Workflow used for data generation in Chapter 4
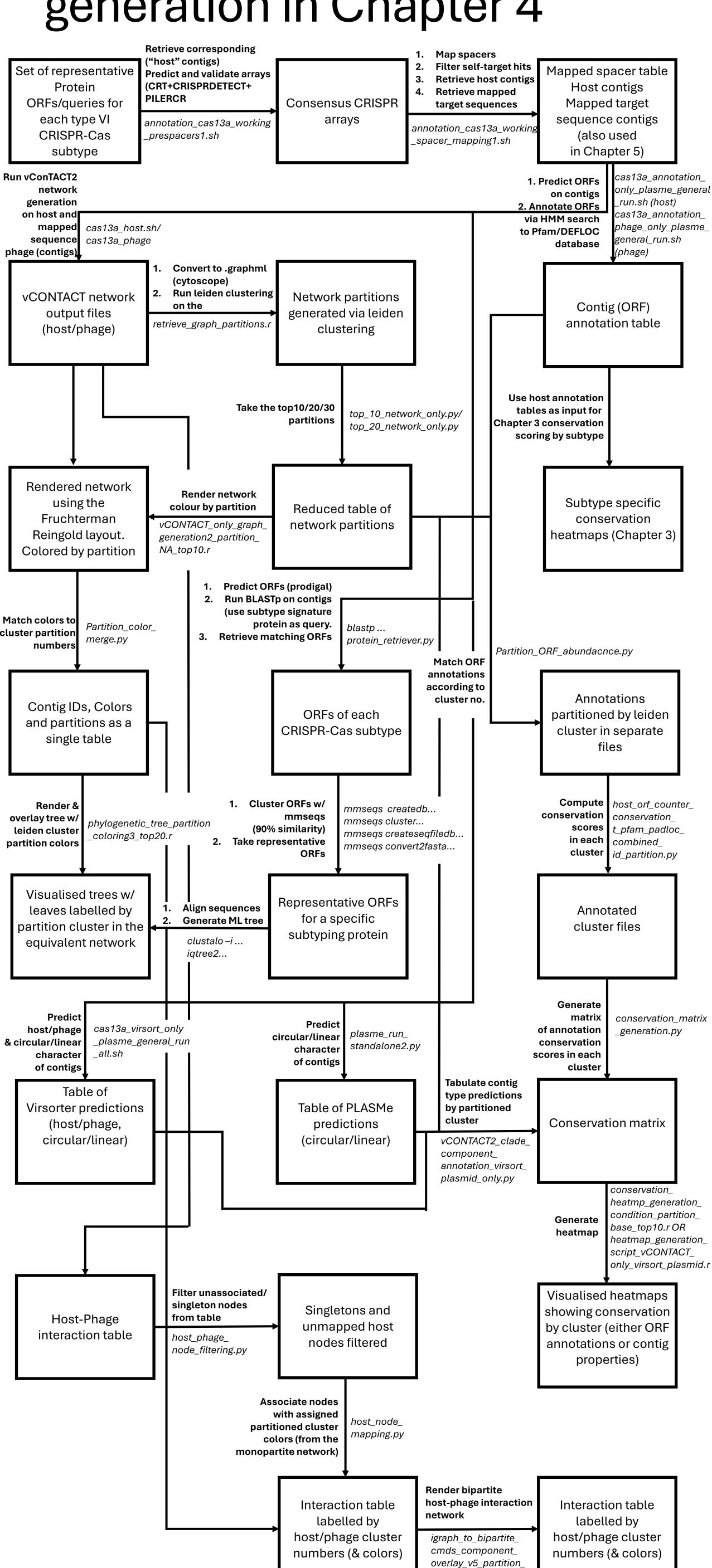
Set of representative Protein ORFs/queries for each type VI CRISPR-Cas subtype

→ **Retrieve corresponding ("host" contigs) Predict and validate arrays (CRT+CRISPRDETECT+ PILERCR)**
annotation_cas13a_working _prespacers1.sh

Consensus CRISPR arrays

→
1. **Map spacers**
2. **Filter self-target hits**
3. **Retrieve host contigs**
4. **Retrieve mapped target sequences**

annotation_cas13a_working _spacer_mapping1.sh

Mapped spacer table Host contigs Mapped target sequence contigs (also used in Chapter 5)

Run vConTACT2 network generation on host and mapped sequence phage (contigs)
cas13a_host.sh/ cas13a_phage

1. **Predict ORFs on contigs**
2. **Annotate ORFs via HMM search to Pfam/DEFLOC database**

cas13a_annotation_ only_plasme_general _run.sh (host) cas13a_annotation_ phage_only_plasme_ general_run.sh (phage)

vCONTACT network output files (host/phage)

→
1. **Convert to .graphml (cytoscope)**
2. **Run leiden clustering on the**
retrieve_graph_partitions.r

Network partitions generated via leiden clustering

Contig (ORF) annotation table

Take the top10/20/30 partitions
top_10_network_only.py/ top_20_network_only.py

Use host annotation tables as input for Chapter 3 conservation scoring by subtype

Rendered network using the Fruchterman Reingold layout. Colored by partition

← **Render network colour by partition**
vCONTACT_only_graph_ generation2_partition_ NA_top10.r

Reduced table of network partitions

Subtype specific conservation heatmaps (Chapter 3)

Match colors to cluster partition numbers
Partition_color_ merge.py

1. **Predict ORFs (prodigal)**
2. **Run BLASTp on contigs (use subtype signature protein as query.**
3. **Retrieve matching ORFs**
blastp ... protein_retriever.py

Match ORF annotations according to cluster no.

Partition_ORF_abundacnce.py

Contig IDs, Colors and partitions as a single table

ORFs of each CRISPR-Cas subtype

Annotations partitioned by leiden cluster in separate files

Render & overlay tree w/ leiden cluster partition colors
phylogenetic_tree_partition _coloring3_top20.r

1. **Cluster ORFs w/ mmseqs (90% similarity)**
2. **Take representative ORFs**
mmseqs createdb... mmseqs cluster... mmseqs createseqfiledb... mmseqs convert2fasta...

Compute conservation scores in each cluster
host_orf_counter_ conservation_ t_pfam_padloc_ combined_ id_partition.py

Visualised trees w/ leaves labelled by partition cluster in the equivalent network

←
1. **Align sequences**
2. **Generate ML tree**
clustalo –i ... iqtree2...

Representative ORFs for a specific subtyping protein

Annotated cluster files

Predict host/phage & circular/linear character of contigs
cas13a_virsort_only _plasme_general_run _all.sh

Predict circular/linear character of contigs
plasme_run_ standalone2.py

Generate matrix of annotation conservation scores in each cluster
conservation_matrix _generation.py

Table of Virsorter predictions (host/phage, circular/linear)

Table of PLASMe predictions (circular/linear)

Tabulate contig type predictions by partitioned cluster
vCONTACT2_clade_ component_ annotation_virsort_ plasmid_only.py

Conservation matrix

Generate heatmap
conservation_ heatmp_generation_ condition_partition_ base_top10.r OR heatmap_generation_ script_vCONTACT_ only_virsort_plasmid.r

Host-Phage interaction table

→ **Filter unassociated/ singleton nodes from table**
host_phage_ node_filtering.py

Singletons and unmapped host nodes filtered

Visualised heatmaps showing conservation by cluster (either ORF annotations or contig properties)

Associate nodes with assigned partitioned cluster colors (from the monopartite network)
host_node_ mapping.py

Interaction table labelled by host/phage cluster numbers (& colors)

→ **Render bipartite host-phage interaction network**
igraph_to_bipartite_ cmds_component_ overlay_v5_partition_ stars_working.R

Interaction table labelled by host/phage cluster numbers (& colors)