Lab 9: Sequence Assembly

This report and the data used for it can be found at https://github.com/abondrn/compbio-labs/tree/master/assembly.

Using SPAdes, you will assemble a bacterial genome de novousing a combination of long PacBio reads and short Illumina reads. Next week, you will analyze your genome to determine the species and obtain an overview of its metabolism. You are expected to keep a thorough record of everything you did in your notebook.

References

- Sequence assembly
- SPAdes
- PacBio SMRT Sequencing
- Illumina sequencing

Background

Genome sequencing and assembly arecommon techniques in biology. To obtain the sequence of a long genome, DNA must be chopped into small pieces that can be read by a sequencer. These short reads must then be stitched back together to form acomplete genome.Often, the genome cannot be fully assembled because there are multiple equally plausible ways of stitching the reads together. Ideally, each chromosome is assembled into a single, long sequence. In practice, chromosomes are often assembled into multiple “contigs,” or contiguous sequences. A genome assembly is generally considered complete only when all (or nearly all) the sequences are accounted for. Otherwise, it is considered a draft genome.

In this lab, you will assemble and analysis a bacterial genome, using Illumina and PacBio reads. This week, you will take the reads and assemble them into a complete genome. Next week, you will analyze the contents of your genome.

Locating the data

DNA from an unknown bacterium was sequenced using PacBio and Illumina technologies. The resulting reads are uploaded onto bCourses. You will need to download these files. You will need to upload the files if you are using DataHub.

`illumina_reads_R1.fastq` – first paired-end read \ `illumina_reads_R2.fastq` – second paired-end read \ `pacbio_reads.fastq` – long PacBio reads

In [7]:
```
!bzip2 -dk reads/*.fastq.bz2
```

```
bzip2: Output file reads/illumina_reads_R1.fastq already exists.
bzip2: Output file reads/illumina_reads_R2.fastq already exists.
```

Running SPAdes

SPAdes is a hybrid genome assembler, meaning that it takes multiple sources of information as input and combines them to produce an optimal assembly. Assemblies using only short reads tend to be highly fragmented (i.e., many contigs). Assemblies using a high-quality short read set and a higher error rate long-read set (like PacBio) tend to be the best.

Why do we expect short reads to produce a more fragmented assembly than long reads?

> We would expect short reads to result in more fragmented assemblies because the shortness of the input leads to a higher amount of contigs versus long reads, which would increase the likelihood of there being gaps between reads, thus splitting up the assembly.

Why does a single-molecule sequencing like PacBio have a higher error rate than Illumina?

> Illuma uses many small reads versus the individual long strands used in SMRT. Thus, in the case of inevitable misreads, Illumina can use multiple overlaping contigs for the same reading frame thus having a greater coverage than the equivalent PacBio sequencing and reducing error.

We need to come up with a SPAdes command. At a minimum, you will need to specify the output directory with -o, the path to the first Illumina read with -1, the path to the second Illumina read with -2, and the path to your PacBio reads with --pacbio. Note: SPAdes must be run from the command line, and can take a while.

Genome assembly requires a relatively large amount of computer memory. Sometimes up to 1TB. Datahub instances have 64GB of memory. We have significantly subsampled the reads in order to run the analysis on DataHub.

SPAdes typically uses multi-threading to speed up assembly. Each thread requires memory. You may need to add -t 4 to your command so that it uses only 4 threads on the system rather than 16. If the program crashes, reduce the number of threads.

```
```

=== KDE started.

```
```

=== KDE finished.

```
```

=== KDE finished.

```
```

Lab 10

You will analyze the bacterial genome you assembled in the previous lab. You will summarizethe quality of your assembly using a few different statistics, identify the genome's taxonomic origin, then obtain two genome annotations via different pipelines. You are expected to keep a thorough record of everything you did in your notebook.

Assembly statistics

- https://en.wikipedia.org/wiki/N50,_L50,_and_related_statistics

rRNA Identification

- https://en.wikipedia.org/wiki/16S_ribosomal_RNA
- http://hmmer.org/
- Barrnap
- SeqMatch

Bedtools

Annotation services

- https://narrative.kbase.us/#catalog/modules/ProkkaAnnotation
- http://rast.nmpdr.org/

In [3]:
```
from Bio import SeqIO
import matplotlib.pyplot as plt
```

In [ ]:
```
!conda install -c bioconda assembly-stats
!conda install -c bioconda -c conda-forge barrnap
!conda install -c bioconda bedtools
!conda install -c bioconda assembly-stats
!conda install -c bioconda hmmer
```

Generate assembly statistics

The genome assembled with the reduced set of reads may not be sufficient to receive a well annotated genome. A genome assembled with the full set of reads has been uploaded to bCourses. Download the compressed file which contains the contigs and scaffolds from bCourses.

Locate both assembled genomes. There should be a `contigs.fasta` and a `scaffolds.fasta` file in your SPAdes output directory as well. Contigs are contiguous sequences that could be assembled from your reads. Scaffolds are sets of contigs that have been stitched together in order, and are generally longer than contigs. Sometimes, the assembler can't tell what sequence connects two contigs in a scaffold, and inserts N's in the gap between them. Other times, the assembler has no additional information that could be used to determine the order and orientation of contigs in a scaffold. In this case, scaffolds == contigs.

Using the `assembly-stats` program, please calculate statistics on both your contigs and scaffolds files and the contigs and scaffolds files on bCourses. There will be 4 sets of statistics total. Report the total length of all contigs/scaffolds, the number of contigs/scaffolds, and the N50 for both assembled genomes in your iPython notebook.

In [3]:
```
!assembly-stats SPAdesoutput/contigs.fasta SPAdesoutput/scaffolds.fasta
```

```
stats for SPAdesoutput/contigs.fasta
sum = 5460976, n = 911, ave = 5994.49, largest = 73853
N50 = 17382, n = 92
N60 = 13875, n = 127
N70 = 11046, n = 171
N80 = 7865, n = 228
N90 = 4047, n = 323
N100 = 128, n = 911
N_count = 0
Gaps = 0
--------------------------------------------------------------------------------
stats for SPAdesoutput/scaffolds.fasta
sum = 5460996, n = 909, ave = 6007.70, largest = 73853
N50 = 17382, n = 92
N60 = 13875, n = 127
N70 = 11046, n = 171
N80 = 7865, n = 228
N90 = 4075, n = 323
N100 = 128, n = 909
N_count = 20
Gaps = 2
```

In [ ]:
```
!tar -xzvf contigs_and_scaffolds.tar.gz
```

In [4]:
```
!assembly-stats contigs.fasta scaffolds.fasta
```

```
stats for contigs.fasta
sum = 6676965, n = 170, ave = 39276.26, largest = 687575
N50 = 213896, n = 10
N60 = 201611, n = 13
N70 = 154231, n = 17
N80 = 124085, n = 22
N90 = 83516, n = 28
N100 = 128, n = 170
N_count = 0
Gaps = 0
--------------------------------------------------------------------------------
stats for scaffolds.fasta
sum = 6676985, n = 168, ave = 39743.96, largest = 687575
N50 = 213896, n = 10
N60 = 201611, n = 13
N70 = 154231, n = 17
N80 = 124085, n = 22
N90 = 92095, n = 28
N100 = 128, n = 168
N_count = 20
Gaps = 2
```

Questions:

Why is N50 a useful statistic to calculate, and what does it indicate? Why not just list the mean or median contig length?

> N50 is the length of the shortest contig, where combined with contigs of an equal or greater length, are 50% of the genome length. It is useful to calculate since it is a measure of quality/fragmentation. N50 is similar to a median in that it measures the half-mass point of the distribution, but whereas the mean/median only take into account individual measurements, N50 is much better at describing a distribution which can vary by several orders of magnitude by taking into account the total length and partial sums, which biases it towards higher statistics than a median.

In your assembly, are scaffolds longer than contigs, or are scaffolds approximately equal to contigs?

> In our assembly, the scaffolds are approximately the same length as contigs, however you would normally expect them to be longer.

What is the difference between the genomes assembled with the full set of reads and the down sampled set of reads?

> While the sum and various N statistics are more or less the same order of magnitude, the max and average lengths were much higher the in the bCourses contigs.

The number of times a contig in your assembly was covered by the reads used to assemble it (the "coverage") is listed at the end of the contig name in contigs.fasta. Extract the coverage from each FASTA header (using your preferred method) and plot a histogram of the coverage for all contigs for both assemblies.

In [12]:
```
next(SeqIO.parse("contigs.fasta", "fasta")).name
```

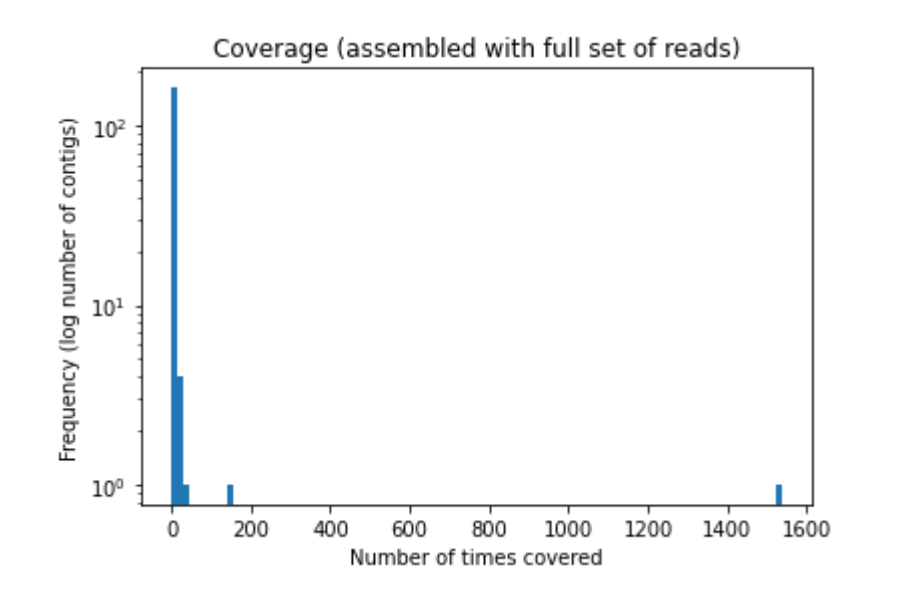
Out[12]: 'NODE_1_length_687575_cov_8.340472'

In [15]:
```
import numpy as np

coverage = []
for config in SeqIO.parse("contigs.fasta", "fasta"):
    coverage.append(float(config.name.rsplit('_', maxsplit=1)[1]))
print(len(coverage), np.mean(coverage), np.max(coverage))

plt.title("Coverage (assembled with full set of reads)")
plt.ylabel("Frequency (log number of contigs)")
plt.xlabel("Number of times covered")
plt.hist(coverage, bins=100, log=True)
plt.show()
```

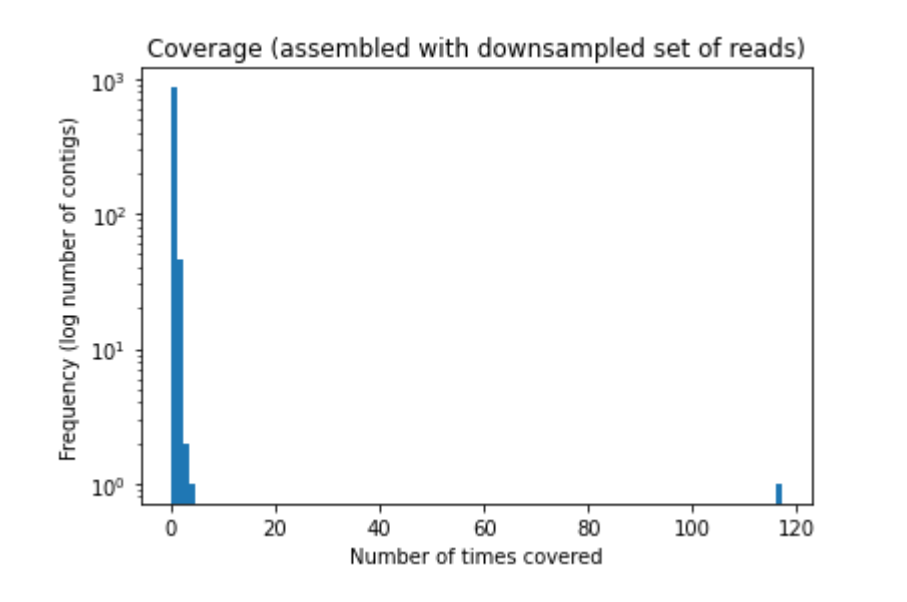
```
170 14.3863769 1536.0
```



In [16]:
```
coverage = []
for config in SeqIO.parse("SPAdesoutput/contigs.fasta", "fasta"):
    coverage.append(float(config.name.rsplit('_', maxsplit=1)[1]))
print(len(coverage), np.mean(coverage), np.max(coverage))

plt.title("Coverage (assembled with downsampled set of reads)")
plt.ylabel("Frequency (log number of contigs)")
plt.xlabel("Number of times covered")
plt.hist(coverage, bins=100, log=True)
plt.show()
```

```
911 0.860009048298573 117.0
```



Questions: **Is coverage uniformly distributed? Does it look Gaussian? Is it bimodal or trimodal?**

> The coverage is clearly not uniformly distributed or Gaussian due to its strong right skew, and due to the strong dropoff (and peaks). The coverage appears to follow a bimodal distribution, visible at a 100 or below vs above (however the max varies).

What explains the presence of contigs with coverage an integer multiple of the mean coverage?

> The mean of the coverage distribution is close to 14, though there are some contigs with coverages of 1536 and 150, which appear to be multiples of the average. This could be explained by the fact that the contigs that are covered more times by the reads from the sequencing have much smaller lengths.

Is there a difference in the histograms between assemblies? If so, what is the difference?

> While the shape is more or less the same, the actual number of contigs in the assembly I ran is much more than that in the full version! So, while they both share an outlier of length ~100, the full assembly resulted in an extreme outlier at around 1500.

Identify the taxonfrom which your genome originated

We know that the genome originated from a taxonof bacteria. One component of bacterial ribosomes is the 16S ribosomal RNA subunit.This functional RNA is conserved throughout all bacteria, and is often used as a taxonomic marker gene.Much of the gene is highly conserved, as function ribosomes are required for protein synthesis, but some regions differ greater between bacterial taxa. These "hypervariable regions" can be used to determine the taxonfrom which a 16S rRNA gene originated.

Note from this point, you are focusing on your assembled genome. To identify your genome, you must 1) scan over the entire genome to locate copies of the 16S ribosomal RNA gene, 2) extract the 16S rRNA genes from your assembly, and 3) compare these genes to a database of known 16S rRNA genes.

First, run the barrnap program on your assembled contigs to locate rRNA genes. `barrnap contigs.fasta > rrna.gff` The first argument is the path to the assembled contigs. The greater-than symbol redirects the output to the General Feature Format (GFF) file rrna.gff. This program uses a Hidden Markov Model (HMM) that describes the structure of ribosomal RNAgenes to find their coordinates inside a genome. After this program completes, delete all lines within rrna.gff than those containing 16S_rRNA genes.

In [6]:
```
!barrnap contigs.fasta > rrna.gff
```

```
[barrnap] This is barrnap 0.9
[barrnap] Written by Torsten Seemann
[barrnap] Obtained from https://github.com/tseemann/barrnap
[barrnap] Detected operating system: linux
[barrnap] Adding /opt/conda/lib/barrnap/bin/../binaries/linux to end of PATH
[barrnap] Checking for dependencies:
[barrnap] Found nhmmer - /opt/conda/bin/nhmmer
[barrnap] Found bedtools - /opt/conda/bin/bedtools
[barrnap] Will use 1 threads
[barrnap] Setting evalue cutoff to 1e-06
[barrnap] Will tag genes < 0.8 of expected length.
[barrnap] Will reject genes < 0.25 of expected length.
[barrnap] Using database: /opt/conda/lib/barrnap/bin/../db/bac.hmm
[barrnap] Scanning contigs.fasta for bac rRNA genes... please wait
[barrnap] Command: nhmmer --cpu 1 -E 1e-06 --w_length 3878 -o /dev/null --tblout /dev/stdout '/opt/
conda/lib/barrnap/bin/../db/bac.hmm' 'contigs.fasta'
[barrnap] Found: 16S_rRNA NODE_17_length_154231_cov_9.579771 L=1532/1585 124549..126080 + 16S ribos
omal RNA
[barrnap] Found: 16S_rRNA NODE_1_length_687575_cov_8.340472 L=1532/1585 46293..47824 + 16S ribosoma
l RNA
[barrnap] Found: 16S_rRNA NODE_23_length_103367_cov_10.080628 L=1532/1585 35538..37069 + 16S riboso
mal RNA
[barrnap] Found: 16S_rRNA NODE_3_length_378948_cov_10.008563 L=1532/1585 348507..350038 + 16S ribos
omal RNA
[barrnap] Found: 16S_rRNA NODE_5_length_332503_cov_9.112282 L=1532/1585 137018..138549 - 16S riboso
mal RNA
[barrnap] Found: 16S_rRNA NODE_6_length_298797_cov_8.927468 L=1532/1585 13936..15467 + 16S ribosoma
l RNA
[barrnap] Found: 23S_rRNA NODE_17_length_154231_cov_9.579771 L=2889/3232 126598..129486 + 23S ribos
omal RNA
[barrnap] Found: 23S_rRNA NODE_1_length_687575_cov_8.340472 L=2889/3232 48342..51230 + 23S ribosoma
l RNA
[barrnap] Found: 23S_rRNA NODE_23_length_103367_cov_10.080628 L=2889/3232 37587..40475 + 23S riboso
mal RNA
[barrnap] Found: 23S_rRNA NODE_3_length_378948_cov_10.008563 L=2889/3232 350556..353444 + 23S ribos
omal RNA
[barrnap] Found: 23S_rRNA NODE_5_length_332503_cov_9.112282 L=2889/3232 133612..136500 - 23S riboso
mal RNA
[barrnap] Found: 23S_rRNA NODE_6_length_298797_cov_8.927468 L=2889/3232 15985..18873 + 23S ribosoma
l RNA
[barrnap] Rejecting short 115 nt predicted 23S_rRNA. Adjust via --reject option.
[barrnap] Rejecting short 70 nt predicted 23S_rRNA. Adjust via --reject option.
[barrnap] Found: 5S_rRNA NODE_17_length_154231_cov_9.579771 L=110/119 129664..129773 + 5S ribosomal
RNA
[barrnap] Found: 5S_rRNA NODE_1_length_687575_cov_8.340472 L=110/119 51408..51517 + 5S ribosomal RN
A
[barrnap] Found: 5S_rRNA NODE_23_length_103367_cov_10.080628 L=110/119 40653..40762 + 5S ribosomal
RNA
[barrnap] Found: 5S_rRNA NODE_3_length_378948_cov_10.008563 L=110/119 353607..353716 + 5S ribosomal
RNA
[barrnap] Found: 5S_rRNA NODE_5_length_332503_cov_9.112282 L=110/119 133325..133434 - 5S ribosomal
RNA
[barrnap] Found: 5S_rRNA NODE_6_length_298797_cov_8.927468 L=110/119 19051..19160 + 5S ribosomal RN
A
[barrnap] Found 18 ribosomal RNA features.
[barrnap] Sorting features and outputting GFF3...
[barrnap] Done.
```

Next, use `bedtools getfasta` to extract nucleic acid sequences of the 16S rRNA genes from your assembly. You will need to specify the path to contigs.fasta with `-fi` and the path to the GFF file you obtained above, with `-bed`. The output will be in FASTA format. Redirect the output to anew fasta file with a name of you choosing.

In [8]:
```
!bedtools getfasta -fi contigs.fasta -bed 16S_rrna.gff > 16S_rrna.fasta
```

Finally, open your web browser and head over to **Ribosomal Database Project's SeqMatch** tool. Copy and paste your 16S sequences one at a time into the window, or upload the resulting FASTA file from the previous step. This program will attempt to identify the 16S sequences as precisely as possible by comparing them to a database of high-quality, curated sequences, obtained from known bacteria.

You may not be able to obtain a "species"-level identification, but please write down your genus-level identification in your iPython notebook along with an explanation for how you came to this conclusion.

> As it turns out, all 6 of the 16S sequences have their highest matches in the genus Pseudomonas, which is definitive. The breakdown of the phylogeny is as follows:
>
> - domain Bacteria (6)
>   - phylum Proteobacteria (6)
>   - class Gammaproteobacteria (6)
>   - order Pseudomonadales (6)
>   - family Pseudomonadaceae (6)
>   - genus Pseudomonas (6)

Genome annotation

Knowledge of a genome's taxonomic provenance can be used to infer quite a bit about its lifestyle. For example, if this genome belongs to the Mycoplasmagenus, we could take a guess that it lacks a cell wall and has a parasitic relationship to an animal host. Given that we know the sequence of the genome, however, we needn't stop at an analysis based on taxonomic labels. We can look inside the genome and determine with high confidence whether it contains genes necessary to produce a cell wall, or virulence factors enabling the infection of an animal host. To do this, we need a program that can break the long genome sequence into genes, then identify their function by identifying orthologs with known function in other, more well-studied genomes. This is called genome annotation.

In this lab, you will upload the genome assembled with the full set of reads to two remote annotation services that will perform the annotation automatically: **RAST** and **KBase Prokka** Annotation. There are many more annotation services, but for the sake of time, focus on only two. You must create an account to upload your genome to RAST and KBase. This process may take some time.

Research and write-up

Now that you have identified your genome and sent off your genome for annotation, take some time to research what is known about its genus and/or species. Search PubMed for recent publications and read through some abstracts. Once your RAST job has finished, you will be presented with a graphical interpretation of the pathways encoded by your genome, and information about its phylogenetic relatives.

Please summarize some of the information obtained in your annotation in a markdown section at the bottom of your notebook, placing it in the context of any papers you have found and read about related taxa. Use no more than 500 words and no more than two references. Include a word count at the top of the section. Try to make it exciting to both you and the reader. Focus on an interesting topic supported by evidence found during your analysis! Here are some ideas for topics, but feel free to choose your own:

1. What environment do relatives of your bacterium live in? Is there evidence of adaptation to this environment in the genome?
2. Based on its genome, is your bacterium auxotrophic for any amino acids? Are its closest relatives also auxotrophic for these?
3. Horizontal gene transfer is common among bacteria. Is there any evidence for HGT in your genome?
4. CRISPR-Cas9 is so hot right now. Does your genome have a CRISPR system? Can you determine where the spacer sequences originated from?
5. Does your genome encode any known bacteriocins, antibiotics, or toxins?
6. Make an argument for why or why not your bacterium would be considered a human pathogen, using the genome and your research as evidence.
7. Does your genome encode any known antibiotic resistance genes? Do you expect it to be susceptible to penicillin, tetracycline, or chloramphenicol?

The environment relatives of Pseudomonas generally inhabit soil, marshes, costal marine habitats, and also human and animal tissues. Pseudomonas can survive in a variety of ecological conditions, and thus have many phenotypes depending on its environmental condition.

These were metadata of both outputs on PROKKA and RAST on KBase. As you can see, they were not able to make a species match:

Field	Value
Permanent Id	77251/4/1
Full Type	KBaseGenomes.Genome-11.0
Saved by	Alex Bondarenko (abondrn)
Taxonomy	Unconfirmed Organism
Size	6635920
Source	PROKKA annotation pipeline
Name	Unknown
GC content	0.58799
Genetic code	11
Number of Genome Level Warnings	2
Number of Protein Encoding Genes	6077
Assembly Object	77251/2/1
Number contigs	65
Domain	Bacteria
Number of CDS	6077
Genome Type	draft isolate
MD5	f71d54fad3aec00973b26a0c51bee0d0

Field	Value
Permanent Id	77251/6/1
Full Type	KBaseGenomes.Genome-11.0
Saved by	Alex Bondarenko (abondrn)
Taxonomy	Unconfirmed Organism
Size	6635920
Source	KBase
Name	Unknown species
GC content	0.58799
Genetic code	11
Number of Genome Level Warnings	2
Source ID	rast
Number of Protein Encoding Genes	6300
Assembly Object	77251/2/1
Number contigs	65
Domain	B
Number of CDS	6300
Genome Type	draft isolate
MD5	f71d54fad3aec00973b26a0c51bee0d0

Overall, KBase was not terribly useful for species classification, however was able to annotate genes well. The output of those annotations are included in this repository.

When running RAST, this was the chart produced:

In [20]:
```
import pandas as pd

pd.read_csv("rast_species.tsv", sep="\t").head()
```

Out[20]:

	Genome ID	Score	Genome Name
0	205922.3	543	Pseudomonas fluorescens PfO-1
1	205922.5	528	Pseudomonas fluorescens Pf0-1
2	1144321.3	422	Pseudomonas sp. GM102
3	1144324.3	412	Pseudomonas sp. GM18
4	220664.3	404	Pseudomonas fluorescens Pf-5

Pseudomonas fluorescens, likely strain PfO-1, is the species whose reads we have analyzed in this lab. The Gram-negative, nonsaccharolytic bacterium Pseudomonas fluorescens are notable for secreting the UV-fluorescent pigment pyoverdine, as well as having multiple flagella and an extremely versatile metabolism. Pseudomonas fluorescens is the most diverse "species" among the Pseudomonas type strains, and so is often refered to as a "species complex." One way to analyze the level of genetic diversity by looking at the number of genes in the "pan-genome" found across all strains: the pan-genome of P. fluorescens bacteria contained 13,782 genes compared to that of P. aeruginosa.

Found mainly in soil and water, P. fluorescens and in particular strain Pf0-1 have been studied thoroughly for their adaptation to the soil environment and flexible metabolism which enables colonization of a many host types like mammals, the soil and surfaces surrounding plants, nonsterilized drug capsules, and even interior walls and showerheads. Bacteria are able to control population density through **quorum sensing**, the release and sensing of signal molecules that regulate genes related to motility, antibiotic production, and biofilm formation. Two types of quorum sensing systems have been described for P. fluorescens: the AHL/lux (in Gram-negative bacteria, first discovered in the strain NCIMB 10586) and hdtS systems (discovered in F113). In PfO-1, it was found that two transposon insertion mutants were deficient in a gene called adnA. Studies have found that this adnA mutation in the wild-type P. fluorescens Pf0-1 results in a lack of flagella and could therefore constrain the extent of spread, movement, and retention in live soil, and limit biofilm formation.

P. fluorescens has many mechanisms that enable cultures to inhabit mammalian hosts, including synthesis of bioactive secondary metabolites, siderophores, T3SSs, and ability to grow at higher temperatures. P. fluorescens contains a two-component GacS-GacA gene system that plays a role in environmental sensing by regulating the expression of secondary metabolites and enzymes. This system activates the transcription of a unique small ncRNA called RsmY, which combines with a riboregulator (RsmA) to upregulate post-transcriptional expression. Type III secretion systems (T3SSs) are highly conserved genomic clusters common in bacteria (often obtained through **horizontal gene transfer**) which form syringe-like complexes to directly deliver effectors (bacterial proteins) from the bacterial cytoplasm into host cells, and mirrors the type of interaction a bacterium has with the eukaryotes in its environment. The Hrp1 family, the most common T3SS found among P. fluorescens strains, targets plants, leading to namesake **hypersensitivity response and pathogenicity**. The production of cyclolipopeptides (CLPs) by P. fluorescens MFN1032, likely associated with swarming motility, biofilm formation, and colonization, is altered during high temperature regimes, and was been shown to be regulated by epigenetic switches.

Pseudomonas fluorescens is also implicated in some unusual cases of diseases in humans, targetting individuals with a compromised immune system. A notable case was an outbreak that affected 80 patients in the US from 2004 to 2006, which was isolated to saline flushes used with immuno-compromised cancer patients. It has been suggested that P. fluorescens might have a role in the pathogenesis of many inflammatory conditions. I2, a peptide encoded within the P. fluorescens pfiT gene, was found to have elevated levels in biopsy samples from patients with inflammatory bowel diseases. Serum Anti-I2 antibody levels have been positively correlated with the diagnosis of Crohn's disease, celiac disease, ankylosing spondylitis arthritis, and chronic granulomatous disease, and this activity was associated with the therapeutic effectiveness of treatments targeting microbiome compared to the immune system.

1. Scales, B. S., Dickson, R. P., LiPuma, J. J., & Huffnagle, G. B. (2014). Microbiology, Genomics, and Clinical Significance of the Pseudomonas fluorescens Species Complex, an Unappreciated Colonizer of Humans. Clinical Microbiology Reviews, 27(4), 927–948. https://doi.org/10.1128/cmr.00044-14
2. Mulet, M., Lalucat, J., & García-Valdés, E. (2010). DNA sequence-based analysis of thePseudomonasspecies. Environmental Microbiology. https://doi.org/10.1111/j.1462-2920.2010.02181.x

In [ ]: