Firstly, we need to predict proteins from our metagenomic assembly in .*fasta* format. To do this, we run *Prodigal* in normal mode using this command:

**prodigal -i <filename.fa> -a protein.translations.faa**

The terminal output will look like this:

Finding genes in sequence #24051 (66 bp)...done!

DEFINITION seqnum=24051;seqlen=66;seqhdr="sc14sc14sc14NODE\_24051\_length\_66\_cov\_0.090909";version=Prodigal.v2.60;run\_type=Single;model="Ab initio";gc\_cont=43.06;transl\_table=11;uses\_sd=0

FEATURES Location/Qualifiers

CDS complement(<1..>66)

/note="ID=24051\_1;partial=11;start\_type=Edge;rbs\_motif=None;rbs\_spacer=None;gc\_cont=0.409;conf=70.60;score=3.81;cscore=2.20;sscore=1.61;rscore=0.00;uscore=0.00;tscore=1.61;"

//

But all we need to perform further analysis (i.e. predicted proteins) will be contained in file *protein.translations.faa*:

>sc14sc14sc14NODE\_1\_length\_61980\_cov\_12.082681\_1 # 1 # 906 # -1 # ID=1\_1;partial=10;start\_type=ATG;rbs\_motif=TAA;rbs\_spacer=12bp;gc\_cont=0.353

MYYRCELLINGLKYRVTDDLENWDEVKASFKRNDYDGVIRTFSNKFSFAGDARKLLLKQY

DEDYLNASASIIISTRNNSWLYNERFSCALNFSTLQDNGRILQINAVDDSVASMIKSKKG

TQYEYSVEEVKRPIPLVYDGLELSESAKWIPTGDTLEDDDTLINVYFSKKMSPMPIYITA

SDSLIKGSLEFNDQTVGGDDVYSIKALKSIRINIEFNIDMFVFREYQSGALGYDVRGVRL

QIMKISNEIDSNGEAVIGSFELTTESETPVEKKVSESYNISLLHNDKIIVRAMYVNEKEE

IV

Secondly, we need to compare the predicted proteins with those associated with capsid, terminase and portal genes. Programme hmmsearch is used to search one or more profiles against a sequence database. Here is an example of comparing them with capsid gene proteins using *hmmsearch*:

**hmmsearch --tblout target.out All\_pVOG\_capsids.hmm protein.translations.faa**

The terminal output will look like:

Internal pipeline statistics summary:

-------------------------------------

Query model(s): 1 (117 nodes)

Target sequences: 32305 (3513431 residues searched)

Passed MSV filter: 1102 (0.0341124); expected 646.1 (0.02)

Passed bias filter: 469 (0.0145179); expected 646.1 (0.02)

Passed Vit filter: 30 (0.000928649); expected 32.3 (0.001)

Passed Fwd filter: 1 (3.0955e-05); expected 0.3 (1e-05)

Initial search space (Z): 32305 [actual number of targets]

Domain search space (domZ): 0 [number of targets reported over threshold]

# CPU time: 0.24u 0.02s 00:00:00.26 Elapsed: 00:00:00.08

# Mc/sec: 5138.39

//

Query: VOG10456 [M=457]

Scores for complete sequences (score includes all domains):

--- full sequence --- --- best 1 domain --- -#dom-

E-value score bias E-value score bias exp N Sequence Description

------- ------ ----- ------- ------ ----- ---- -- -------- -----------

------ inclusion threshold ------

0.17 11.8 0.0 0.19 11.6 0.0 1.1 1 sc14sc14sc14NODE\_4506\_length\_649\_cov\_2.099327\_2 # 270 # 647 # -1 # ID=4506\_2;par

Domain annotation for each sequence (and alignments):

>> sc14sc14sc14NODE\_4506\_length\_649\_cov\_2.099327\_2 # 270 # 647 # -1 # ID=4506\_2;partial=01;start\_type=Edge;rbs\_motif=No

# score bias c-Evalue i-Evalue hmmfrom hmm to alifrom ali to envfrom env to acc

--- ------ ----- --------- --------- ------- ------- ------- ------- ------- ------- ----

1 ? 11.6 0.0 5.9e-06 0.19 359 399 .. 56 98 .. 35 115 .. 0.81

Alignments for each domain:

== domain 1 score: 11.6 bits; conditional E-value: 5.9e-06

VOG10456 359 ngvqtnpgddvgiitqsykti..pifesddvvqdtlsrvylvd 399

v++npg dvg++t + + + i +++ qd++ r+y +

sc14sc14sc14NODE\_4506\_length\_649\_cov\_2.099327\_2 56 VAVEANPGHDVGVVTLTGRLVklQIKKANLKSQDDIKRIYRLA 98

5799\*\*\*\*\*\*\*\*\*\*988775511688899999\*\*\*\*\*\*\*\*754 PP

Our main interest is focused on *target.out* - it is a simple tabular (space-delimited) file summarizing the per-target output, with one data line per homologous target sequence found:

# --- full sequence ---- --- best 1 domain ---- --- domain number estimation ----

# target name accession query name accession E-value score bias E-value score bias exp reg clu ov env dom rep inc description of target

#------------------- ---------- -------------------- ---------- --------- ------ ----- --------- ------ ----- --- --- --- --- --- --- --- --- ---------------------

sc14sc14sc14NODE\_15520\_length\_348\_cov\_1.259386\_1 - VOG2384 - 1.1e-07 31.7 0.1 1.2e-07 31.7 0.1 1.0 1 0 0 1 1 1 1 # 45 # 347 # 1 # ID=15520\_1;partial=01;start\_type=ATG;rbs\_motif=None;rbs\_spacer=None;gc\_cont=0.597

In our case, in this file we have a brief information about contigs containing at least one capsid protein. You may also search for contigs containing terminase and portal gene proteins simply replacing the hidden Markov model for capsids (*All\_pVOG\_capsids.hmm*) by hidden Markov models for terminase (*All\_terminase.hmm*) and portal genes (*All\_portal\_hmms.hmm*).

Finally, using *target.out* you may calculate statistics needed via Python3 script *find\_contigs.py*.

The general approach is to run commands of this kind:

**$ python -c ‘import find\_contigs; print find\_contigs.<function\_name>(<filename(s)>)’**

For example, if you want to see a list of contigs containing capsid genes, run this:

**$ python -c ‘import find\_contigs; print find\_contigs.get\_capsid(target.out)’**

Here you should use *target.out* extracted from *hmmsearch* for hidden Markov models of capsid proteins.

Output example:

['sc14sc14sc14NODE\_10292\_length\_422\_cov\_1.787466'

'sc14sc14sc14NODE\_10429\_length\_419\_cov\_1.846154'

'sc14sc14sc14NODE\_10440\_length\_419\_cov\_1.109890']

If you want to see the number of contigs containing capsid genes, run this:

**$ python -c ‘import find\_contigs; print find\_contigs.get\_capsid\_number(target.out)’**

Here you should use *target.out* extracted from *hmmsearch* for hidden Markov models of capsid proteins.

Output example:

3

If you want to see a list of contigs containing only terminase genes, run this:

**$ python -c ‘import find\_contigs; print find\_contigs.terminase\_only(portal\_target.out, terminase\_target.out, capsid\_target.out)’**

Here you should use *portal\_target.out* extracted from *hmmsearch* for hidden Markov models of portal proteins, *terminase\_target.out* extracted from *hmmsearch* for hidden Markov models of terminase proteins, *capsid\_target.out* extracted from *hmmsearch* for hidden Markov models of capsid proteins.

If you want to see the number of contigs containing capsid and portal genes only, run this:

**$ python -c ‘import find\_contigs; print find\_contigs.portal\_and\_capsid\_only\_number(portal\_target.out, terminase\_target.out, capsid\_target.out)’**

Here you should use *portal\_target.out* extracted from *hmmsearch* for hidden Markov models of portal proteins, *terminase\_target.out* extracted from *hmmsearch* for hidden Markov models of terminase proteins, *capsid\_target.out* extracted from *hmmsearch* for hidden Markov models of capsid proteins.

If you want to see the average number of contigs containing capsid, terminase or portal genes, run this:

**$ python -c ‘import find\_contigs; print find\_contigs.average\_capsid\_terminase\_portal\_number(portal\_target.out, terminase\_target.out, capsid\_target.out)’**

Here you should use *portal\_target.out* extracted from *hmmsearch* for hidden Markov models of portal proteins, *terminase\_target.out* extracted from *hmmsearch* for hidden Markov models of terminase proteins, *capsid\_target.out* extracted from *hmmsearch* for hidden Markov models of capsid proteins.

For more information see *find\_contigs.py*.