Arena Sequence Analysis Protocol

This document describes the sequence analysis protocol developed Brian Saunders for Arena Pharmaceuticals. The analysis employs the NCBI Blast tool and databases, utilizing Perl scripts to tie together the various steps. This was designed for a Windows based system, though with little modification it could run on a Linux system.

The first step of the analysis is to build species specific NCBI identifier lists (by GI number), which then are processed with a Blast tool that builds species specific Blastable databases. The protein sequences from the Arena databases are then compared to these databases (i.e. “Blasted against”) so that the best representative sequence and gene identifiers can be found: these can be Refseq accessions, other NCBI accessions, NCBI Gene ID, and Uniprot ID. The reason for obtaining these IDs is twofold: one for augmenting the Arena database, and two for using as annotation for subsequent sequence analysis steps. This information is stored in a tabular text file (suitable for import into Excel).

Another analysis step, as a Perl script, finds SNP information from dbSNP at NCBI, and then prints it in a tabular text file. It program runs on a singular gene ID rather than a gene list, given the relatively large amount of information it produces. It will use the previously describe annotation file to augment the SNP details with Arena information, but it isn’t required.

Another analysis step compares the Arena protein sequences to simpler organisms (e.g. flatworms) with Blast, to find potential homologs. The Blast commands are run as described later in this document, and then a Perl script is use to analyze the results and print them in a tabular text format.

# Required Software

## Blast

The Blast software from NCBI is required to do the sequence analysis steps. The installer should be available at this address (NCBI’s FTP site): <ftp://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/> . The “-win64” version of the software executable should run on most Windows machines.

A helpful document on installing the software on Windows machines can be found at: <http://www.ncbi.nlm.nih.gov/books/NBK52637/> . When configuring the BLASTDB environment variable, one should use the database directory described below.

## Perl / Cygwin

Any Perl installation should be able run the scripts developed for this analysis, as long as they have a shell-like interface for script execution (especially those using Unix-style paths rather than Windows) and gzip available. Cygwin with a standard Perl installation was used for the script development, and this is probably the safest option. Also, it contains a version of tar needed to uncompress the Blast database files (described below).

# Databases

All the database files described below should be stored in the same directory. It is generally hardcoded into the scripts that were developed, but changing it is relatively easy. To avoid having the script updated for only that purpose, one make the directory a program option rather than hardcoded. The Blast program (described above) requires the definition of a default data directory, which among other things makes it shorter to write the commands.

## Blast

The non-redundant protein Blastable database needs to be downloaded from NCBI. It can be found here: <ftp://ftp.ncbi.nih.gov/blast/db/> , and all files of the form “nr.\*.tar.gz” should be downloaded. In addition, the taxdb.tar.gz file should be downloaded. Once obtained, these files can be uncompressed with a number of file decompression utilities (for example, the tar command in the Cygwin installation). To speed up downloads greatly, one can install the Aspera utility for NCBI, but downloads with this utility can flood a network with limited speed. If using Aspera, one should configure it for only one concurrent download, and preferably do the download after work hours or on the weekend. The Aspera download directories can be found at: <http://www.ncbi.nlm.nih.gov/public/> , navigating to the blast/db subdirectory.

## Taxonomy

In order to build the taxonomy tree for the species definition, a copy of the NCBI taxonomy database needs to be obtained at: <ftp://ftp.ncbi.nih.gov/pub/taxonomy/> . Download the taxdmp.zip file, then uncompress it. The nodes.dmp file is the only one required for the scripts.

## UniProt / InterPro

A Uniprot generated ID mapping file is used in both the Arena annotation and homology finding scripts, and can be found here: <http://ebi.edu.au/ftp/databases/uniprot/current_release/knowledgebase/idmapping/> . Download the idmapping\_selected.tab.gz file. It gets expanded within the Perl scripts so it does not need to be expanded beforehad. A Uniprot to Interpro ID file is used by the homology finding script, and can be found at the same web site: <http://ebi.edu.au/ftp/databases/interpro/> . Navigate to the most recent release and download the protein2ipr.dat.gz file.

This is an EBI mirror site in Australia. One could find these files in the main EBI ftp site, but the download speeds from there tend to be a lot slower. At some point in time, Canada had an EBI mirror and downloads from there were quite a bit faster, but that mirror no longer exists.

# Arena Annotation

## Formation of species-specific Blast databases

In order to the relevant Blast comparisons, it is necessary to form species-specific Blast databases. This needs to be done for two purposes: to set up the appropriate target databases to find the best representative Arena protein database IDs (e.g. human, mouse), and to set up target database for finding homology (e.g. flatworms, fungal). The script which builds these databases does two things. First, it first dumps the GI numbers and taxonomy IDs for each protein in the NCBI non-redundant protein database, and writes to separate species lists (see below), and it then builds Blastable sub-databases for each species ID list.

To do the species separation, a file needs to be input that contains 3 columns with tab delimination: root NCBI taxonomy ID, Blast database name, and classification name (the latter is not used for this step, but is used for subsequent steps). If a root taxonomy ID has children, then all of the sequences with any child IDs will be in the Blastable database as well (for example, choosing the ID 10114, which is the Rattus genus, will result in over 100 species and strains being included in the database, e.g. Rattus norvegicus, Rattus rattus.

To run this script, the user should go to the database directory, and then execute the “gi\_multtax.pl” command. The input to the command is the taxonomy file described above. Assuming the taxonomy file is named “spectaxon.txt” and is in the database directory, and the command located in a separate code directory, the command would be executed in the following:

/cygwin/u/code/gi\_multtax.pl spectaxon.txt

This will create a number of Blastable databases limited to the species groups we defined in spectaxon.txt, using the Blast database names in that file. The script generally takes around 2 hours to run on a relatively modern PC.

## Sequence input

All of the protein sequences were dumped from the Arena database into a Fasta format file, with the Arena ID as the header of the sequence. This entire file was split into species-specific files, so that each species could be Blasted against its corresponding Blastable database. The script used for splitting the entire sequence file is named “arsplit.pl”, and is executed in the following manner, given the entire file has the name “arena.fa”:

/cygwin/u/code/arsplit.pl arena.fa

Given that the Arena database likely won’t change the sequence for any given entry, this should not need to be rerun as long as one shares each file between data runs. The homology finding step described later can be run either on the entire database or just a particular subset (e.g. the human sequences).

## Arena species Blast

A script which runs a Blast for each species sequence file against its corresponding species database is called “run\_arbl.pl”. It uses tabular Blast output in with a specific defined format, which is then used by the other scripts which read in that output. Its input is the taxonomy definition file described above.

Generally, each run of the analysis should be in its own directory. Since the file names are fixed, multiple runs would result in the old result files being overwritten. The directory should contain a copy of each Arena species specific sequence file, with the command being executed in the following matter:

/cygwin/u/code/run\_arbl.pl spectaxon.txt

## Arena sequence annotation

Once the Blast results files are obtained, they should be processed by the arenaprot.pl Perl script. This script goes through the Blast results, and attempts to pick the best representative sequence (and its corresponding database IDs). The top hit is usually the best one, but in some cases it isn’t. When there are multiple splice variants all giving the same hit score, the hit with the closest sequence length to the input sequence won’t always be ranked at the top, so the script needs to account for that. Also, if the Blast hit is a sequence which only has record IDs that are from the pir or prf databases, it should be skipped because those aren’t going to be linked to entities such as NCBI Gene IDs.

NCBI Gene IDs are found from database linkages to the sequence IDs (the script uses NCBI Eutilities via a web interface to do this). Usually this results in a unique Gene ID, but in a few cases no Gene ID is found, or multiple Gene IDs are found.

If there are Refseq curated record IDs associated with the chosen hit, then they are selected as the representative NCBI ID. Otherwise, all the NCBI accession IDs associated with that record are chosen. If a hit has SwissProt (curated) IDs associated with it, those are chosen as the representative Uniprot ID. If not, then all Uniprot IDs related to the each NCBI database ID for the sequence are returned.

The command is run on all of the Blast output files concurrently, and prints its tabular results to standard output. Usually this gets redirected to a file:

/cygwin/u/code/arenaprot.pl \*.blnr > arenamatch.txt

This file can be imported to Excel for more detailed analysis, as well as used as input for other programs.

# SNP Analysis

This Perl script uses dbSNP from NCBI as its basis for SNP analysis. Because of the large amount of information for any individual gene, it works with only one gene at a time – but a master script could run through a list of genes if necessary and call this script repeatedly.

The script uses the NCBI E-utilities to link the specified Gene ID with dbSNP. A structured XML format for each dbSNP record is returned, and the record is parsed to obtain important information, such as the type of SNP, coordinates of where it occurs, the frequency at which it occurs (if provided), and other types of verification. The output is directed to a file with a defined name, and a summary of the SNPs is directed to a second file. The input to the program is simply a NCBI Gene ID, and an option input file is the Arena sequence annotation file generated above, allowing Arena information to be added to the output. The program can be run without that annotation file, though, so that any gene can be used outside of the Arena database as well. Again, this should be run in the working directory as described above, as the program could overwrite previously existing files. The script is named “snpfgene.pl”, and the example below uses Arena 21Z (CNR2 or CB2):

/cygwin/u/code/snpfgene.pl 1269 arenamatch.txt

The \*\_snp.txt (1269\_snp.txt in the example above) files that are generated contain the detailed information for each SNP. The \*\_infosnp.txt (1269\_infosnp.txt in the example above) file contains summary information (which is also sent to the screen).

# Homolog Finding with Blast

The taxonomy definition file defined above is also used as the basis for making Blastable databases for particular groups of species, for example algae, fungal species, flatworms, and protozoan species. The Arena protein sequences can be compared to these databases, to determine if there is strong homology between the GPCRs in Arena’s database and proteins in those targets.

A Perl script named “run\_eukbl.pl” is used to execute the Blast jobs. It can be used with the entire group of Arena sequences, or just a specific species (e.g. human):

/cygwin/u/code/run\_eukbl.pl spectaxon.txt human

Once the Blast results are generated (in the same tabular file that was generated for annotating the Arena database), they can be processed with the “parseepprot.pl” script. This program will also scan a UniProt to InterPro list to see if any of the results match GPCR domains, or have any shared domains with the Arena sequence. Three files get generated: prefhits.txt shows any proteins matching a subset of Arena proteins that are of most interest (defined within the script), allmatches.txt contains each unique sequence that matched the Arena proteins with Blast, and allblhits.txt contains detailed information on every Blast match between the Arena proteins and the target protein databases.

These particular files have predefined names, so the code will need to be modified to come up with specialized names, or subdirectories should be used when multiple Blast runs want to be analyzed separately. Here is an example run of the command:

/cygwin/u/code/parseepprot.pl spectaxon.txt armatch.txt human\_algal.blnr human\_flatworm.blnr human\_fungal.blnr human\_protoz.blnr