1) Create <u>required</u> VICSIN inputs (`input_file.txt, config_file.yml`):

Write file config_file.yml:

```
# DATA PATHS
input_path: /path/to/input_directory/
#input _path directory must contain all sequence files
#can be genbank, fasta, fasta + gff, or a combination
output_path: /path/to/output_directory/
#ouput_path to where output goes, cannot already exist
spacer_fasta_file: /path/to/spacers.fna
#optional input
#muliseq fasta format (each spacer is a seq)
#looks for matches to CRISPR spacers from your organism by BLAST
spine_core_file: /path/to/output.backbone.fasta
#advanced optional input
#use if you do not want to compute new core genome
known_viral_types: /path/to/known_mges.fasta
#optional input
#fasta file with DNA sequences of known MGEs infecting your organism
#looks for similar MGEs by BLAST
masking_file: /path/to/masks.txt
#optional input
#recommended if your genomes have CRISPR arrays
#prevents matching CRISPR spacers to host arrays
# PROGRAM PATHS
genbank_to_seed: /path/to/genbank_to_seed.py
phispy: /path/to/phiSpy.py
virsorter_data_dir: /path/to/VirSorter/virsorter-data
prodigal: /path/to/prodigal.linux
spine: /path/to/spine.pl
agent: /path/to/AGEnt.pl
# PROGRAM PARAMETERS
spine_agent_min_size_core: 500
```

Write file input_file.txt:

```
<genome_name_1>
<genome_name_2>
<genome_name_n>
```

This is just a list of names. These names must match input file names located in /path/to/input_directory/, but not necessarily the header lines in the FASTA.

EXAMPLE input_file.txt content:

```
Bacteroides_fragilis_CL03T12C07
Bacteroides_fragilis_NCTC_9343
Bacteroides_sp_3_2_5
Bacteroides_fragilis_638R
Bacteroides_fragilis_CL05T00C42
Bacteroides_fragilis_YCH46
```

Names of corresponding input files (located in /path/to/input_directory/):

```
Bacteroides_fragilis_3_1_12.fna
Bacteroides_fragilis_CL03T12C07.fna
Bacteroides_fragilis_NCTC_9343.fna
Bacteroides_sp_3_2_5.fna
Bacteroides_fragilis_3_1_12.gff
Bacteroides_fragilis_CL03T12C07.gff
Bacteroides_fragilis_NCTC_9343.gff
Bacteroides_sp_3_2_5.gff
Bacteroides_fragilis_638R.fna
Bacteroides_fragilis_CL05T00C42.fna
Bacteroides_fragilis_YCH46.fna
Bacteroides_fragilis_638R.gff
Bacteroides_fragilis_CL05T00C42.gff
Bacteroides_fragilis_YCH46.gff
```

FASTA or Genbank is required. GFF is optional if you have annotations, otherwise VICSIN will annotate ORFs for you with Prodigal.

***CHECK YOUR FASTA HEADER LINES. VICSIN is very sensitive to special characters. Underscores ("_") are ok. Pipes ("|") *will* break the program.

***BEWARE EXTRA LINES. VICSIN is very sensitive to empty lines in input_file.txt and config_file.yml.

5) Create optional VICSIN inputs (`masks.txt, spacers.fna, output.backbone.fasta, known_mges.fasta`):

Write file masks.txt:

```
<contig name>\t<start>\t<stop>
<contig name>\t<start>\t<stop>
<contig name>\t<start>\t<stop>
```

Contig name should match header line in FASTA file. Each region will be excluded from VICSIN prediction. Each region must be on a separate line. BEWARE EMPTY LINES.

EXAMPLE masks.txt:

```
NZ_JH636044      2843204    2843943
Bacteroides_fragilis_NCTC_9343   2998099    2998517
Bacteroides_fragilis_NCTC_9343   4661217    4663160
Bacteroides_fragilis_638R  4711809    4713911
Bacteroides_fragilis_YCH46 2924517    2924932
```

Write file (s) spacers.fna/known_mges.fasta. Multisequence nucleotide fasta file with each spacer or MGE as an individual sequence.

output.backbone.fasta should be an output from Spine, or written to mimic spine output format.

6) Run VICSIN

```
$ vicsin <input_file.txt> <config_file.yml>
```

Use best assembled genomes possible. VICSIN works best with complete genomes, but also works well with genomes in up to 10 contigs. It will run with more fragmented genomes, but will miss predictions.

Recommended for 5-10 genomes. More genomes takes longer. Runs with 7-8 genomes will take 4-5 hours (using these run conditions on the IGB biocluster).

7) Interpret output

The contents of the output_directory should look like **this**:

```
Agent_Runs #all of the AGEnt output files: accessory genome regions
Converted_Input_Files #processed/reformatted input files
Output_Files     #THIS IS THE MAIN OUTPUT
Pre_Reblast_Output_Files    #ReBLAST intermediate files
Spine_Runs #all of the Spine output files: core genome regions
Virsorter_Runs   #all of the Virsorter output files: MGE predictions
(*_global-phage-signal.csv) and gene annotations (*_mga_final.predict)
Cluster_Output   #networking file to show relatedness between MGE
predictions; out.tbl can be read directly into Cytoscape
CRISPR_Runs      #outputs and intermediates from CRISPR BLAST
(*_CRISPR.aln)
PhiSpy_Runs      #all of the PhiSpy output files: MGE predictions
(prophage.tbl)
ReBlast_Runs     #final ReBLAST processed extensions of MGE predictions
VICSIN-20180216-1349.txt
```

***Output_Files
VICSIN predictions get assigned to a "Type" which reflects our confidence in the prediction. Type 1 predictions are most confident. In general, predictions from VirSorter or with two or more methods of support should be kept. Be wary of predictions supported by AGEnt and PhiSpy, but no other tools. Predictions supported by AGEnt and CRISPR BLAST should be examined to determine if they are MGEs, or host CRISPR spacer arrays.

prediction name = <Genome name>-<Contig name>-<number>
Sequence = Contig name
methods = A (Agent), V (Virsorter), P (Phispy), R (Reblast), C (Crispr), B (BLAST to known MGE)

EXAMPLE output file (you will have many more predictions than this):

```
# Type 1: Predicted by >2 methods
# Prediction     Sequence    methods     start end
Bacteroides_fragilis_638R-Bacteroides_fragilis_638R-97
      Bacteroides_fragilis_638R   A,V,P 4293636     4329074
# Type 2: Predicted by 2 methods
# Prediction     Sequence    methods     start end
Bacteroides_fragilis_638R-Bacteroides_fragilis_638R-11
      Bacteroides_fragilis_638R   A,P   442623      457213
```

```
Bacteroides_fragilis_638R-Bacteroides_fragilis_638R-13
     Bacteroides_fragilis_638R  A,P  488208     501792
# Type 3: Predicted by 1 1° method
# Prediction    Sequence   methods    start end
# Type 4: Predicted by 1 2° method
# Prediction    Sequence   methods    start end
Bacteroides_fragilis_638R-Bacteroides_fragilis_638R-0
     Bacteroides_fragilis_638R  A    591   14052
Bacteroides_fragilis_638R-Bacteroides_fragilis_638R-1
     Bacteroides_fragilis_638R  A,R  46507 69771
```