

## Query NCBI Taxonomy

To query NCBI taxonomy database and traverse the hierarchy to get names at various taxonomic depths, the fastest tools are to download the [taxonomy DB files](#) and run this as local queries.

See [a working script](#) for one set of queries, also appended below.

This is Perl code and the main module is [Bio::DB::Taxonomy](#) which has several underlying access modes. The fast, local access mode is [Bio::DB::Taxonomy::sqlite](#).

Additional web-query only is supported through a module that talks to the NCBI Entrez database [Bio::DB::Taxonomy::entrez](#). Another uses a DB Hash approach to indexing and querying but is much slower now with the many millions of entries in the database and should be replaced by the superior performance in the sqlite module.

```
#!/usr/bin/env perl
use strict;
use warnings;
use Bio::DB::Taxonomy;
use DB_File;

my %lookup;
my $cachefile = 'names.idx';

tie %lookup, "DB_File", $cachefile, O_RDWR | O_CREAT, 0666, $DB_HASH
    or die "Cannot open file '$cachefile': $!\n";
my $ifile = shift || 'lib/jgi_names.tab';

open( my $in => $ifile ) || die "Cannot open $ifile: $!";

my $namesfile = 'tmp/taxa/names.dmp';
my $nodesfile = 'tmp/taxa/nodes.dmp';
my $indexdir  = '/scratch/indexes';
my $taxdbidx  = "$indexdir/taxdb.db";

if ( !-d $indexdir ) {
    mkdir($indexdir);
}

my $db = Bio::DB::Taxonomy->new(
    -source    => 'sqlite',
    -db        => "$indexdir/taxdb.db",
    -nodesfile => $nodesfile,
    -namesfile => $namesfile,
);
```

```

my $header = <$in>;
print join(" ",qw(Portal Species Strain Taxonomy)), "\n";
while (<$in>) {
    chomp;
    my ( $name, $species ) = split( /\t/, $_ );
    $species =~ s/_v\d+\.\S+//;
    $species =~ s/_$//;
    my ( $genus, $sp, $strain ) = split( /\_/, $species,3);
    $strain = "" unless defined $strain;
    $strain =~ s/[\\)\(\,]_/_/g; # remove , and paren
    $strain =~ s/_+_/_/g; # remove double _

    my $species_string = join( " ", ( $genus, $sp ) );
    my $str = "";

    if ( exists $lookup{$species_string} && length($lookup{$species_string}) &&
        $lookup{$species_string} ne ";;;;;" ) {
        $str = $lookup{$species_string};
    } else {

        my $h = $db->get_taxonid($species_string);

        #print("taxonid is $h\n");

        unless ( $h ) {

            # warn looking up genus instead
            warn("Looking up genus:$genus instead of $species_string\n");
            $h = $db->get_taxonid($genus);
        }

        my $node = $db->get_taxon( -taxonid => $h );

        if ($node) {
            my @tax;
            my %ranks;
            while ($node) {

                #         print("rank=", $node->rank, ". node name is ",
                #         join(" ", @{$node->name('scientific')}, "\n"));

                if ( $node->rank ne 'no rank' ) {
                    $ranks{ $node->rank } = scalar @tax;
                }

                push @tax, [ $node->rank, @{$node->name('scientific')} ];
            }
        }
    }
}

```

```

        my $ancestor = $node->ancestor;
        $node = $ancestor;
    }
    $str = join(";",
        map {
            exists $ranks{$_}
            ? join( ":", @{ $tax[ $ranks{$_} ] } )
            : ''
        } qw(phylum subphylum class subclass family genus)
    );
    $lookup{$species_string} = $str;
}
else {
    warn( "no taxonomy for $species_string or $genus");
    $str = "";
}
}
print join( ",", $name, $species_string, $strain, $str ), "\n";
}

```