

Nov 01, 2017

Version 1

Gene calling with Prodigal V.1



Forked from [MG_HW6: Gene Calls with Prodigal](#)

DOI

dx.doi.org/10.17504/protocols.io.kixcufn

James E Thornton Jr¹

¹Hurwitz Lab

Metafunc Course 2017



James E Thornton Jr

Hurwitz Lab



Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.kixcufn>

Protocol Citation: James E Thornton Jr 2017. Gene calling with Prodigal. **protocols.io**

<https://dx.doi.org/10.17504/protocols.io.kixcufn>



License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

Created: November 01, 2017

Last Modified: March 27, 2018

Protocol Integer ID: 8503

Keywords: gene call, prodigal this protocol, using prodigal, procedure, prodigal, protocol, contig

Abstract

This protocol provides the procedure to generate gene calls on your contigs using Prodigal.

Troubleshooting



- 1 Login to the HPC and move into ocelote.

Command

```
$ ssh hpc  
$ ocelote
```

- 2 Move into your class directory.

Command

Use YOUR username to be in the right directory

```
$ cd /rsgrips/bh_class/username
```

- 3 Copy the following into a new script called run-interactive.sh :

**Command****Replace netid**

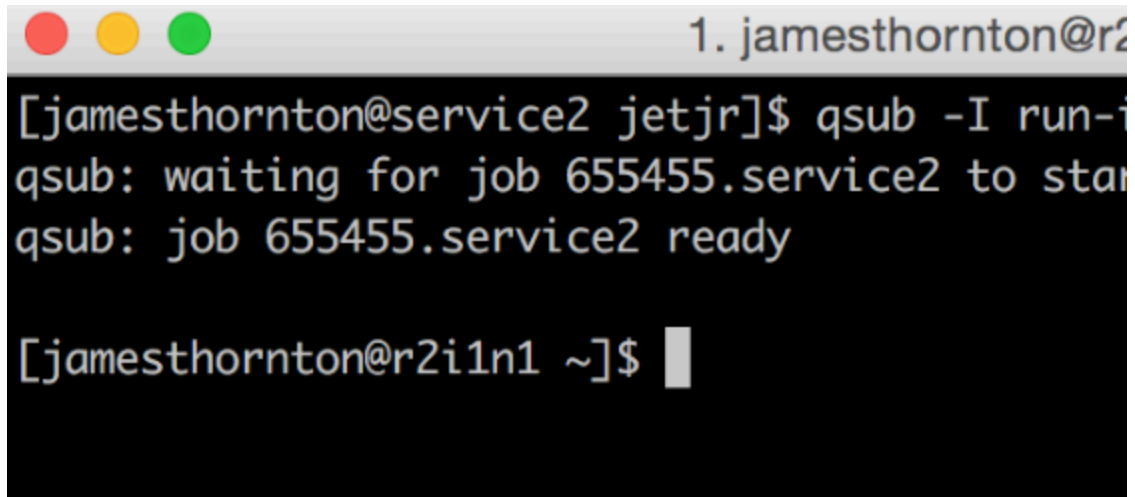
```
#!/bin/bash
#PBS -W group_list=bh_class
#PBS -q windfall
#PBS -l select=1:ncpus=2:mem=4gb
#PBS -l walltime=24:00:00
#PBS -l cput=24:00:00
#PBS -M netid@email.arizona.edu
#PBS -m bea
```

- 4 Submit run-interactive.sh interactively using qsub:

Command**The capital**

```
qsub -I run-interactive.sh
```

Expected result



```
1. jamesthornton@r2i1n1 ~$  
[jamesthornton@service2 jetjr]$ qsub -I run-  
qsub: waiting for job 655455.service2 to start  
qsub: job 655455.service2 ready  
[jamesthornton@r2i1n1 ~]$
```

- 5 Once the job is ready move back into your class directory.

Command

Use YOUR username

```
$ cd /rsgprs/bh_class/username
```

- 6 Make a directory named prodigal:

Command

```
$ mkdir prodigal
```

- 7 Move into your assembly directory which contains your contigs:



Note

Note: Its possible your contigs are in another directory (megahit-out). Move to the directory where final_contigs.fa are located. Remember final_contigs.fa is the combined assemblies from your partner.

Command

```
$ cd /rsgprs/bh_class/username/assembly
```

- 8 Load prodigal and run it on your final_contigs.fa to generate gene calls.

Note

Since final_contigs.fa may be located somewhere other than the assembly directory, you can write out the full paths to make sure the output goes where its suppose to:

```
$ prodigal -i /path/to/final_contigs.fa -o /rsgprs/bh_class/username/prodigal/gene_calls -  
a /rsgprs/bh_class/username/prodigal/proteins.faa -d  
/rsgprs/bh_class/username/prodigal/nucleotides.fna
```

Note

The prodigal command is executed in one line.

Command

IMPORTANT: make sure to run prodigal on the final_contigs.fa file which was generated when combining assemblies with your partner.

The output will be placed in the prodigal directory that was created and the file name will be gene_calls in addition to a file containing the protein (proteins.faa) and nucleotide (nucleotide.faa) sequences for the genes.

```
$ prodigal -i final_contigs.fa -o ../prodigal/gene_calls -a  
../prodigal/proteins.faa -d ../prodigal/nucleotides.fna
```

- 9 Move into your prodigal directory and make sure the gene calls were generated.

Command

```
$ cd /rsgtps/bh_class/username/prodigal  
$ head gene_calls
```

Expected result



A terminal window titled "1. jamesthornton@r2i1n1:/rsgrps/bh..." displays the output of the command "head gene_cat". The output shows the first few lines of a Prodigal-generated file, including a "DEFINITION" line with sequence information and a "FEATURES" section listing CDS (Coding DNA Sequences) with their locations and various annotations like GC content, confidence, and scores.

```
[jamesthornton@r2i1n1 prodigal]$ head gene_cat
DEFINITION  seqnum=1;seqlen=3384;seqhdr="1";ve
ngle;model="Ab initio";gc_cont=59.13;transl_to
FEATURES
    CDS
        <1..585
            /note="ID=1_1;partial=10;
s_spacer=None;gc_cont=0.694;conf=100.00;score=
core=0.00;uscore=0.00;tscore=3.22;"
    CDS
        complement(611..1393)
            /note="ID=1_2;partial=00;
AGG;rbs_spacer=5-10bp;gc_cont=0.674;conf=100.0
re=5.32;rscore=0.27;uscore=0.84;tscore=3.13;"
    CDS
        complement(1412..1933)
            /note="ID=1_3;partial=00;
G;rbs_spacer=5-10bp;gc_cont=0.600;conf=100.00;
.29;rscore=2.87;uscore=-0.31;tscore=3.13;"
    CDS
        complement(1930..>3384)
            /note="ID=1_4;partial=01;
s_spacer=None;gc_cont=0.635;conf=99.99;score=2
score=0.00;uscore=0.00;tscore=3.22;"
[jamesthornton@r2i1n1 prodigal]$
```

- 10 Use your scripting or unix skills to detect how many genes were detected in your contigs file. Create a table that shows the number of 'ORFs' or open reading frames detected on the contigs.