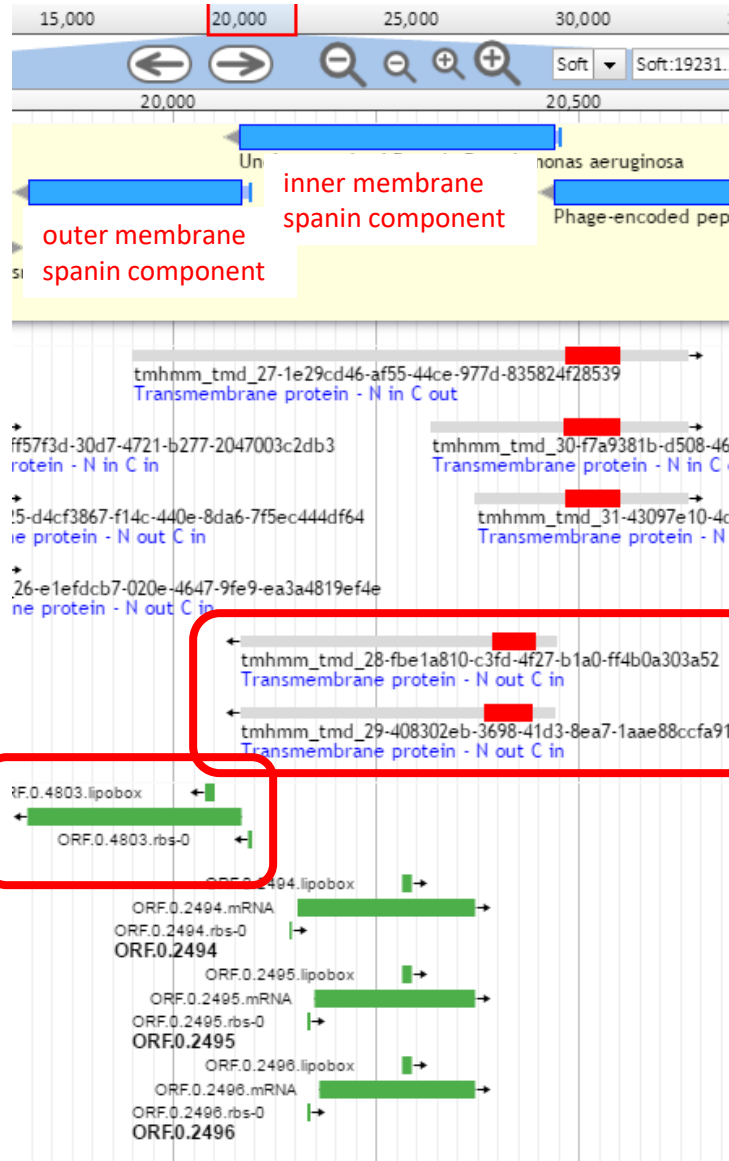
Spanin finding

2017-03-29 Functional Annotation 14

- Blast 4
 - Nucleotide 1
 - ☐ NT
 - Protein 3
 - ☐ Canonical Phages
 - ☐ NR
 - ☐ UniRef90
- Sequence Analysis 10
 - Phage 2
 - ☐ Possible Frame Shifts
 - ☐ Possible Intron Locations
 - Spanin 3
 - ☐ Candidate ISPs
 - ☐ Candidate ISPs and OSPs from BLAST
 - ☐ Candidate OSPs
 - Structural 5
 - ☐ InterProScan
 - ☐ TMHMM
 - ☐ TMHMM Inside Probability
 - ☐ TMHMM Membrane Probability
 - ☐ TMHMM Outside Probability

- Candidates from BLAST
 - Low sequence conservation in spanins
- Candidate ISPs (i-spanin)
 - Naive ORF calls analyzed by TMHMM
- Candidate OSPs (o-spanin)
 - Naive ORF calls analyzed for N-terminal lipobox signals

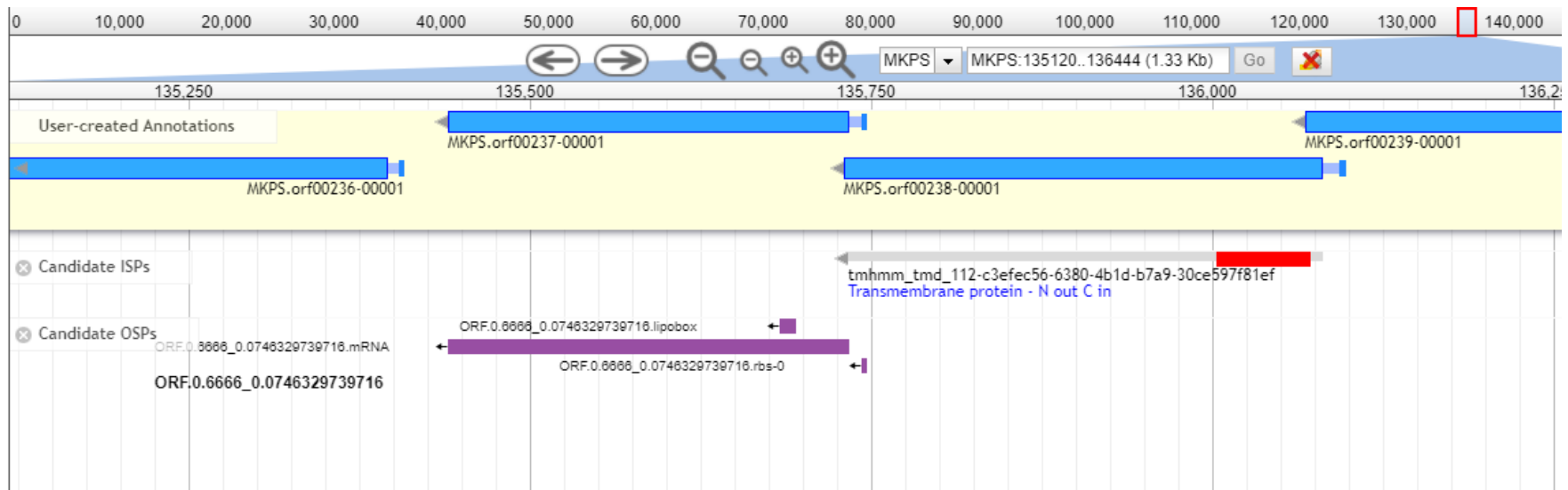
Spanin finding



- Likely spanin gene pairs
 - 1 protein with N-terminal TMD (top)
 - 1 protein with N-terminal lipobox (bottom)
 - Adjacent or o-spanin embedded in i-spanin
 - i-spanin is never embedded in o-spanin
- OSP tool conducts naive ORF calls, should find embedded genes not found during structural annotation

Spanin finding

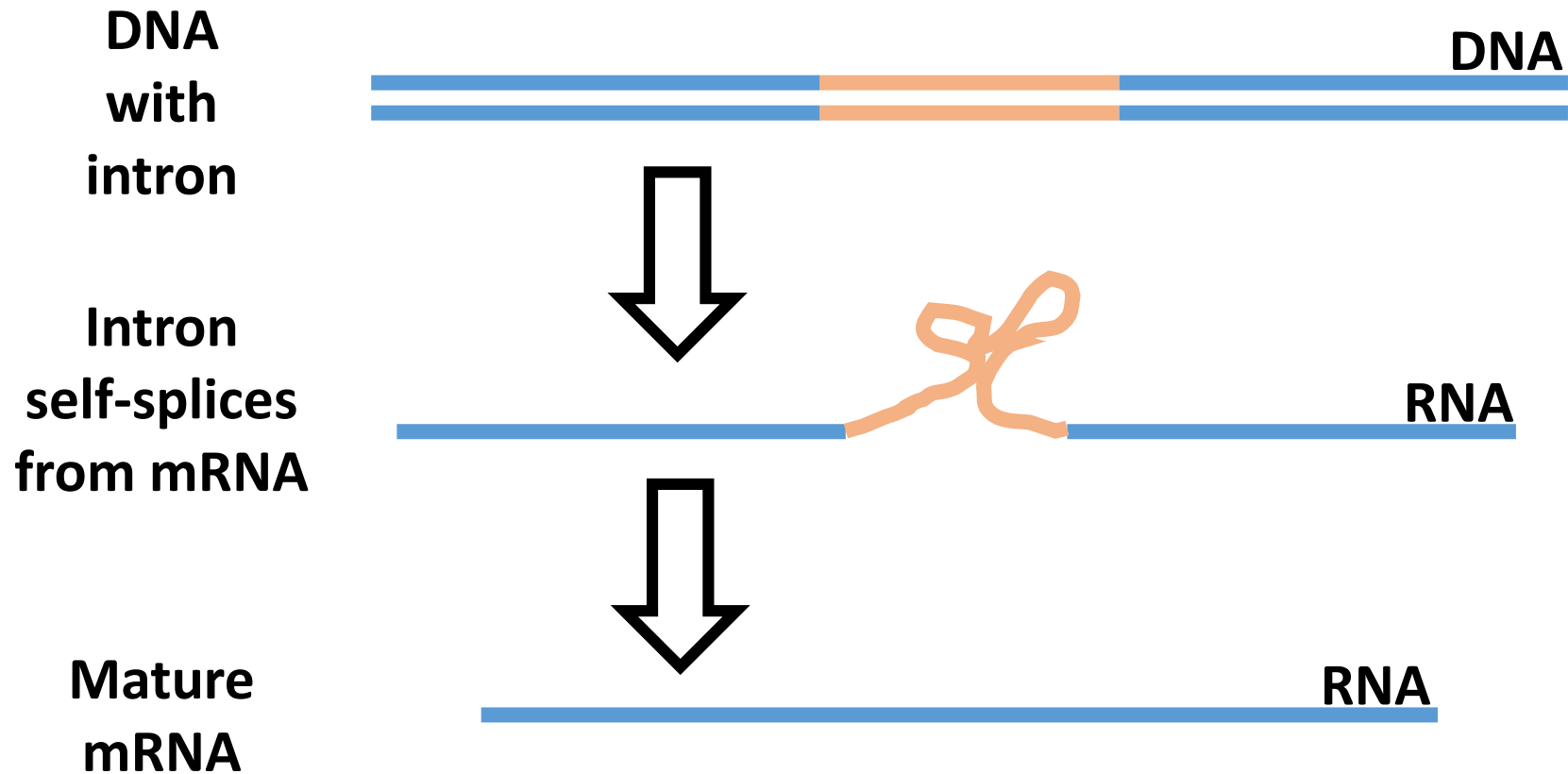
- Likely candidate spanin pair, adjacent non-embedded genes
- At least one of the genes was found during structural annotation
- Candidate ISP has 1 N-terminal TMD
- Candidate OSP has SPII signal and is immediately downstream of ISP



Introns

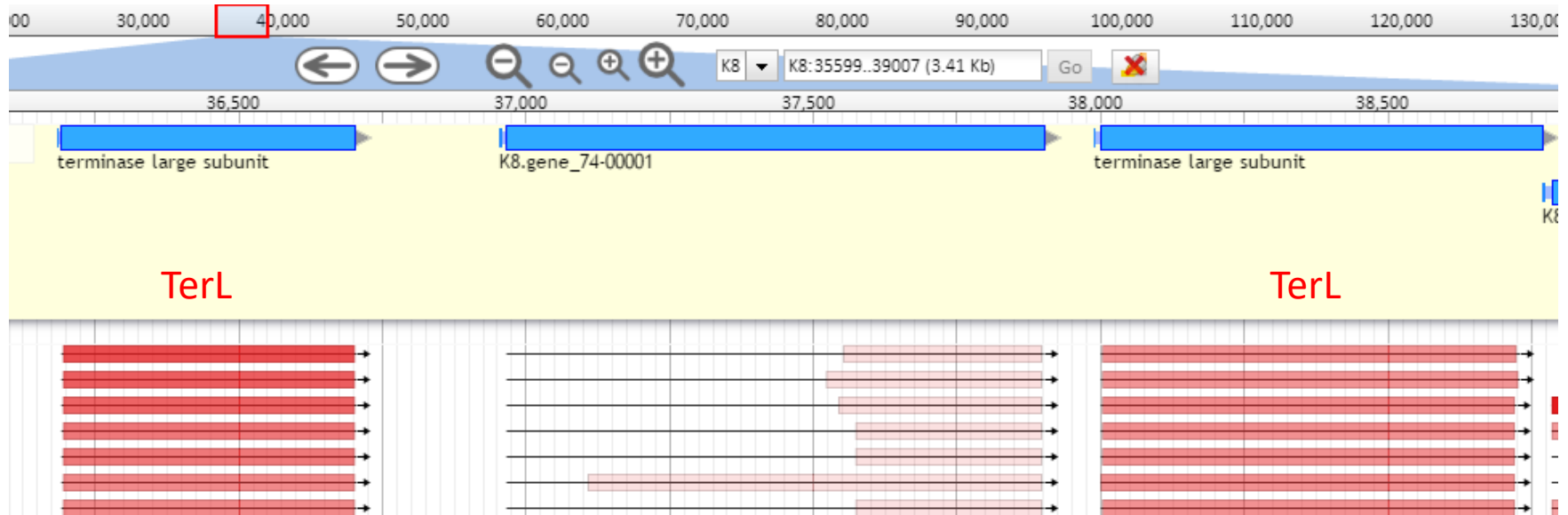
- An intron is an extra section of DNA that interrupts the protein-coding sequence of a gene
- This sequence has *ribozyme* activity and splices itself out of the mRNA, leaving an intact message and a free intron RNA
- Introns often (but not always) contain a homing endonuclease gene
- These are often found in *essential* genes, and often in genes involved in DNA metabolism

Introns are self-splicing elements



Introns

- Two or more genes that BLAST to the same protein could indicate an intron
- You can look at the BLAST results: do the two genes in your genome align to different portions of the same protein?



Intron finding track

2017-03-29 Functional Annotation 14

▼ Blast 4

▼ Nucleotide 1

☐ NT

▼ Protein 3

☐ Canonical Phages

☐ NR

☐ UniRef90

▼ Sequence Analysis 10

▼ Phage 2

☐ Possible Frame Shifts

☐ Possible Intron Locations

▼ Spanin 3

☐ Candidate ISPs

☐ Candidate ISPs and OSPs from BLAST

☐ Candidate OSPs

▼ Structural 5

☐ InterProScan

☐ TMHMM

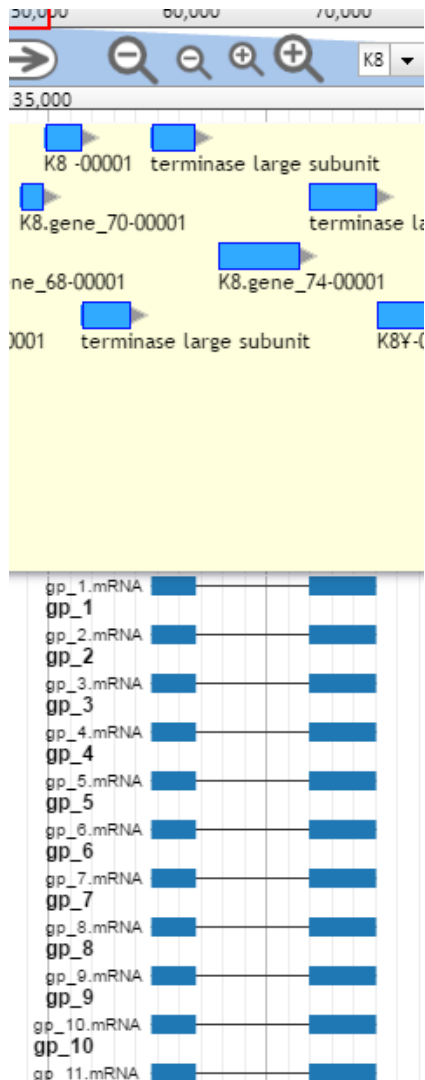
☐ TMHMM Inside Probability

☐ TMHMM Membrane Probability

☐ TMHMM Outside Probability

- Will highlight genes that may have been disrupted by introns
 - Difficult to check all BLAST hits manually
- Searches BLAST results for nearby genes that BLAST to the same proteins
- Only works if the intron-disrupted gene has non-disrupted homologs in the database

Intron finding



- Searches BLAST results for nearby genes that match to the same protein
- Will highlight possible genes that have been disrupted by introns
- May have a CDS within the intron, or not
- CDS may have a HNH or GIY-YIG domain, or not

Submitting to GenBank

- GenBank submissions are handled through the BankIt website so the genome must be exported from Galaxy/Apollo
- Final genome is retrieved into Galaxy, converted to GenBank (.gbk) format and renumbered to add the /locus_tag features
- Final proofing and conversion to 5-column table handled by Sanger Artemis
 - <https://www.sanger.ac.uk/science/tools/artemis>
 - Artemis handles all GenBank feature keys and reads GenBank formatted files directly

Comparative genomics

- Placing your phage into the general context of other known phages
- Proposing a phage genus or species if possible
- Most related phage
 - Highest similarity among all organisms in the nr database
- Most related type phage
 - Highest similarity to a known phage type

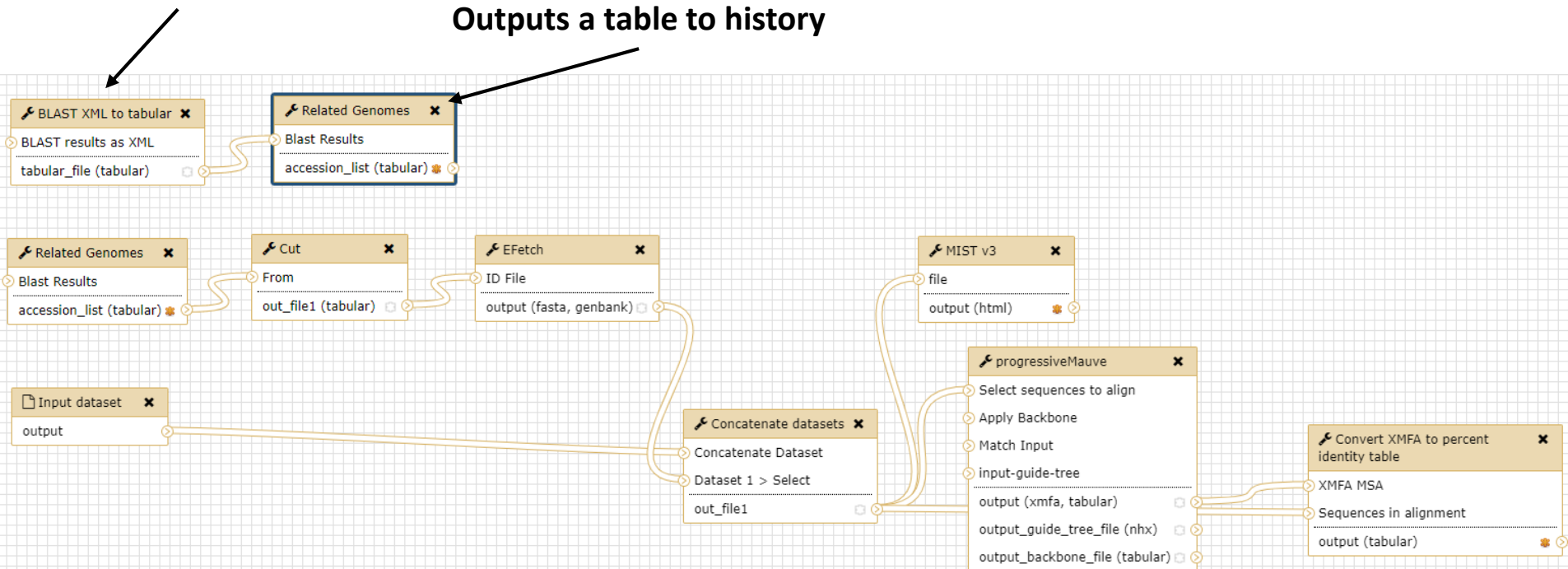
Comparative genomics

- The outputs of a comparative workflow will appear in Galaxy
 - The workflow will analyze BLASTp and BLASTn results against **nr** and **nt** and present the top five results
 - Will then run a dotplot comparison on DNA sequences
- **Nucleotide percent identity table**
 - Global DNA:DNA identity, listed by GenBank accession
- **MIST dotplot**
 - Low-resolution alignment of DNA sequence
- **Top BLASTp hits**
 - Provides accession, name, and number of matching proteins at $E < 0.001$

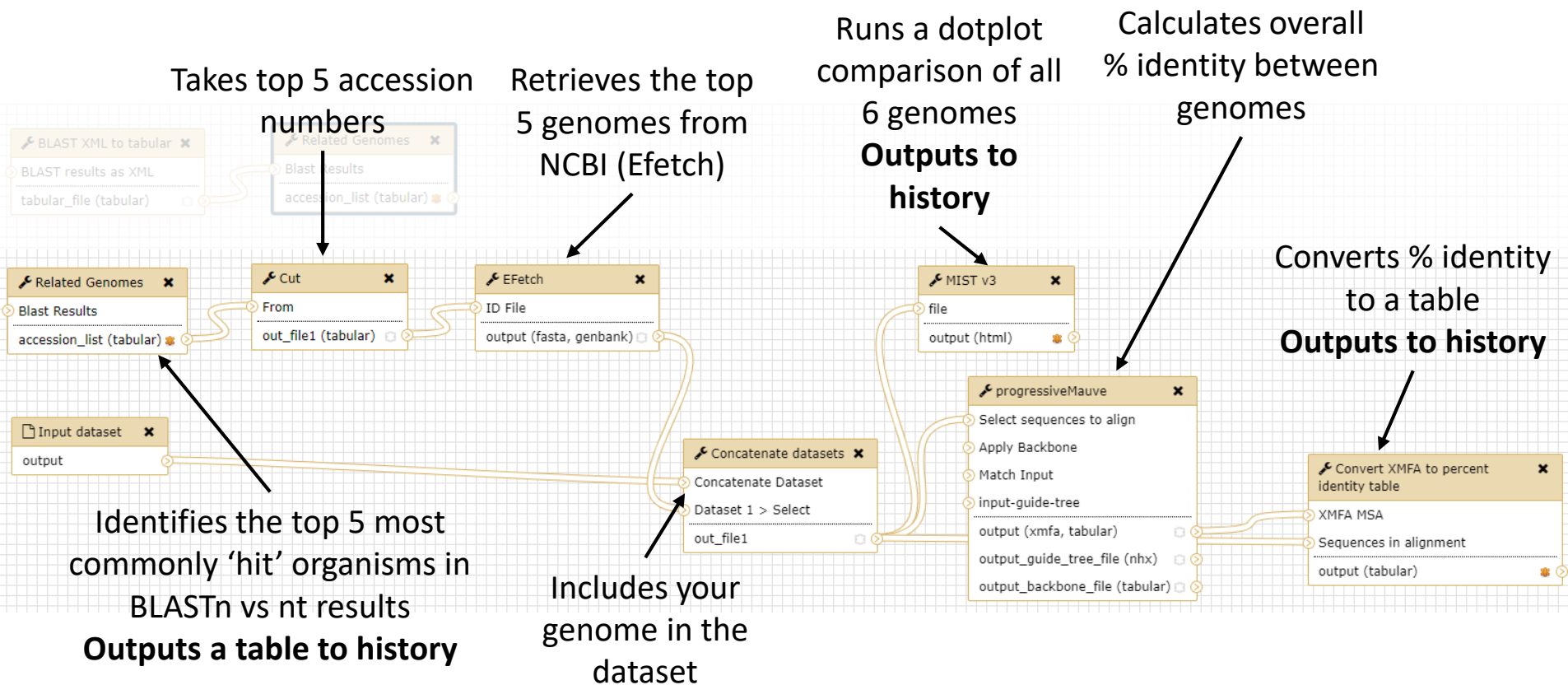
Comparative genomics workflow (protein)

Converts BLASTp
vs. nr results to a
table format

Identifies the top 5 most
commonly 'hit' organisms in
BLASTp results
Outputs a table to history



Comparative genomics workflow (DNA)



Phage comparative genomics (v1.5)

Workflow: imported: Phage comparative genomics (v1.5)

✓ Run workflow

History Options

Send results to a new history

Yes No

1: BLAST XML to tabular Convert BLAST XML output to tabular (Galaxy Version 0.1.01)

BLAST results as XML

73: Canonical Phages
72: SwissProt
71: TrEMBL
70: NR

BLASTp results in

XML format

Output format

Tabular (extended 25 columns)

Job Post Actions

Hide output 'tabular_file'.

2: Related Genomes based on nucleotide-blast results (Galaxy Version 1.2)

Blast Results

54: NT

BLASTn results in tabular

format (may have to unhide)

TSV/tabular (25 Column)

3: Input Dataset

44: Sequence(s) from Apollo

Phage DNA sequence in

FASTA format

Phage comparative genomics (v1.5)

- Inputs
 - BLASTp output should be labeled “NR”
 - BLASTn output should be labeled “NT” (may have to unhide)
 - Phage DNA sequence should be labeled “Sequences from Apollo”

70: NR

NCBI Blast XML data
format: **blastxml**, database: ?

```
<?xml version="1.0"?>
<!DOCTYPE BlastOutput PUBLIC "-//NCBI/
<BlastOutput>
<BlastOutput_program>blastp</BlastOutput
<BlastOutput_version>BLASTP 2.2.31+</B/
```

54: NT

652 lines
format: **tabular**, database: ?

1	2
Mushu2_2018	gi 827196798 gb KR011062.1
Mushu2_2018	gi 827196798 gb KR011062.1
Mushu2_2018	gi 827196798 gb KR011062.1
Mushu2_2018	gi 827196798 gb KR011062.1
Mushu2_2018	gi 827196798 gb KR011062.1

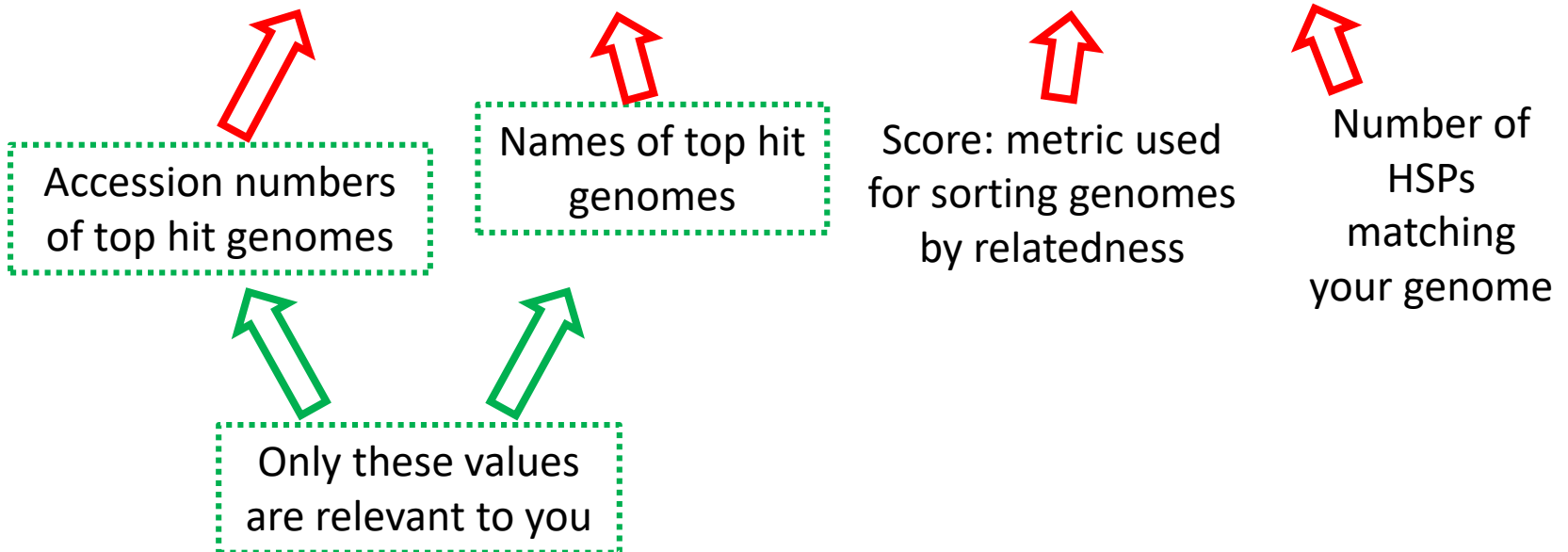
44: Sequence(s) from Apollo

1 sequences
format: **fasta**, database: ?

```
>Mushu2_2018
ATGCAAACGGCCCGCGTTTATCCCCGGCGATATCGCC
GGAAATCCTCGGCCCGGTTTCCCCGTCACGGTCGCAAC
GCCAGGGCCAGTGATCCGGCCAGCACGCCGCTCGGCG
GACCTCGGCGGGTAGCGGAAATCTCAGTAATCGCTACCO
```

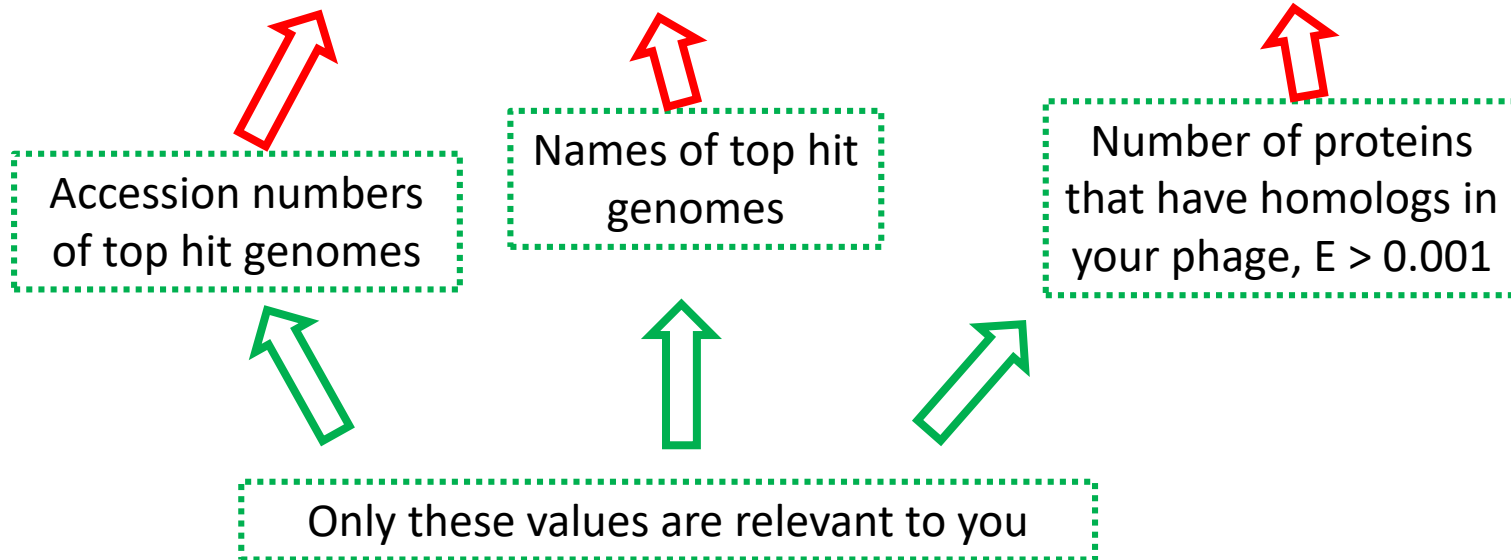
Top BLASTn hits

1	2	3	4
# ID	Name	Score	Nucleotide Hits
GU580940.1	Rhodococcus phage ReqiDocB7, complete genome	0.051	9
KR063279.1	Gordonia phage GMA3, complete genome	0.029	10
KX557281.1	Gordonia phage Jumbo, complete genome	0.028	9
KU998254.1	Gordonia phage Kampe, complete genome	0.024	6
KU998253.1	Gordonia phage Orchid, complete genome	0.024	6




Top BLASTp hits

1	2	3	4
# ID	Name	Score	Similar Unique Proteins
KU998252.1	Gordonia phage PatrickStar	12.066	34
KU998254.1	Gordonia phage Kampe	12.066	34
KU998253.1	Gordonia phage Orchid	12.066	34
KP790010.1	Gordonia phage GordDuk1	11.030	33
KX557281.1	Gordonia phage Jumbo	10.837	33



Percent nucleotide identity table

1	2	3	4	5	6	7
	Mushu2_2018	KX557281.1	KU998254.1	KU998253.1	KR063279.1	GU580940.1
Mushu2_2018	100.00	7.09	3.00	3.00	6.28	11.16
KX557281.1	7.09	100.00	5.11	5.11	23.20	4.87
KU998254.1	3.00	5.11	100.00	99.98	7.02	3.40
KU998253.1	3.00	5.11	99.98	100.00	7.01	3.40
KR063279.1	6.28	23.20	7.02	7.01	100.00	5.00
GU580940.1	11.16	4.87	3.40	3.40	5.00	100.00



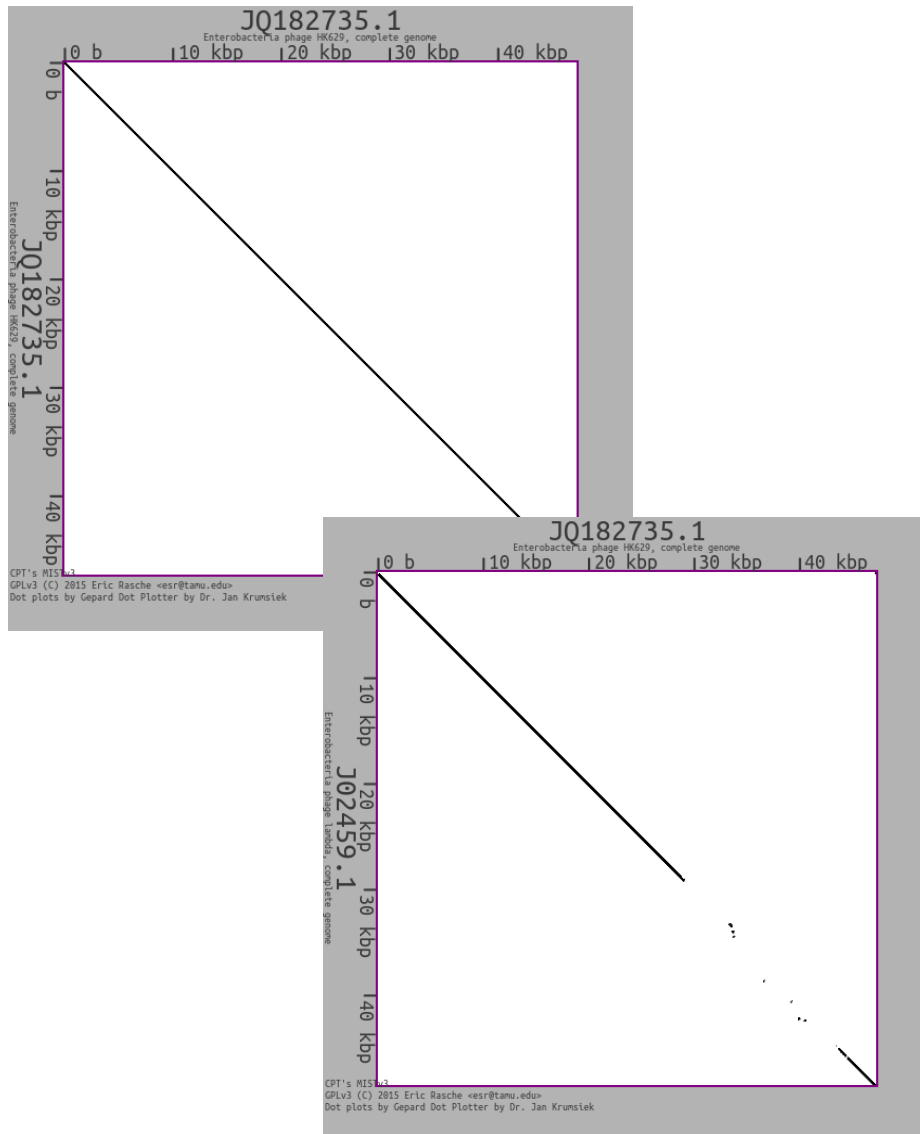
Accession numbers
of top hit genomes



Pairwise % nucleotide identity by the
Dice coefficient

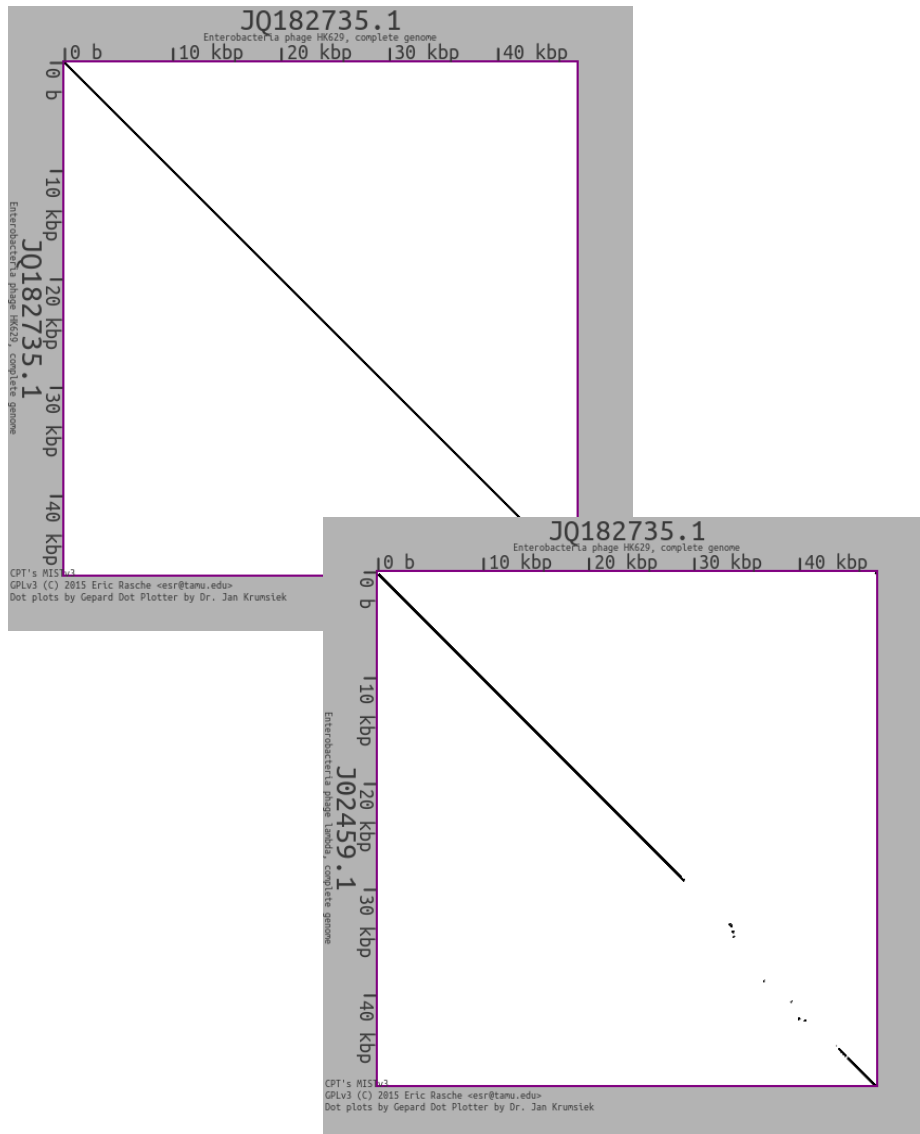
$$\text{Dice} = \frac{2 \times (\# \text{ identical bases})}{(\text{length of seq 1}) + (\text{length of seq 2})}$$

Dotplot comparisons



- A **dotplot** is a simple way to visualize the similarity between two sequences
- Lays out two sequences along X and Y axes and compares them in sliding “windows” and draws a dot if the 2 windows match above a threshold
- Can visualize gross similarity, **synteny** and major genome rearrangements

Dotplot comparisons



- Dotplots are “low resolution” comparisons
- Window sizes are usually >20 bp, so individual SNPs or indels are not visible
- As sequence similarity degrades, plot line becomes patchy or disappears

Galaxy training resources

- Galaxy home: <https://usegalaxy.org/>
- Galaxy 101: <https://galaxyproject.org/tutorials/g101/>
- CPT Galaxy training: <https://cpt.tamu.edu/training-material>

Welcome to CPT Galaxy Training

Collection of tutorials for CPT Galaxy users and BICH464 students. Further tutorials developed and maintained by the worldwide Galaxy community are available here.

CPT Galaxy for Students

Topic	Tutorials
Introduction to Galaxy and Apollo	6
Additional Analyses	5
Phage Annotation Pipeline in CPT Galaxy	4

CPT Galaxy for Scientists

Topic	Tutorials
De Novo Assembly	2