This is a description of the scripts used for bioinformatic analysis of SMO pathway prevalence in the paper ‘Oxidative desulfurization pathway for complete catabolism of sulfoquinovose by bacteria’ (INSERT LINK TO PAPER HERE).

Input files were prepared as thus (copied from text of paper, addition in bold): Each gene within the *A. tumefaciens* C58 SMO gene cluster (Atu3277-Atu3285) was submitted as a query to the NCBI BLASTp algorithm to search a database comprised of non-redundant protein sequences with A. tumefaciens (taxid: 358) sequences excluded. Standard algorithm parameters were used, except the maximum target sequences was set to 10,000. **Results were downloaded as Hit Table (.csv).**

**evalue\_script.py**

This script filters the input file by a user-specified E-value threshold.

**ipg-final\_new.py**

This script takes all the protein accession numbers in an input file and downloads the corresponding nucleotide accession number(s) and organism name(s) from the NCBI database. You will require an NCBI account and an API key. Any protein accession number(s) that have no corresponding nucleotide accession number(s) will be printed in the cmd window.

**step3\_4.py**

This script combines all the input files for each protein into a single file (referred to as the output file in this document). It then removes duplicates. The output file lists the non-redundant nucleotide accession numbers found in the input files, the organism that nucleotide accession number corresponds to, and what input file(s) the nucleotide accession number was found in (indicated by 1 if found in the input file, 0 if not found in the input file). There is also a column titled Sum, which indicates the total number of input files the nucleotide accession number was found in.

**Row\_Deleter\_5.py**

Due to the nature of the Identical Protein Groups on the NCBI database, even though *A. tumefaciens* protein sequences are excluded from the initial BLASTp searches, if an *A. tumefaciens* genome sequence contains a protein sequence identical to a protein sequence found in a non-*A. tumefaciens* genome sequence, it will be downloaded by ipg-final\_new.py and wind up in the output file. In order to remove these unwanted *A. tumefaciens* sequences, this script is used. It will delete all entries from the organism *A. tumefaciens* from the output file.

**NZ\_Duplicate\_Remover.py**

Some of the sequences with the prefix NZ\_ are redundant – nucleotide accession numbers NZ\_XXX.1 and XXX.1 are the same genome sequence. This script will remove redundant NZ sequences from the output file.

**example\_candidate\_list.csv**

This is an example list of candidate genome sequences. It is an output file from step3\_4.py that has had the Row\_Deleter\_5.py and NZ\_Duplicate\_Remober.py scripts run on it. It is also the list of candidate genome sequences used for the bioinformatic analysis in the paper ‘Oxidative desulfurization pathway for complete catabolism of sulfoquinovose by bacteria’

**multigeneblast\_3.py**

This script is a slight modification of the original multigeneblast.py script from MultiGeneBLAST that results in the displaypage1.xhtml and clusterblast\_output.txt files produced by multigeneblast.py having unique names (XXX.1clusterblast\_output.txt and XXX.1displaypage1.xhtml). This is necessary if you are going to run MultiGeneBLAST in parallel, or an error will occur – the wrong clusterblast\_output.txt and/or displaypage1.xhtml file will be placed in the MultiGeneBLAST output folder (eg. displaypage1.xhtml for YYY.1 will be placed in the folder for XXX.1