Discovering new targets and drugs for neglected diseases by paralog matching.

Author: Jeremy B. Singer

Affiliations:

Jeremy B. Singer is a private scholar studying for a masters in bioinformatics at Brandeis University. He works as a Senior Database Engineer at LifeIMAGE, and can be reached at [jbsing@brandeis.edu](mailto:jbsing@brandeis.edu)

# Abstract

This paper describes a method that can be used to discover and repurpose existing drugs and drug targets by discovering cross species genomic sequence similarities. It uses public domain databases (MyCheEMBL, EnSEMBL, NCBI) and open source software to find measures of sequence conservation with existing targets.

This method can be applied to pathogens with at least a medium sized genome (several thousand genes.) *Neglected tropical diseases* caused by pathogenic protists are good subjects for this approach because many have genomes of sufficient size and because many have genomic features in common with organisms for which there are known targets.

The genome of the apicomplexan parasite *Plasmodium falciparum*, which is responsible for the most virulent form of malaria, was chosen to validate a method that identifies paralogs to existing disease targets because it has known cross-species targets.

MyChEMBL is a downloadable self-installing platform that contains a PostgreSQL collection of thousands of protein targets, and includes the sequences of their associated proteins. Using relational database and open source software, the methods described in this paper identified 29 distinct drugs and 592 targets validating this approach.

Five other pathogens (*Trypanosoma Brucei, Trypanosoma Cruzi, Leishmania Major,*  *Chlamidia trachomatis, and Toxoplasmosis Gondii)* were also downloaded and run through the same pipeline, identifying potential targets and drugs.

# Introduction

The *World Health Organization* has said that “In 2015, there were an estimated 214 million new malaria cases worldwide, causing 438,000 deaths.” *Plasmodium falciparum*, an obligate apicomplexan parasite, causes the most acute form of this disease. Mutations decreasing the effectiveness of existing medicines increase the need to discover new ones. The author has chosen *p. falciparum* as a model to validate a method to repurpose existing targets and drugs.

Protein targets are those proteins that provide vital functions to the organisms to which they belong. Because they provide vital function, we expect that their amino acid sequences would be conserved.

The “targetness” of a protein has to do with how indispensable its function is to the disease organism. This description of the nature of protein targets suggests that paralogous proteins in our organism of interest could also be targets, if they are sufficiently similar to existing targets. The closeness of the match will suggest that the function of the protein has been conserved between the previously identified target organism and our organism of interest.

MyCheMBL, a downloadable virtualized platform, contains a database of targets and target sequences in a PostgreSQL database that also contains molecule and drug information. With suitable similarity metrics and discriminant criteria, we can distinguish new targets for our organism of interest, as well as drugs suitable to treat disease related to those targets. However, by itself, MyCheMBL does not contain everything we need to discriminate these targets.

*BLASTP,* a protein sequence search and alignment tool, finds matches for query sequences while scoring matches for *conservation* . Querying a *BLASTP* database with all the protein sequences from an organism produces reports from which we can extract similarity scores. Creating a new table, *blast\_statistics* , enables us to analyze the target similarity data using the “*score”* metric as a measure of conservation.

# Literature Review and Introduction

### The need for Drug and Target Repurposing Methods for Neglected Diseases

Early methods for seeking new drugs or repurposing existing drugs were largely a process of trial and error, either noticing effects of substances on disease or actively screening substances against pathogens in vitro or in vivo.

While targets in the most general sense could include any component of infection, transmission, or other disease mechanism, for the purposes of this paper, we are referring to proteins or genes that may be affected by drugs.

As our knowledge of biological mechanisms at the molecular level have improved, it has become possible to identify properties of drug molecules which may be used to predict what effect they will have on target organisms. Knowledge of these mechanisms includes understanding which molecules of the organisms are the targets of these drugs.

Computational methods can save considerable time and expense over wet-lab screening methods by re-using existing experimental data to uncover new features and relationships. In contrast to “in vivo” and “in vitro” methods, these methods can be referred to as “in silico” (referring to computation done using computers.)

Current computational methods of repurposing drugs and targets for infectious diseases follow three approaches:

1. Annotation-centric approaches identify targets based on existing knowledge and annotations for genes/proteins (such as structure, protein family, process, and evolutionary relationship).
2. Ligand-centric approaches search ligand databases according to chemical similarity to known drugs as well as ligand annotations.
3. Hybrid approaches combine gene/protein similarity with ligand similarity approaches to find new opportunities for known drugs.

## Knowledge based, Annotation-Centric Approaches

### Reduced ribosomes of the apicoplast and mitochondrion of Plasmodium spp. and predicted interactions with antibiotics

( A. Gupta, et al, Open Biol. 4: 140045. <http://dx.doi.org/10.1098/rsob.140045> )

This paper by Ankit Gupta, et al, exemplifies a knowledge centric approach to identify likely targets and drugs against infection by the malaria parasite, *Plasmodium falciparum.*

“Apicomplexan protists such as Plasmodium and Toxoplasma contain a mitochondrion and a relic plastid (apicoplast) that are sites of protein translation.” Thus, ribosomal proteins provide a focus for analysis and curation as potential cross-species targets because of their similarity with bacterial proteins to which they are closely related.

(see Figure 11: Plasmodium structural diagram showing organelles.)

Gupta, et al conducted a survey of available sequences of apicomplexan apicoplast genomes, comparing ribosomal proteins encoded by different species. They applied various methods to identify shared proteins; for example:

“To identify ribosomal proteins and ribosome assembly factors, we searched apicomplexan genomes using the GenBank nonredundant nucleotide and CDS translations [88] using TBLASTN and BLASTP, respectively. We additionally performed direct alignments between protein sequences and organellar genomes using BLAST2SEQ….”

They include a Venn diagram in their report showing which of these ribosomal proteins are shared by various protists (See Figure 4 A five-set Venn diagram showing the distribution of nuclear- or plastid-encoded ribosomal proteins that would constitute the plastid ribosomes of apicomplexans P. falciparum and T. gondii, red alga C. merolae, green alga C. reinhardtii and diatom T. pseudonana.)

### Results and limitations of this approach

The proteins translated from these genes are known malaria targets:

“Several antibiotics, including clindamycin, chloramphenicol and the macrolides erythromycin and azithromycin, bind in the vicinity of the ribosome LSU peptidyl transferase centre or the peptide exit tunnel and inhibit parasite growth. This group also includes thiostrepton that contacts ribosomal protein L11 and the GTPase region of 23S rRNA. Translation inhibitory antibiotics have two putative target organelles, the apicoplast and mitochondrion, of the parasite. Some antibiotics (e. g. clindamycin, azithromycin, chloramphenicol and tetracycline) have been demonstrated to have a delayed-death effect, a phenotype associated with apicoplast-specific action... Thiostrepton causes immediate parasite killing and is proposed to have additional targets in P. falciparum.” – Ankit Gupta, et al.

This is a good example of a focused, knowledge based approach used to identify putative targets and potential drugs. It uses phylogenetic and detailed knowledge of the organisms to direct searches for genetic commonality leading to common drug targets.

The disadvantage of this approach is that its applicability is narrowed to the knowledge and expertise of the researchers and to a specific small set of organisms. This is a highly supervised approach.

## An Ontology for Pharmaceutical Ligands and Its Application for in Silico Screening and Library Design

(Schuffenhauer, et al J. Chem. Inf. Comput. Sci. 2002, 42, 947-955)

Annotation-centric approaches have progressed from individual expert curation of drugs and targets to categorization schemes for drug and organism ontologies. By applying these ontologies to drugs and targets, investigators have developed searchable databases for screening targets for drugs.

Drug ontologies relate classes of proteins hierarchically by structural similarity annotations. The authors of this paper created a drug screening database that connects genomic targets with ligands by extending known ligand / target relationships ranking similarity of candidate compounds to known drugs.

The investigators used the MDDR01.1 database, which includes target information for many of its ligands, and applied a chemical similarity ranking process to annotate additional ligands-target annotations for four target families based on references established by the EC, GPCRDB, NuclearDB, and LIGCDB organizations. The resulting Oracle database can use “connect by” syntax to relate compounds hierarchically within the protein ontology model, extending its annotations to the candidate compounds.

Ligand similarity uses 2D Unity fingerprints to represent drug compounds. The proprietary Unity fingerprints are created by analyzing ligands into fragments and associating them with specific bits in the Unity fingerprint. Presence of a particular fragment sets the corresponding bit in the 988 bit fingerprint to “1”. Tanimoto (Jacqard coefficient) comparisons can then be used to rank the compounds.

Tanimoto coefficients provide a similarity ranking for sets that are represented by bit strings. This measure is defined by the ratio of intersection of elements divided by the union of elements of two compared sets. These set elements can be represented by bit strings, such as the Unity 2D fingerprint. The formula for Tanimoto coefficient is:

Where A and B are sets to be compared (represented by bit strings) and C are the elements in common, obtained by ANDing bit strings A and B. The bit strings are all the same length.

The function n() counts 1 bits in a string.

To compute this operation, first find the intersection of A and B by doing bitwise AND.

After performing these bit operations, the coefficient is computed by counting the 1 bits in each of the bit strings and computing their ratio, as indicated in the formula above.

In chemistry and in biology, molecular similarity predicts activity well. For an in-depth treatment of the applicability of these similarity measures, see *Why Is Tanimoto Index an Appropriate Choice for Fingerprint-Based Similarity Calculations?* Bajusz, et al, *Journal of Cheminformatics* 7 (2015): 20. *PMC*. Web. 4 Nov. 2017.

### Accomplishments and limitations of this approach:

“Of the 799 activity keys used in MDDR01.1, 309 could be linked to a target in our classification scheme. This allowed us to annotate 53 211 of the total 113 821 compounds within MDDR01.1.”

However, “…there are targets still missing or incomplete in our ontology, the most important of these are

the protein kinases, which need to be classified more in detail as well as some oxidoreductase families like the cyclooxygenases or the monoamine oxygenases. Missing completely in the ontology are the signal transducing membrane receptors such as the cytokine receptors.”

Relative ranking of Tanimoto scores was used to categorize known targets, but not to assign targets to any particular organism. Knowing of existing target mechanisms, one could choose newly identified ligands.

Sheridan and Kearsly also noted that “the most important limit of patent-based databases is ‘false inactives’: only one or two activities are reported for any given compound but that compound might actually be active in another area, if only it were tested.” (see “Why do we need so many chemical similarity search methods?” *Drug Discov. Today* **7**, 903–911 (2002).

## Ligand-Centric approaches

### Calculating Similarities between Biological Activities in the MDL Drug Data Report Database

(Sheridan and Shpungin, *J. Chem. Inf. Comput. Sci.* **2004,** *44,* 727-740)

The investigators used automated methods replacing hand-curation approaches to find connections between biological activity and ligands by data mining annotations. Key words were stemmed (suffixes removed) and related to other ligands by chemical and biological activity words in common.

“LSI (Latent Semantic Indexing),7 a branch of LSA (Latent Semantic Analysis),8 is a linguistic method for document searching that uncovers latent relationships between documents TIMI (Text Influenced Molecular Indexing) is a chemistry-oriented extension by which one can uncover latent relationships between documents, molecules, and chemical descriptors.”

“For this exercise we used three flavors of the **X** matrix: TIMI-WC (words and chemistry) has all the terms described above. TIMI-W (words only) has the activity labels and words, with no TT descriptors. TIMI-C (chemistry only) has the activity labels and TT descriptors with no words. “

TIMI uses Cosine similarity, where discrete words belonging to chemical entities are compared. Cosine similarity does not require that compared sets have the same length, which is an advantage over Tanimoto similarity.

The probability that compounds are shared (SC) is:

“We can think of each activity label *i* is a list of compounds with that activity, or as a vector with =1 if compound k is active on I and = 0 otherwise.”

*Trend Vector* *Similarity* is a QSAR (Quantitative Structural Annotation Regression), machine learning method. A trend vector is the one-dimensional array of correlations between the biological activity of interest and a set of properties of compounds in a training set.

“Generating a QSAR requires constructing a training set for each activity label. We randomly chose up

to 1000 molecules with the activity label. These were given a response of “1”. We then added 1000 randomly selected compounds without the label. These were given a response of “0”.”

Some linguistic analysis and preparation were necessary to make word annotation sets suitable for this analysis. For example, annotations that might otherwise correlate as the same (agonist vs. antagonist) needed to be mapped to a normalized terminology (i.e. Inhibiter = Blocker=antagonist, stimulant = agonist = promoter). (Authors note – this resembles practices used in sentiment analysis for text mining used in social media analysis.)

Ibrahimov clustering algorithm was used to assign objects to clusters (See *A NOVEL SIMILARITY BASED CLUSTERING ALGORITHM FOR GROUPING BROADCAST NEWS*, Ibrahimov et al.) This clustering method uses random walk techniques that take into account global similarity as well as near neighbor similarity, which improves its sensitivity.

### Results and limitations

“LSI knows only about word co-occurrence, ignores word order, and knows nothing about the meaning of the words. This useful behavior has also been previously demonstrated for words and descriptors in our derived method TIMI. However, such methods suffer from a number of drawbacks. First, results depend on *k*, and it is not obvious a priori what value of *k* is optimum for a given problem.”

TIMI requires some supervision to produce meaningful results.

TIMI and trend vector approaches produced the best results. Results were clusters of closely related compounds. These approaches required some supervision and judgement to decide how many neighbors should be used to define a cluster.

To use this for repurposing drugs for neglected disease organisms, one must choose known targets for that organism and see if new ligands have been associated with it.

## Hybrid approaches

### SEA (Similarity Ensemble Approach)

This method was described in a paper by Keiser, et al, “Relating protein pharmacology by ligand chemistry.”

This approach combines ligand similarity and protein similarity approaches to find ligands that may have similar effects to known ligands for a given target. It builds on the method described previously by Schuffenhauer, et al, in comparing activity classes. It combines these comparisons with sequence comparisons between the targets, creating similarity ensembles that create islands of similarity qualified by annotation similarity, chemical similarity, and target similarity.

The investigators “mapped 193 MDDR activity classes to their protein target sequences and determined

the sequence similarity among them using PSI-BLAST”. They then “computed a heat map highlighting the differences between pharmacological similarity and sequence similarity among these targets. In this heat map, many ligand sets with enzyme targets were pharmacologically similar but sequence dissimilar.”

SEA was able to find unexpected activities that were not indicated by MDDR, and which were verified experimentally.

#### Methods

Multiple ligands in any annotation defined a set of functionally related molecules.

Only sets containing five or more ligands were used.

nonmolecular targets were excluded (e.g., the annotation “Anticancer” was not used).

All ligands were represented as SMILES strings.

Similarity scoring of pairs of ligands used scoring methods inspired by BLAST.

Tanimoto coefficients (Tc) were used to compare standard two-dimensional topological Daylight fingerprints.

All annotations in a given database were exhaustively compared against all others, resulting in a matrix of SEA E-values among the ligand sets, each of which defined a strongly connected graph. It was filtered by removing all edges with significance less than an E- value cutoff of 1.0. (This is called a “threshold” graph.)

A minimum spanning tree was created using Kruskal’s algorithm, which the investigators called a “similarity map.”

Protein sequences were obtained for the activity classes from several sources and sequence similarity computed with PSI-BLAST.

The SEA E-value matrix was compared with the sequence comparison E-value matrix by taking the difference of the natural logs of each E-value pair. The authors noted that it was necessary to confine the E-value range between 1x 10-50 to 1x105 to avoid math range errors. *A smaller E-value cap would allow for greater resolution of high-end E-values... but would be at the expense of differentiating from insignificant similarity.(*italics mine.)

## Review conclusions

Focused, knowledge based approaches will continue to have a place when searching for new targets and drugs, as well as for reusing existing targets and drugs. Gupta, et al, were able to produce high quality results through this highly curated effort, identifying common ribosomal features in protozoans that they were expected to share targets with bacteria. They used sequence similarity to find these shared features between the ribosomal proteins in the protists, and related them to bacterial ribosomal targets.

There are two main disadvantages of this method. The narrowness of focus means that only one kind of target (ribosomal) was considered, only those that are shared with bacteria, and only with certain kinds of protists were in the study set. The second disadvantage is that a high level of curation is involved in this kind of investigation – the level of effort to duplicate this effort for some other organism set or type of target is high.

Schuffenhauer, et al created a database that embodies a hierarchical protein ontology that makes it easier to search, en mass, for ligands and targets for those who are who know what they are looking for. This approach enables automated screening for ligands with the desired characteristics based on this ontology.

There are several drawbacks to this approach when used for drug and target repurposing:

1. The ontology may not have enough categories for the searcher. Specifically, it is not clear how useful this ontology is for infectious disease targets.
2. Processing the annotations requires a fair amount of supervision and expertise.

Lastly, the SEA method described by Keiser, et al provides a comprehensive method combining knowledge about sequence similarity of the targets and ligand similarity which correlate with each other providing a more comprehensive approach to re-using targets and drugs.

SEA was able to find unexpected activities that were not indicated by MDDR, and which were verified experimentally.

# Materials and Methods

In contrast to the methods described in the literature review, this paper demonstrates a fourth approach, which is genome-centric . This approach relies on the sequence similarity of the translated genes of disease organisms to known target proteins. These proteins sequences act as keys, revealing their “target-ness” by similarity to proteins in a targets database. The author has used MyChEMBL’s *PostgreSQL* database, which links target protein sequences to annotations and drugs. This approach extends the usefulness of these associations by indicating new associations to proteins in other organisms of interest.

## The computation platform

All work was performed on a consumer grade laptop PC, with 12 GB of ram and 1 terabyte of disk storage, with an Intel® Core™ i7-5500U CPU running at 2.4 ghz clock. This CPU has 4 cores, which can improve performance when multiple virtual machines are running. The ram configuration leaves plenty of RAM for multiple VMs (Virtual machines) to run simultaneously. This laptop runs Windows 10. (See Figure 5:Hardware Configuration.)

Most bioinformatics software runs under Linux. Using Oracle’s *VirtualBox* software, multiple Linux virtual machines can be supported on a PC.

## Setting up a Linux Virtual Machine using Oracle’s VirtualBox software

Download the latest VirtualBox software from Oracle (available at <https://www.virtualbox.org/> .)

Windows update can cause this software to stop working, which requires downloading newer software.

A CentOS 6.5 x86\_64 vdi image was downloaded from <https://virtualboximages.com> . This virtual device image is compatible with *Guest Additions* that makes mouse integration between Windows and a running Linux virtual machine easier.

A VM was created with 2gb of ram, using the downloaded VDI. This VM has its own IP address, which can be reached by *ssh* software installed on the PC. (See Figure 6: VM configuration parameters.)

A user account was created with *sudoer* privileges. This account was used for all other setup activities. The *sudo* command can be used to assume *root* privileges temporarily to manage resources on the VM for a command’s duration, and then returning to lower user privilege for safety. Typically, applications run with lower privilege using only resources that are allowed for them. (See Linux user setup.)

## Installations

The following tools were used for the analytical pipeline:

### Install tools on work Linux VM.

* Glimmer3.02 – identify genes.
* EMBOSS Tools (version 6.5.7) – to translate genes to peptide ORFs.
* BLAST Tools - ncbi-blast-2.3.0 – to score similarity of peptides to targets.
* PostgreSQL server and client – create a blast\_statistics table on the work server first.
* Install MyChEMBL 20 vagrant VM. (See Installing MyChEMBL VM.)
* Install ssh client on PC to communicate with work VM.
* Install PostgreSQL client on PC to communicate with MyChEMBL PostgreSQL database.

## Analytical Workflow

(See Figure 8: Genome target analysis workflow)

The target blast database was created on the work VM using target sequences extracted from the *chembl\_20* PostgreSQL database provided by the *myChEMBL* VM.

The genome of *plasmodium falciparum* was downloaded to the work VM.

The 14 chromosomes of *p. falciparum* were downloaded from this URL: <ftp://ftp.ensemblgenomes.org/pub/protists/release-32/fasta/plasmodium_falciparum/dna/>

Mitochondrial dna sequence was downloaded from here: GenBank: AJ276844.1 <https://www.ncbi.nlm.nih.gov/nuccore/8346980?report=fasta>

Apicoplast sequence was downloaded from here: Plasmodium falciparum genome assembly, organelle: plastid:apicoplast

NCBI Reference Sequence: NC\_030754.1

<https://www.ncbi.nlm.nih.gov/nuccore/1052489052?report=fasta>

*Glimmer3.02* identified putative genes using a bundled script *g3-iterated.csh*. Some installation tweaks were required to make the build work. Training sequences from p. falciparum were provided to glimmer, and the full genome was processed into files named according to the scheme *<chromosome>.gene*, with each *.gene* file containing predicted genes locations. Mitochondrial genes were named *mt.gene* and apicoplast genes were named *apicoplast.gene*.

ORF files (Open Reading Frame) were created using *transeq (*from EMBOSS-6.5.7), creating 12,480 files under directories named by chromosome or organelle. These files are peptide *.FASTA* formatted files.

### BLASTP searches, aligns, and scores each ORF against the BLASTP database.

A bash script runs the **BLASTP** command for each ORF in all the chromosome/organelle directories, creating an alignment report for each ORF named *<ORF name>.blastp.txt*. (See BLAST.) This query produces the top 10 target sequence alignments for each ORF.

### An Extract Translate and Load (ETL) process populates the BLAST\_STATISTICS table in a PostgreSQL database on the work VM.

For each ORF, a file named like *<ORF>.blastp.stats* is createdby blast\_stats.sh, which runs Perl script **extract\_header.pl.** that extracts the blast statistics from the BLASTP reports.

The **create\_populate\_blast\_statistics.sh** bash script assembles the statistics into the necessary plsql command file **populate\_blast\_statistics.**sql that will be used to load the **blast\_statistics** table, once the table is created.

Using the *psql* client, the **blast\_statistics.sql** table creates the *blast\_statistics* and *tmp\_blast\_statistics* tables. The *tmp\_blast\_statistics* table is a temporary data table that is used like a bucket to do inserts of one ORF at a time into the *blast\_statistics* table. After each insert , the tmp\_blast\_statistics table is truncated.

Finally, the **populate\_blast\_statistics.sql**file is run, populating the *blast\_statistics* table.

Once the *blast\_statistics* table is populated in the database on the work VM, it can be exported and re-imported into a similarly created table on the myChEMBL VM.

After *blast\_statistics* are populated in the *chembl\_20* databases, we can analyze the data more easily using relational database tools, and R.

# Results and Discussion

## Sequence Similarity approach used in this paper

In contrast to the SEA approach, which includes comparing sequence similarity between targets, the approach used in this paper exhaustively compares sequence similarity between all genes in a disease organism with a database of known targets.

The author noticed that that the principles of sequence conservation and “target-ness” share commonality with respect to infectious diseases. Those sequences in the pathogen organism which are most necessary for its survival are also least likely to change, as mutation would tend to impair functions necessary for survival. At the same time, we are searching exactly for those critically necessary proteins as targets for drugs that can impair them.

At the same time, we do not wish to target those sequences that are also necessary for survival of the patient that is infected. Fortunately, our target database provides us many target sequences that have already been somewhat qualified; by searching for similarity between them and other pathogens, we may be able to find targets that can be reused.

The method described in this paper relies on two complementary statistical principles:

1. A statistical measure of similarity, as implemented by BLAST and BLASTP can provide a robust method of measuring pairwise similarity and conservation between a database of known target sequences and genes found in a pathogenic organism.

2. Understanding the implications of normality within the population of BLASTP scores provides a method for assigning a significance threshold to the scores. Scores above this chosen threshold are likely to be similar enough to known targets to be targets themselves.

## Understanding BLASTP scoring

BLASTP scoring is closely related to the concept of *information content* and *Shannon Entropy* involved between matches of two strings of a given alphabet having a certain frequency distribution.

iyj) log( P(xiyj) / P(yi))

“H” is the entropy of string X given string Y, where p(xi,yj) means the probability of x=y.

“b” ary entropy takes into account the number of letters in the alphabet used by the strings being compared.

So Iyj) logb(p(xi,yj)/p(yj))

Where S is a source alphabet S =(S,P) with source alphabet S = { a1..ab} and probability distribution P= {p1..pb} .

For example, for nucleotides, b=4, and for amino acids, b=20.

BLASTP has been tuned to take conservation of amino acids into account, since substitutions of amino acids by other amino acids have different likelihoods. Transition matrices, such as BLOSUM62 (the default transition matrix used for the ETLs described in this paper) define the appropriate values for the probabilities of matches and substitutions. BLASTP uses these log odds probabilities to assign a positive integer score for each match or conserved substitution, or an appropriate penalty for each non-match, approximating the calculations above. BLASTP uses additional scoring strategies, such as including gap penalties to the scores.

BLASTP also computes the *expected* values, which can be very small numbers. In contrast, the *scores* values are always integers that are easily comparable.

Other statistics computed by BLASTP are *identities*, which are the percentage of exact matches, and *positives*, which are inexact matches that conserve function because the amino acids involved in the comparison function in a compatible manner to each other in the protein.

An advantage of the *score* statistic is that it is additive. While both statistics reflect the cumulative values for matches, products of probability scores that accumulate in the *expect* statistic can lose resolution during computation. (See Figure 10: Pairwise comparison of BLAST statistics.)

### How implications of normality affect the choice of a significance threshold

Choosing a match metric and threshold is a conundrum. What determines whether a match is sufficiently close to suggest that it is likely to be a target?

While a very close match of the *positives* or *identities* (100% or 99%) would indicate that a match identifies a target by tautology, lower scores are hard to justify. A small sequence with a fairly high percentage may still not be as likely a match as a much larger string with a lower *positives* score and a much higher *score* statistic, as the cumulative score takes into account the greater difficulty of achieving a high score for the larger string.

The distribution of log(*score)* for all the matches in *p. falciparum*  appears to be somewhat normal. (Figure 12: Plasmodium falciparum score histogram.) There are several reasons why we should expect these results.

1. The distribution is not symmetric around the mean because BLASTP discards matches where the score is too low, and only returns up to the ten highest scores per query.
2. Normally distributed matches are what we would expect due to random mutation for non-conserved sequences, and account for the normal appearance of the graph.
3. Due to conservation of essential peptides, there may be more high scoring matches than predicted by normally distributed processes. We can look for these likely target candidates in the area of the distribution which normally should be approaching zero asymptotically.
4. In addition, the target universe may be “lumpy” in the sense that there may be more than one centroid that has a family of similarities in the *p. falciparum* genome. This “lumpiness” is not important for purposes of finding unusual target similarity, as it is buried in a normal cohort.

It is interesting to note that the smallest *score* value for a comparison that has a 100% positives value is 20.8, which is close to the median of 27.7 ( or log values of 3.03 and 3.32). Searching the upper tail means missing some targets that are closer to the median.

Median and Median Absolute Deviation were used as measures of centrality and dispersion rather than mean and standard deviation because these measures are more appropriate considering the asymmetry and contamination of the distribution (in the sense that the distribution contains more than a single normal distribution plus other non-normally distributed data.)

## How genome files were chosen.

Malaria was the initial disease of interest. The review by K.T. Andrews, et al (Drug repurposing and human parasitic protozoan diseases) influenced the choice of the other organisms examined here.

The author investigated two apicomplexan species (*p. falciparum, toxoplasmosis gondii)*, two trypanosomal species (*trypanosoma brucei, trypanosoma cruzi)* as well as two other species (*chlamidia trochamatis, leishmania major).*

The author analyzed top level dna datasets for gross statistics about each genome; for analyzing gene details and amino acid distribution, individual chromosome and plastid genomes were translated and analyzed.

Genome file sets are offered in several flavors; to reduce non-significant matching, translations were done from genome files in which random repeats were masked.

## Preliminary whole genome statistical comparison.

Statistics were computed using R functions for nucleotide frequencies (See **nucleotide\_frequencies.r** in supporting information) and a Perl program to calculate amino acid frequencies (See **residue\_frequencies.pl**.) Charts show relative abundance of nucleotides (Figure 1: Nucleotide Frequencies Across Species), relative amino acid frequencies (Figure 2: Amino Acid Frequencies by species), and relative genome size (Figure 3: Genome size comparison by species.)

Nucleotide frequencies are always paired, *adenine/thymine (A/T)* and *guanine/cytosine (G/C)* because of DNA’s complementary base pairing.

*Plasmodium falciparum,* the malaria organism, shows an unusually high *A/T* content of 80% (40% adenine, 40% thymine.) *P. falciparum’s* amino acid frequencies also show over-represented members. The top three overperforming amino acids are *asparagine (N) 14.23%, lysine* (*K*) 11.67%, and *isoleucine (I)* 9.72%. The codons for these amino acids account for the high A/T content of the genome:

|  |  |  |
| --- | --- | --- |
| **Amino Acid** | **Abreviation** | **Codons** |
| Asparagine | N | AAT, AAC |
| Lysine | K | AAA, AAG |
| Isoleucine | I | ATT, ATC, ATA |

Of these 21 codon amino acid positions, only 2 are G/C.

Even though *p. falciparum* and *toxoplasmosis gondii* are apicomplexans, they do not share similar nucleotide or amino acid distributions. Their genome sizes also differ greatly (23,263,391 bp for *p. falciparum vs* 62,966,896 for *toxoplasmosis gondii.*

Despite differences in genome size and nucleotide/amino acid composition, the methods used in this paper were able to function well. Attempts to process a much larger genome (*Schistosoma mansoni*) exceeded the capacity of these tools. A larger virtual machine would be required, and some of the scripts re-written from *bash* to *Perl* or *python.*

## Preparing peptide sequences.

To avoid missing any relevant genes, the author searched for peptide sequences to find as many putative peptides as possible by using **glimmer3.02**, which wrote out separate files by chromosome or plastid (mitochondrion or apicoplast) named according to the scheme *chromosome/plastid>.genes*. Each line in these files corresponds to one peptide; we identified 12,480 peptides in this way (See Gene count.)

Using EMBL tools, each gene was extracted and translated into an ORF (Open Reading Frame) file to a directory corresponding to the *genes* file.

These processes took about an hour for *p. falciparum*.

## BLASTP scoring

Applying BLASTP to each ORF created a BLAST report file containing up to 10 alignments to targets and including their scores. The reports were named *<ORF>.blastp.txt*. This process took about an hour. (See Figure 9:Understanding BLAST statistics.)

Each blast report was parsed into a file named *<ORF>.blastp.stats*. Parsing took less than a minute.

The ETL process took about 15 minutes, and created 55,848 *blast\_statistics* records in the PostgreSQL database on the work VM. These records were exported and re-imported into the *chembl\_20* database for integration with target and drug information for analysis.

## Testing the hypothesis: Sequences with similarity to existing targets that transcend normality are likely to be targets themselves.

The author attempted to separate the population of cross-species targets from the general population of proteins in *p. falciparum* using the signature of normal distribution of data to separate these two groups. The following concepts support this approach:

1. Protein targets for infectious diseases have a functional role that is vital to the pathogenic organism. Drugs interfere with these targets.
2. Sequences that are functionally important are conserved, as changes to these sequences interfere with their function, preventing their propagation.
3. Over time, gene sequences mutate randomly. Sequences that are highly similar are either closely related, or functionally similar.

These concepts lead us to expect that there should be separate populations of proteins having different distributions of scores: a lower scoring set whose distribution would be explained by random mutation, and a higher scoring set consistent with functional conservation. Principle #2 (conservation of vital function) above tends to create a population that tends to retard changes that would cause a population to become randomly ordered and dissimilar, while principle #3 tends to cause populations to assume more randomly distributed populations that become less and less similar.

### Data analysis using R for *p. falciparum*.

Using R, the *blast\_statistics* table was imported from the *chembl\_20* database where we had created it (See” Loading *blast\_statistics* dataframe.”) using the ETL procedure described (Figure 8: Genome target analysis workflow.) Histograms show that the *score* data are log normally distributed (See Organism Log Score Histograms.)

R’s **qqnorm** plot describes visually how well a distribution approaches normality. The graph produced by the **qqnorm** plot combines with a red line drawn by **qqline** delineating expected normality for comparison.

The complete dataset of all malaria similarity scores departs markedly from normality in the upper range of scores (Figure 13: "All malaria scores" qqnorm plot.)

We can construct an upper bound on scores that discriminates the normally distributed scores from those that are not.

BLASTP is configured to omit low scores in the reports, which means that *mean* and *standard deviation* are not the best measures of centrality and dispersion. *Median* and *Median Absolute Deviation (MAD)* provide better metrics of the behavior of our dataset. We chose a *discriminant* *threshold* as *median(score) + 2 \* mad(score)*. Below this threshold score, we expect the data to be distributed normally. The value of this threshold score for *p. falciparum* is 36.89212.

Plotting the dataset obtained by excluding values greater than this threshold displays normal behavior, as expected (Figure 14: p.falciparum normal scores.)

The remainder of statistics greater than the significance threshold belong to sequences that are more than normally similar to targets sequences.

### Selecting promising sequences as targets using PostgreSQL in the augmented *chembl\_20* database.

Besides the similarity criteria, we also need exclusion criteria for targets so that we do not choose targets that are too similar to human or model organisms that we wish to protect. We also wish to exclude targets that are already identified as malaria targets so that we will show only novel results.

The author created the *exclude\_organisms* table to designate those organisms whose targets we wish to avoid (Table 4: Exclude\_organisms.)

To demonstrate the usefulness of the selection criteria, mapped targets meeting our criteria to known drugs (Figure 17: From statistics to drugs.) Results show 53 hits ( Table 5: Drugs and targets found for Malaria showing cross species organism and mechanism.) Some drugs were duplicated; for example, some drugs were reported for more than one organism, and some drugs were similar but slightly different formulations.

The drugs clindamycin, chloramphenicol and tetracycline mentioned by Ankit Gupta, et al were identified. Neomycin, which was identified, is the same as thiostrepton, was also identified. Other antibiotics not mentioned in their paper were also identified. (See Table 5: Drugs and targets found for Malaria showing cross species organism and mechanism.)

The Query for new malaria targets returns 592 records, which are available for further study. Many of these targets match to more than one genome location.

## Genomes of five other Neglected Disease pathogens were examined using the same pipeline and analysis methods.

The distribution signatures of the data for each of these organisms show that they are eligible for the same analytic treatment as for *p. falciparum*. (See Figure 15: Testing normality for all organisms.)

The discriminant threshold values for each organism defines the distance from the target sequences, which are over-conserved. Scores below this threshold value are less conserved, and have a distribution corresponding to the evolutionary distance from the targets. This value is the distance from a “Center of Mass” like value of an organism from the “Center of Mass” of the targets. Computed values for each organism are stored in table tax\_norm\_threshold (See **tax\_norm\_threshold.sql**.) and contents (See

Table 3: tax\_norm\_threshold values.) These values can be displayed as a dendrogram of the distances for each organism (See Figure 15: Testing normality for all organisms.)

### Trypanosoma Brucei

This organism causes *Human African Trypanosomiasis,* also known as *Sleeping Sickness.*

The Genome of this kinetoplastid organism was downloaded from EnsemblProtists from this URL:

<ftp://ftp.ensemblgenomes.org/pub/protists/release-34/fasta/trypanosoma_brucei/dna/>

Results:

Number of ORFs: 45,179

Number of Drugs Found: 53

(See Table 6: Trypanosoma brucei drugs.)

(See Trypanosoma brucei drugs query.)

Number of targets found: 555

(See Trypanosoma brucei targets query and Table 7: Trypanosoma brucei targets.)

### Trypanosoma Cruzi

This organism causes *Chagas Disease*.

The genome was downloaded from EnsemblProtists at this URL:

<http://protists.ensembl.org/Trypanosoma_cruzi_dm28c/Info/Index>

Results:

Number of ORFs: 1344

Number of Drugs Found: 0

(See Trypanosoma cruzi drugs query.)

The genome sequence of this organism is incomplete, and has not been assorted to individual chromosomes. Only 3% of the number of ORFs found in Trypanosoma brucei were found for Trypanosoma cruzi. Nevertheless, the discriminant threshold computed for Trypanosoma cruzi is similar to that of Trypanosoma brucei to 4 decimal places (See

Table 3: tax\_norm\_threshold values.) We would expect a similar number of genes to be found for *Trypanosoma cruzi,* and because of the similarity of these organisms, we would expect them to share many targets, as targets are highly conserved proteins.

Number of targets found: 16 – this is about 3% of the number of targets found for *Trypanosoma brucei.*

(See Trypanosoma cruzi targets query, Trypanosoma cruzi targets.)

Of 16 targets, only 2 Trypanosoma cruzi targets were not found in Trypanosoma brucei. A linear projection would lead us to expect that 7/8 of drugs and targets for Trypanosoma brucei may also be effective for Trypanosoma cruzi.

(See Queries to find which Trypanosoma cruzi targets are not Trypanosoma brucei targets.)

### Leishmania Major

This organism causes *Leishmaniosis.*

The genome was downloaded from EnsemblProtists at this URL:

<ftp://ftp.ensemblgenomes.org/pub/protists/release-34/fasta/leishmania_major/dna/>

Results:

Number of ORFs: 52,201

Number of Drugs Found: 53

(See

Table 6: Trypanosoma brucei drugs.

|  |  |  |
| --- | --- | --- |
| **pref\_name** | **chembl\_id** | **mechanism\_of\_action** |
| ALBENDAZOLE | CHEMBL1483 | Tubulin inhibitor |
| AMIKACIN SULFATE | CHEMBL1164318 | Bacterial 70S ribosome inhibitor |
| CAPREOMYCIN SULFATE | CHEMBL2218913 | 70S ribosome inhibitor |
| CHLORAMPHENICOL | CHEMBL130 | Bacterial 70S ribosome inhibitor |
| CHLORAMPHENICOL PALMITATE | CHEMBL1506 | Bacterial 70S ribosome inhibitor |
| CHLORAMPHENICOL SODIUM SUCCINATE | CHEMBL1200729 | Bacterial 70S ribosome inhibitor |
| CHLORTETRACYCLINE HYDROCHLORIDE | CHEMBL2146063 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN HYDROCHLORIDE | CHEMBL1200588 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN PALMITATE HYDROCHLORIDE | CHEMBL1237086 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN PHOSPHATE | CHEMBL3184512 | Bacterial 70S ribosome inhibitor |
| DALFOPRISTIN | CHEMBL1200937 | Bacterial 70S ribosome inhibitor |
| DEMECLOCYCLINE HYDROCHLORIDE | CHEMBL1200474 | Bacterial 70S ribosome inhibitor |
| DIRITHROMYCIN | CHEMBL3039471 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN | CHEMBL532 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN ESTOLATE | CHEMBL2218877 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN ETHYLSUCCINATE | CHEMBL1200688 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN GLUCEPTATE | CHEMBL1200657 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN LACTOBIONATE | CHEMBL1200506 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN STEARATE | CHEMBL1200510 | Bacterial 70S ribosome inhibitor |
| GENTAMICIN SULFATE | CHEMBL3039594 | Bacterial 70S ribosome inhibitor |
| GRISEOFULVIN | CHEMBL562 | Tubulin inhibitor |
| KANAMYCIN SULFATE | CHEMBL1446 | Bacterial 70S ribosome inhibitor |
| LINCOMYCIN HYDROCHLORIDE | CHEMBL1201097 | 70S ribosome inhibitor |
| LINEZOLID | CHEMBL126 | 70S ribosome inhibitor |
| MEBENDAZOLE | CHEMBL685 | Tubulin inhibitor |
| MECLOCYCLINE SULFOSALICYLATE | CHEMBL261772 | Bacterial 70S ribosome inhibitor |
| METHACYCLINE HYDROCHLORIDE | CHEMBL2146123 | Bacterial 70S ribosome inhibitor |
| MINOCYCLINE HYDROCHLORIDE | CHEMBL1200881 | Bacterial 70S ribosome inhibitor |
| NEOMYCIN SULFATE | CHEMBL1275977 | Bacterial 70S ribosome inhibitor |
| NETILMICIN SULFATE | CHEMBL1200872 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE | CHEMBL1517 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE CALCIUM | CHEMBL2068727 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE HYDROCHLORIDE | CHEMBL1607480 | Bacterial 70S ribosome inhibitor |
| PAROMOMYCIN SULFATE | CHEMBL2206196 | Bacterial 70S ribosome inhibitor |
| PYRAZINAMIDE | CHEMBL614 | 70S ribosome inhibitor |
| QUINUPRISTIN | CHEMBL1200649 | Bacterial 70S ribosome inhibitor |
| RETAPAMULIN | CHEMBL409542 | 70S ribosome inhibitor |
| RIFABUTIN | CHEMBL444633 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAMPICIN | CHEMBL374478 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAPENTINE | CHEMBL1660 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAXIMIN | CHEMBL1617 | Bacterial DNA-directed RNA polymerase inhibitor |
| SPECTINOMYCIN HYDROCHLORIDE | CHEMBL1200407 | Bacterial 70S ribosome inhibitor |
| STREPTOMYCIN SULFATE | CHEMBL453087 | Bacterial 70S ribosome inhibitor |
| TELITHROMYCIN | CHEMBL1136 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE | CHEMBL1440 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE HYDROCHLORIDE | CHEMBL454950 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE PHOSPHATE COMPLEX | CHEMBL1201071 | Bacterial 70S ribosome inhibitor |
| THIABENDAZOLE | CHEMBL625 | Tubulin inhibitor |
| TIGECYCLINE | CHEMBL376140 | Bacterial 70S ribosome inhibitor |
| TOBRAMYCIN | CHEMBL1747 | 70S ribosome inhibitor |
| TOBRAMYCIN SULFATE | CHEMBL1200780 | 70S ribosome inhibitor |
| TROLEANDOMYCIN | CHEMBL564085 | 70S ribosome inhibitor |
| VIOMYCIN SULFATE | CHEMBL1200526 | 70S ribosome inhibitor |

#### Trypanosoma brucei drugs query.

select distinct md.pref\_name, md.chembl\_id, mechanism\_of\_action

FROM blast\_statistics bs

, tax\_norm\_threshold t

, target\_components tc

, target\_dictionary td

, drug\_mechanism dm

, molecule\_dictionary md

, binding\_sites bind

WHERE bs.score > t.threshold

and bs.tax\_id = t.tax\_id

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and bs.tax\_id = 5691

and bs.tax\_id = t.tax\_id

and tc.tid = td.tid

and bind.tid = tc.tid

and dm.site\_id = bind.site\_id

and md.molregno = dm.molregno

and not exists (select 1

from exclude\_organisms eo

where eo.tax\_id = td.tax\_id

)

order by md.pref\_name;

## Trypanosoma brucei targets query

select max(bs.score) score, chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

from

(select distinct td.chembl\_id, td.target\_type, td.organism, td.tax\_id, bs.targ\_comp

from blast\_statistics bs

, target\_components tc

, target\_dictionary td

, tax\_norm\_threshold tnt

where bs.tax\_id=5691

and bs.tax\_id = tnt.tax\_id

and bs.score > tnt.threshold

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and tc.tid = td.tid

and not exists (select 1

from exclude\_organisms xo

where td.tax\_id = xo.tax\_id

)

) targets

,blast\_statistics bs

where bs.targ\_comp = targets.targ\_comp

group by chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

order by score desc;

Table 7: Trypanosoma brucei targets



## Trypanosoma cruzi drugs query.

select distinct md.pref\_name, md.chembl\_id, mechanism\_of\_action

FROM blast\_statistics bs

, tax\_norm\_threshold t

, target\_components tc

, target\_dictionary td

, drug\_mechanism dm

, molecule\_dictionary md

, binding\_sites bind

WHERE bs.score > t.threshold

and bs.tax\_id = t.tax\_id

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and bs.tax\_id = 5693

and bs.tax\_id = t.tax\_id

and tc.tid = td.tid

and bind.tid = tc.tid

and dm.site\_id = bind.site\_id

and md.molregno = dm.molregno

and not exists (select 1

from exclude\_organisms eo

where eo.tax\_id = td.tax\_id

)

order by md.pref\_name;

### Trypanosoma cruzi targets query

select max(bs.score) score, chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

from

(select distinct td.chembl\_id, td.target\_type, td.organism, td.tax\_id, bs.targ\_comp

from blast\_statistics bs

, target\_components tc

, target\_dictionary td

, tax\_norm\_threshold tnt

where bs.tax\_id=5693

and bs.tax\_id = tnt.tax\_id

and bs.score > tnt.threshold

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and tc.tid = td.tid

and not exists (select 1

from exclude\_organisms xo

where td.tax\_id = xo.tax\_id

)

) targets

,blast\_statistics bs

where bs.targ\_comp = targets.targ\_comp

group by chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

order by score desc;

Table 8: Trypanosoma cruzi targets



Table 9: Leishmania drugs.)

(See Leishmania drugs query.)

Number of targets found: 648

(See Leishmania major targets query, Table 10:Leishmania major targets.)

### Chlamidia Trachomatis

This organism causes a number of diseases, including *Chlamidia* and *Trachoma*.

The genome was downloaded from this URL:

<ftp://ftp.ensemblgenomes.org/pub/bacteria/release-34/fasta/bacteria_0_collection/chlamydia_trachomatis_d_uw_3_cx/dna/Chlamydia_trachomatis_d_uw_3_cx.ASM872v1.dna.chromosome.Chromosome.fa.gz>

Results:

Number of ORFs: 966

Number of Drugs: 49

(See Table 11: Chlamidia drugs.)

(See Chlamidia drugs query.)

Number of targets: 636

(See Chlamidia targets query, Table 12: Chlamidia trochamatis targets)

### Toxoplasmosis Gondii

This organism is the agent of the disease Toxoplasmosis.

The genome was downloaded from here:

<ftp://ftp.ensemblgenomes.org/pub/protists/release-37/fasta/toxoplasma_gondii/dna/>

Number of ORFs: 150,062

Number of Drugs: 49

(see **Error! Reference source not found.**.)

Number of targets: 606

(See **Error! Reference source not found.**.)

# Conclusions

* This paper validates a method for discovering cross species targets by identifying 29 distinct drugs for malaria (53, counting different formulations) and 592 targets.
* Drugs and targets were also found for five other neglected disease organisms: Trypanosoma brucei, Trypanosoma cruzi, Leishmania major, Chlamidia trachomatis, and Toxoplasmosis Gondii.
* BLASTP provides a means of measuring the similarity of proteins of a disease organism and a database of target sequences. Removing populations of normally similar proteins reveals a highly similar population of highly conserved proteins that may be targets.
* The author describes a method to compute the **discriminant threshold** criterion for a given genome which is used to separate sequences that are target candidates from those that are not.
* As more organisms are added to the database, a taxonomy emerges from the *discriminant* *thresholds* that serve as a distance metric between pathogen genomes and target sequences.
* The *discriminant threshold* may help identify similarities between organisms that may suggest interpolations of missing genomic data for organisms for which complete genomic data is not available. In this investigation, only 3% of the number of ORFs found in *Trypanosoma brucei* were found for *Trypanosoma cruzi.* Nevertheless, the *discriminant threshold* computed for *Trypanosoma cruzi* is similar to that of *Trypanosoma brucei* to 4 decimal places, suggesting that missing information for *Trypanosoma cruzi* might be predicted from genome information from the more completely sequenced species *Trypanosoma brucei*.
* This platform provides a way to choose candidate drugs without knowing the identity of the pathogen if the pathogen’s genome can be obtained.
* This platform provides methods for ranking preference for a target by similarity, by mechanism, or by other molecular criteria provided by the ChEMBL database.

Blast statistics values can be imported into MyChEMBL database from supporting data provided here to explore useful annotations for targets and drugs.

# Acknowledgements

Many thanks to Nikolaus Obholzer and Sylvain Meylan for the critical reading of the manuscript.

Thanks to Eric Fischbach for formatting help.

# References

1. Fact Sheet: World Malaria Report 2015, p. 1.

World Health Organization

9 December 2015

<http://www.who.int/malaria/media/world-malaria-report-2015/en/>

1. myChEMBL: A.P. Bento, A. Gaulton, A. Hersey, L.J. Bellis, J. Chambers, M. Davies, F.A. Krüger, Y. Light, L. Mak, S. McGlinchey, M. Nowotka, G. Papadatos, R. Santos & J.P. Overington (2014) 'The ChEMBL bioactivity database: an update' Nucl. Acids Res. Database Issue. 42 D1083-D1090 DOI:10.1093/nar/gkt1031 PMID:242149652.
2. M. Davies, M. Nowotka, G. Papadatos, F. Atkinson, G.J.P. van Westen, N Dedman, R. Ochoa and J.P. Overington (2014) 'myChEMBL: A Virtual Platform for Distributing Cheminformatics Tools and Open Data' Challenges 5 (334-337) [pdf](http://doi.org/10.1093/bioinformatics/btt666)
3. Reduced ribosomes of the apicoplast and mitochondrion of *Plasmoidium* spp. And predicted interactions with anitbiotics

Gupta, A, Shah P, Haider A, Gupta K, Siddiqi MI, Rlph SA, Habib S. 2014

*Open BIol. 4: 140045.* <http://dx/doi.org/10.1098/rsob.140045>

1. Ankit Gupta, et al. “Reduced ribosomes of the apicoplast and mitochondrion of Plasmodium spp. and predicted interactions with antibiotics”, Figure 1 and quotation.

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4042851/figure/RSOB140045F1/>

© 2014 The Authors. Published by the Royal Society under the terms of the Creative Commons Attribution License<http://creativecommons.org/licenses/by/3.0/>, which permits unrestricted use, provided the original author and source are credited.

1. An Ontology for Pharmaceutical Ligands and Its Application for in Silico Screening and Library Design

Ansgar Schuffenhauer, Jurg Zimmermann, Ruedi Stoop, Jan-Jan van der Vver, Steffano Lecchini, and Edgar Jacoby

J. Chem Inf. Comput. Sci 2002, 42, 947-955

Received November 9, 2001

1. Sheridan, R.P. & Kearsley, S.K. “Why do we need so many chemical similarity search methods?

*Drug Discov. Today* **7**, 903–911 (2002)

1. Bajusz, Dávid, Anita Rácz, and Károly Héberger. “Why Is Tanimoto Index an Appropriate Choice for Fingerprint-Based Similarity Calculations?” *Journal of Cheminformatics* 7 (2015): 20. *PMC*. Web. 4 Nov. 2017.

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4456712/>

[Copyright](https://www.ncbi.nlm.nih.gov/pmc/about/copyright/) © Bajusz et al. 2015

1. Calculating Similarities Between Biological Activities in the MDL Drug Data Report Database

Rober P. Sheridan and Joseph Shpungin

J. Chem. Inf. Comput. Sci. 2005,44, 727-740

1. Novel similarity-based clustering algorithm for grouping broadcast news

Oktay Ibrahimov, Ishwar Sethi, and Nevenka Dimitrova

Publication date: 03/2002

Proc. SPIE 4730, Data Mining and Knowledge Discovery: Theory, Tools, and Technology IV, (12 March 2002); doi: 10.1117/12.460239; <http://dx.doi.org/10.1117/12.460239>

<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=EBA0799D3A740AB30145778CE6C824C1?doi=10.1.1.88.6622&rep=rep1&type=pdf>

1. Relating protein pharmacology by ligand chemistry

Keiser MJ, Roth BL, Armbruster BN, Ernsberger P, Irwin JJ, Shoichet BK

Published: 2007 Nature Publishing Group; Nat Biotech 25 (2), 197-206 (2007).

1. Kruskal, J. On the shortest spanning subtree and the traveling salesman problem.

Proc. Am. Math. Soc. 7, 48-50 (1956).

1. Stephen F. Altschul, Thomas L. Madden, Alejandro A.Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J.Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.
2. A Mathematical Theory of Communication

C.E. Shannon

[The Bell System Technical Journal](http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=6731005) ( Volume: 27, [Issue: 3](http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=6773023), July 1948 )

1. Han, Te Sun & Kobayashi, Kingo (2002). [*Mathematics of Information and Coding*](https://books.google.com/books?id=VpRESN24Zj0C&pg=PA19). American Mathematical Society. pp. 19–20. [*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*978-0-8218-4256-0*](https://en.wikipedia.org/wiki/Special:BookSources/978-0-8218-4256-0).
2. The Mean and Median Absolute Deviations

T. Pham-Gia, T.L.Hung

[Volume 34, Issues 7–8](https://www.sciencedirect.com/science/journal/08957177/34/7), October 2001, Pages 921-936

1. Virtualbox Wiki Downloads: <https://www.virtualbox.org/wiki/Downloads>

The 14 chromosomes of *p. falciparum* were downloaded from this URL: <ftp://ftp.ensemblgenomes.org/pub/protists/release-32/fasta/plasmodium_falciparum/dna/>

Mitochondrial dna sequence was downloaded from here: GenBank: AJ276844.1 <https://www.ncbi.nlm.nih.gov/nuccore/8346980?report=fasta>

Apicoplast sequence was downloaded from here: Plasmodium falciparum genome assembly, organelle: plastid:apicoplast

NCBI Reference Sequence: NC\_030754.1

<https://www.ncbi.nlm.nih.gov/nuccore/1052489052?report=fasta>

1. Microbial gene identification using interpolated Markov models

S. Salzberg, A. Delcher, S. Kasif, and O. White

Nucl. Acids Res., 26(2):544–548, 1998

1. Plasmodium structure. <https://upload.wikimedia.org/wikipedia/commons/f/fa/Plasmodium.png>
2. Drug repurposing and human parasitic protozoan diseases

Katherine T. Andrews, Gillian Fisher, Tina S. Skinner-Adams

2014 Published by Elsevier Ltd. on behalf of Australian Society for Parasitology Inc. This is an open

access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

1. Trypanosoma brucei genome. <ftp://ftp.ensemblgenomes.org/pub/protists/release-34/fasta/trypanosoma_brucei/dna/>

[The genome of the African trypanosome Trypanosoma brucei](http://europepmc.org/abstract/MED/16020726).  
Berriman M, Ghedin E, Hertz-Fowler C, Blandin G, Renauld H, Bartholomeu DC, Lennard NJ, Caler E, Hamlin NE, Haas B et al. 2005. Science. 309:416-422

1. Trypanosoma cruzi genome:

<http://protists.ensembl.org/Trypanosoma_cruzi_dm28c/Info/Index>

[Trypanosoma cruzi Clone Dm28c Draft Genome Sequence](http://europepmc.org/abstract/MED/24482508).  
Grisard EC, Teixeira SM, de Almeida LG, Stoco PH, Gerber AL, Talavera-Lpez C, Lima OC, Andersson B, de Vasconcelos AT. 2014. Genome Announc. 2

1. Leishmania major genome:

<ftp://ftp.ensemblgenomes.org/pub/protists/release-34/fasta/leishmania_major/dna/>

[The genome of the kinetoplastid parasite, Leishmania major](http://europepmc.org/abstract/MED/16020728).  
Ivens AC, Peacock CS, Worthey EA, Murphy L, Aggarwal G, Berriman M, Sisk E, Rajandream MA, Adlem E, Aert R et al. 2005. Science. 309:436-442.

1. Chlamidia trachomatis genome:

<ftp://ftp.ensemblgenomes.org/pub/bacteria/release-34/fasta/bacteria_0_collection/chlamydia_trachomatis_d_uw_3_cx/dna/Chlamydia_trachomatis_d_uw_3_cx.ASM872v1.dna.chromosome.Chromosome.fa.gz>

Genome sequence of an obligate intracellular pathogen of humans: Chlamydia trachomatis. Stephens R.S., Kalman S., Lammel C., Fan J., Marathe R., Aravind L., Mitchell W., Olinger L., Tatusov R.L., Zhao Q., Koonin E.V., Davis R.W. - *Science* 1998, **282**(5389):754-9 PubMed: [9784136](http://europepmc.org/abstract/MED/9784136)

# Supporting Information

## Figures

Figure : Nucleotide Frequencies Across Species

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **malaria** | **toxoplasmosis** | **chlamidia** | **brucei** | **cruzi** | **leishmania** |
| **A** | 40.31% | 23.86% | 29.42% | 26.90% | 24.85% | 19.97% |
| **T** | 40.32% | 23.86% | 29.27% | 26.86% | 24.60% | 20.31% |
| **C** | 9.66% | 26.16% | 20.65% | 23.20% | 25.57% | 29.99% |
| **G** | 9.70% | 26.12% | 20.66% | 23.03% | 24.99% | 29.73% |

Table : Nucleotide frequencies by species.

Figure : Amino Acid Frequencies by species

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| residue | malaria | toxoplasmosis | chlamidia | brucei | cruzi | leishmania |
| A | 1.84% | 7.76% | 7.50% | 7.06% | 6.33% | 10.67% |
| C | 1.82% | 3.56% | 1.61% | 2.96% | 3.55% | 3.36% |
| D | 6.26% | 3.26% | 4.54% | 4.00% | 2.53% | 3.78% |
| E | 6.72% | 5.31% | 6.63% | 5.64% | 3.63% | 4.76% |
| F | 4.79% | 5.05% | 4.87% | 5.61% | 5.37% | 3.79% |
| G | 2.73% | 5.63% | 6.34% | 5.75% | 5.15% | 6.03% |
| H | 2.44% | 2.81% | 2.29% | 2.84% | 4.04% | 3.43% |
| I | 9.72% | 3.13% | 6.60% | 4.73% | 4.76% | 2.77% |
| K | 11.67% | 3.94% | 5.78% | 4.92% | 6.30% | 2.93% |
| L | 7.63% | 9.98% | 11.24% | 10.28% | 8.82% | 9.91% |
| M | 2.34% | 1.70% | 1.98% | 2.26% | 1.69% | 1.97% |
| N | 14.23% | 2.38% | 3.50% | 3.63% | 3.89% | 2.15% |
| P | 1.91% | 5.86% | 4.38% | 4.88% | 6.49% | 6.81% |
| Q | 2.69% | 3.54% | 4.19% | 3.42% | 3.97% | 3.42% |
| R | 2.55% | 9.43% | 4.83% | 6.50% | 8.58% | 8.05% |
| S | 6.26% | 11.37% | 8.13% | 8.25% | 9.56% | 9.47% |
| T | 4.00% | 5.47% | 5.12% | 5.72% | 6.20% | 6.08% |
| V | 3.74% | 6.54% | 6.43% | 7.40% | 5.56% | 7.34% |
| W | 0.48% | 1.13% | 0.96% | 1.12% | 1.43% | 1.12% |
| Y | 6.17% | 2.16% | 3.07% | 3.04% | 2.15% | 2.17% |

Table : Amino Acid percentages, by species.

Figure : Genome size comparison by species.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| malaria | chlamidia | leishmania | brucei | cruzi | toxoplasmosis |
| 23,263,391 | 1,042,519 | 32,855,082 | 25,785,970 | 27,304,309 | 62,966,896 |

An external file that holds a picture, illustration, etc.
Object name is rsob-4-140045-g1.jpg

Figure A five-set Venn diagram showing the distribution of nuclear- or plastid-encoded ribosomal proteins that would constitute the plastid ribosomes of apicomplexans P. falciparum and T. gondii, red alga C. merolae, green alga C. reinhardtii and diatom T. pseudonana.

From “Reduced ribosomes of the apicoplast and mitochondrion of Plasmodium spp. and predicted interactions with antibiotics” , Gupta et al.



Figure :Hardware Configuration.



Figure : VM configuration parameters.



Figure :MyCHeMBL landing page.



Figure : Genome target analysis workflow

The analysis workflow uses genome information downloaded from NCBI.gov and from EMBL to a *work* virtual machine running on a PC. Target and molecule information resides in a PostgreSQL database named *chembl\_20* that resides in a different MyChEMBL virtual machine running on the same PC, which acts as the integration platform.

The *p .falciparum* genome is decompressed as it is downloaded to the *work* VM. It consists of files with filetype *.fa* containing FASTA formatted sequences of nucleotides for each chromosome. The mitochondrial and apicoplast genome files were downloaded from <https://www.ncbi.nlm.nih.gov/nuccore/8346980?report=fasta> for mitochondrial genome, <https://www.ncbi.nlm.nih.gov/nuccore/1052489052?report=fasta>

for apicoplast genome, and chromosomal genome files were downloaded from <ftp://ftp.ensemblgenomes.org/pub/protists/release-32/fasta/plasmodium_falciparum/dna/>*.*

The workflow continues as glimmer 3.02 identifies the coordinates of putative genes. Extract stitches together these together into gene files for each chromosome. A Perl script (See chrom\_genes\_to\_proteins.pl Perl Script) parses gene files, creating Open Reading Frame files for each peptide by invoking **transeq**. These ORF files are amino acid .FASTA files that encode polypeptides.

The split\_to\_fasta.pl Perl script converts the exported target sequences from the **chembl\_20** PostgreSQL database to a concatenated .FASTA file, which is converted to a blast database by the **makeblastdb** command.

BLASTP queries the *target blast database* using each ORF (Open Reading Frame) file creating a blast report file for each. The **extract\_blast\_stats.pl**Perl script extracts a blast stats file for each ORF file. The **create\_populate\_blast\_statistics.pl** Perl script creates the **populate\_blast\_statistics.sql** file which loads the blast stats files into the *blast\_statistics*table, for analytical use.



Figure :Understanding BLAST statistics.

Each match and alignment contains various scores which BLASTP computed for it.

The “Expect” statistic measures the probability that the matches could occur by chance. For example: if we assume equal probability for the occurrence of a particular amino acid in a sequence (.05), this is the expected value for a random match of this length sequence. This is much too high an expected value to accept this as a non-random match. Results with an expected value >= .001 were filtered out.

The ratio of exact matches to the length of the *subject* gives the *identities* statistics; The ratio of conserved matches (including identities and functionally similar amino acids) to the total number of characters in the *subject* gives the *positives* statistics. In the same way, gaps are tabulated and compared with the total subject length. The score statistic gives BLASTP’s estimate of the query and target sequences similarity by accumulating the positive matches weighted by the match length minus gap penalties.

We use score in this paper to rank similarity of proteins. It is computed by adding successive matching scores using the BLOSUM62 scoring matrix to rate the “goodness” of a particular match given log odds of the frequencies of the given amino acids for the particular “blocks” database built with sequences having no more than 62% in common. Experiments have shown that this matrix is one of the best for detecting even weak protein similarities. The matrix values are calculated from these data by calculating the log base 2 odds of the observed frequencies vs expected, rounded to the nearest unit.  A score of zero indicates that the frequency with which a given two amino acids were found aligned in the database was as expected by chance, while a positive score indicates that the alignment was found more often than by chance, and negative score indicates that the alignment was found less often than by chance. The score is cumulative (computed for each residue in the query string.)

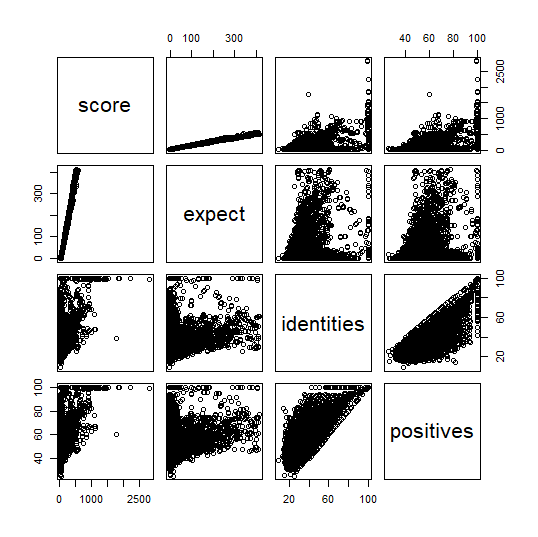


Figure : Pairwise comparison of BLAST statistics.

Plotted BLAST statistics are for malaria (*P. falciparum, tax\_id=5833)*

Plotted *expect* variable is computed as -log(expect); score is always positive, so this graph shows that that this transformation indicates the same thing as *score*, scaled differently. *Expect* refers to probability that similarity is randomly explained, so a very small value indicates more significance. The negative log of expected value indicates significance.

Representation of very small magnitudes in common math packages such as those used by R has a lower bit density than that of whole numbers. For this reason, the *score* statistic has higher resolution than the *expect* statistic for many computations.

*Identities* and *positives* are very similar; the *positives* set includes imprecise sequence matches, where similar amino acids can count for a match. For this reason, the *positives* set has values that are greater than or equal to the *identities* set. Comparison between *scores* and *identitities*  is practically the same as the comparison between  *scores* and  *positives*.

Differences between the comparison of *scores* by *identities* and *expect* by *identities* varies substantially due to the imprecision caused by computations using *expect.*



Figure : Plasmodium structural diagram showing organelles.

Apicoplast and mitochondria have their own genomes which are closely related to those of bacteria.

## Organism Log Score Histograms

Histograms of log BLASTP scores, showing superimposed normal plot in red.

Vertical blue lines show thresholds that are 1, 2, 3, and 4 median absolute deviations from the median.

Normal curve was calculated using **R dnorm** function, which expects a standard deviation parameter.

The **mad** value (denoted by is smaller ()

The maximum value for

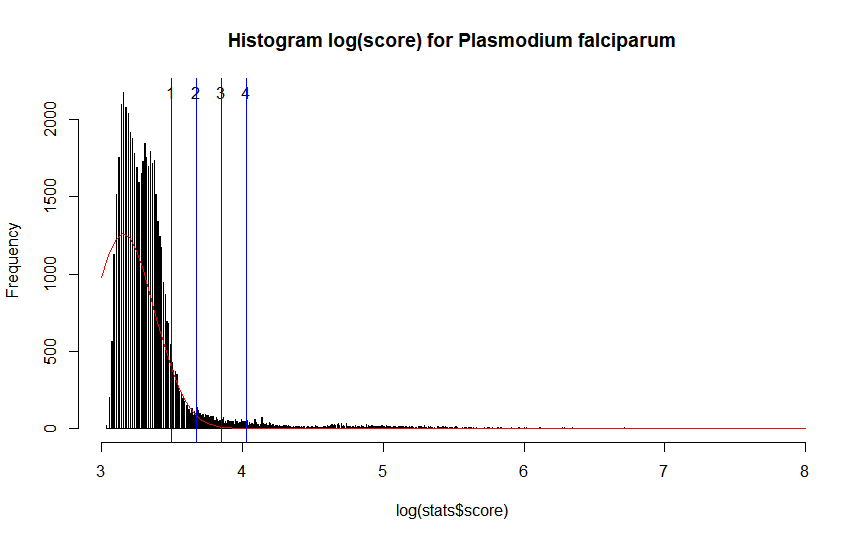


Figure : Plasmodium falciparum score histogram

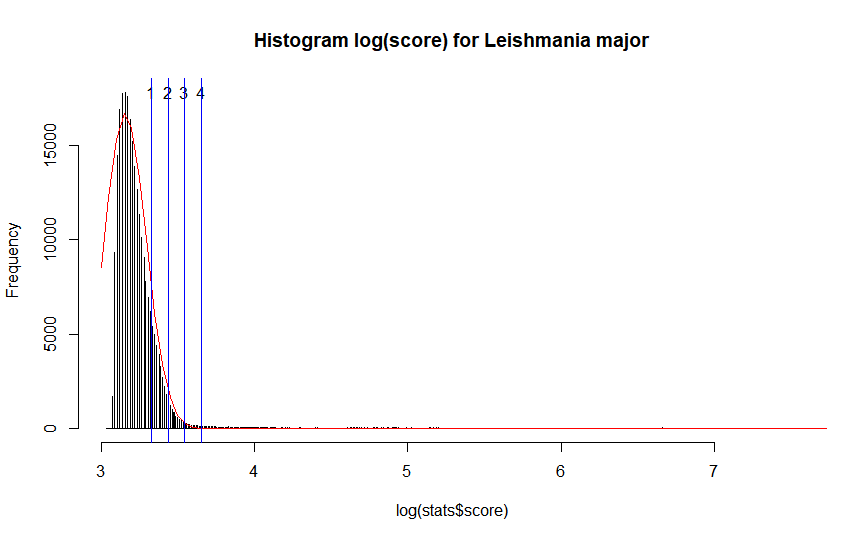


Figure :Leishmania major score histogram

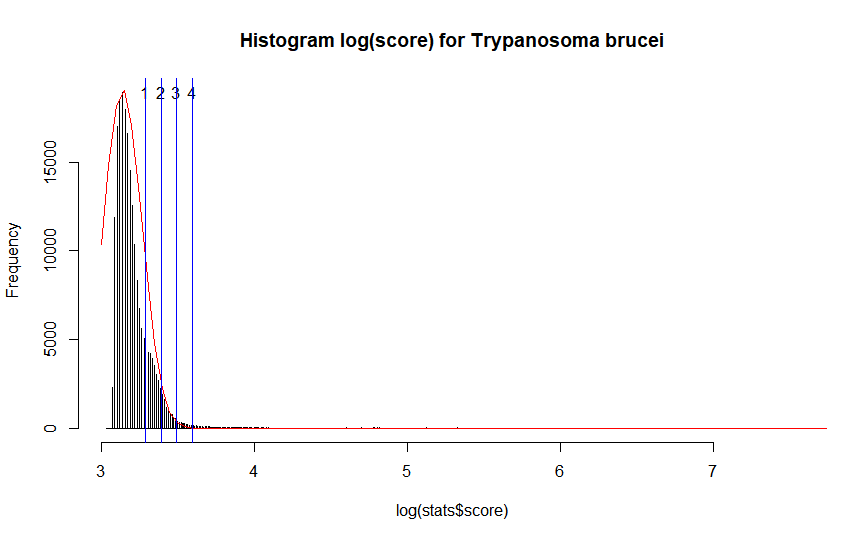


Figure : Trypanosoma brucei score histogram

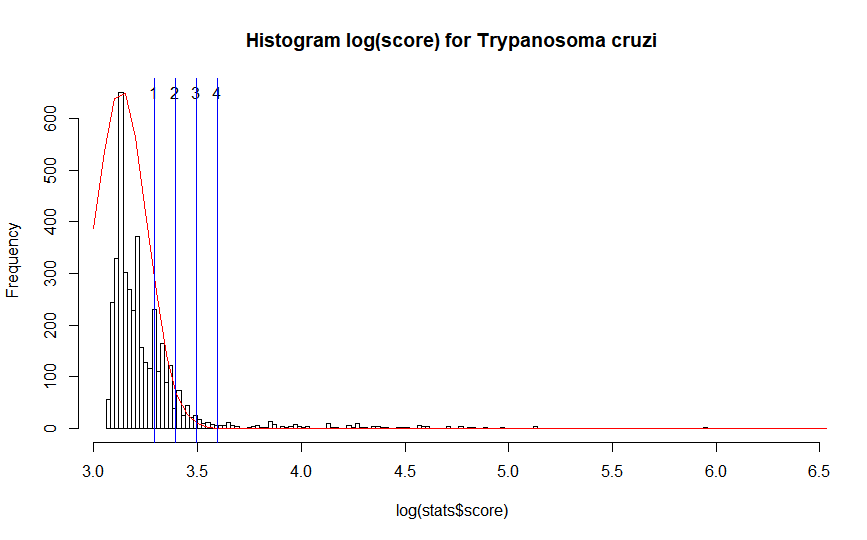


Figure : Trypanosoma cruzi score histogram

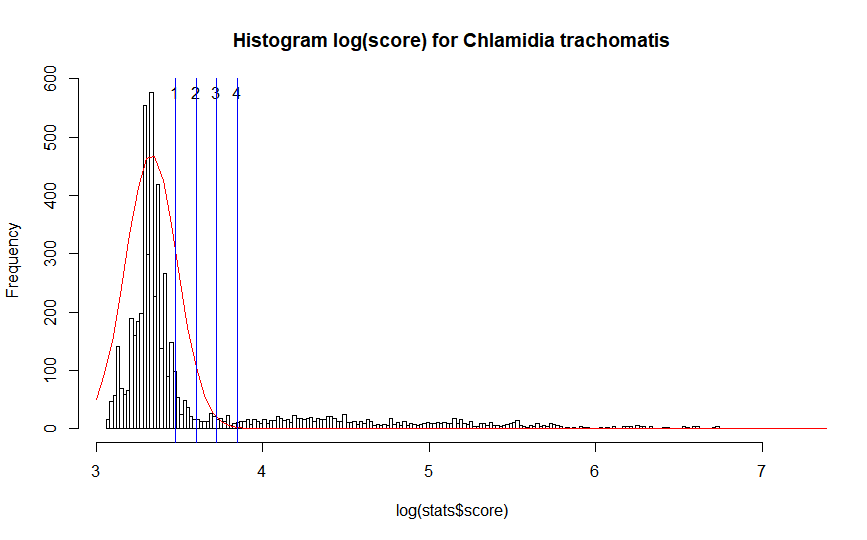


Figure : Chlamidia trachomatis score histogram

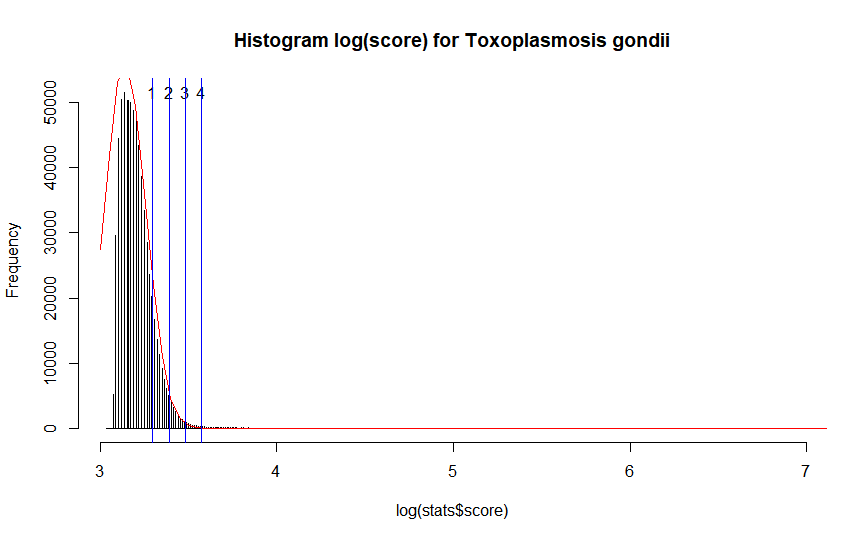


Figure : Toxoplasmosis gondii score histogram

## Q-Q plots



Figure : "All malaria scores" qqnorm plot.

The red line shows expected normality. This plot shows departure from normality at the right (upper) portion of the graph.

These statements produce the graph:

qqnorm(log(malaria\_statistics$score),main='All Malaria Scores')

qqline(log(malaria\_statistics$score),col='red',lwd=3)



Figure : p.falciparum normal scores.

Plot was produced by these R commands:

significance\_threshold<-function(score){

median(score)+2\*mad(score)

}

malaria\_threshold=significance\_threshold(malaria\_statistics$score)

norm\_falciparum=malaria\_statistics[malaria\_statistics$score < malaria\_threshold,]

qqnorm(norm\_falciparum$score,main='p. falciparum normal scores')

qqline(norm\_falciparum$score,col='red',lwd=3)

The red line indicates expected normality.



Figure : Testing normality for all organisms.

The red line in each of these graphs shows where a normal distribution would be predicted to fall.

The top 5 graphs show that the total datasets for each organism are not normal.

The bottom 5 graphs show how the data below the computed discrimination threshold conforms approximately to a normal distribution.



Figure : Cluster Dendrogram

This figures groups organisms by similarity to the target universe. This applies only to the normally scored subset, as defined by the *discriminant threshold.*

The *discriminant threshold* defines a score beneath which similarity belongs to a normal distribution of scores. Scores that belong to the normally distributed subset are expected to be less conserved. For this reason, we attribute their distribution to random mutation, which would correspond to evolutionary time between their sequences and those of the targets, which tend to be conserved.

|  |  |
| --- | --- |
| **tax\_id** | **organism** |
| 9483 | Callithrix jacchus |
| 9541 | Macaca fascicularis |
| 9541 | Macaaca fascicularis |
| 9544 | Macaca mulatta |
| 9606 | Homo sapiens |
| 9615 | Canis lupus famiaris |
| 9615 | Canis lupus familiaris |
| 9615 | Canis familiaris |
| 9823 | Sus scrofa |
| 9913 | Bos taurus |
| 9940 | Ovis aries |
| 9986 | Oryctolagus cuniculus |
| 10090 | Mus musculus |
| 10116 | Rattus norvegicus |

Table : Exclude\_organisms.

Entries for tax\_id 9541 and 9615 show multiple entries due to variations in the spelling of the organism name in the database. These organisms were excluded because of close relationship to humans or close relationship to an experimental model organism.



Figure : From statistics to drugs.

This query produces a list of targets that have drugs for malaria that are not already identified in the database:

select distinct td.tax\_id, td.organism, md.pref\_name, md.chembl\_id, mechanism\_of\_action

FROM blast\_statistics bs

, target\_components tc

, target\_dictionary td

, drug\_mechanism dm

, molecule\_dictionary md

, binding\_sites bind

, tax\_norm\_threshold tnt

WHERE bs.score > tnt.threshold

and bs.tax\_id = tnt.tax\_id

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and bs.tax\_id = 5833

and tc.tid = td.tid

and bind.tid = tc.tid

and dm.site\_id = bind.site\_id

and md.molregno = dm.molregno

and not exists (select 1

from exclude\_organisms eo

where eo.tax\_id = td.tax\_id

)

and upper(td.organism) not like '%PLASMODIUM%'

order by md.pref\_name;

## Installing MyChEMBL VM.

MyChEMBL is a virtual machine platform that is downloadable to a personal computer. It runs in a separate virtual machine running CENTOS (a variety of Redhat Linux) under Oracle’s Virtualbox software. Each MyChEMBL virtual machine contains its own PostgreSQL database and Tomcat webserver. We recommend using a PC with at least 8 gb of memory; we used 12 gb.

To install, we can download one of the releases:

[ftp://ftp.ebi.ac.uk/pub/databases/chembl/VM/myChEMBL/releases](ftp://ftp.ebi.ac.uk/pub/databases/chembl/VM/myChEMBL/releases/)

We are using release 20.

Install Oracle’s Virtualbox:

(See Setting up a Linux Virtual Machine using Oracle’s VirtualBox software.)

Install Vagrant:

<https://www.vagrantup.com/downloads.html>

From a command prompt, run :

**vagrant init chembl/myChEMBL && vagrant up**

wait a bit…

In a browser, go to this url: <http://127.0.0.1:8000/>

Which takes you to the MyChEMBL landing page hosted on your own P.C.

(see Figure 7:MyCHeMBL landing page. )

## Loading *blast\_statistics* dataframe.

library(RPostgreSQL)

drv=dbDriver('PostgreSQL')

con=dbConnect(drv,dbname='chembl\_20',port=5432,host='localhost',user='mychembl')

blast\_statistics=dbReadTable(con,'blast\_statistics')

## Linux user setup.

1. Start VM
2. Login as **adminuser**, password **adminuser** # default for this distro
3. From *Applications*, run *System Tools -> Terminal*
4. **adduser <username>**
5. **passwd <username>**
6. **sudo visudo** Edits the visudo file in VI. Uncomment the line that mentions %%wheel.
7. **usermod –G wheel <username>** # this makes user <username> a sudoer.

## Gene count

These commands were performed on the work VM.

[~/genome/genes] wc -l \*.genes

30 apicoplast.genes

402 chromosome.1.genes

691 chromosome.2.genes

675 chromosome.3.genes

728 chromosome.4.genes

252 chromosome.5.genes

2010 chromosome.6.genes

786 chromosome.7.genes

258 chromosome.8.genes

1031 chromosome.9.genes

329 chromosome.10.genes

1349 chromosome.11.genes

1418 chromosome.12.genes

575 chromosome.13.genes

1940 chromosome.14.genes

6 mt.genes

12480 total

## chrom\_genes\_to\_proteins.pl Perl Script

#!/usr/bin/perl

# chrom\_genes\_to\_proteins.pl

# invoke: chrom\_genes\_to\_proteins.pl <input chrom nucleic acid fasta>

# Input filename looks like <chromosomename.genes>

# Program creates a directory that looks like <chromosomename> and

# creates a separate file named <orfname>.fasta in that directory.

if (@ARGV < 1) {die "Specify gene filename.\n"}

my $gene\_filename = shift @ARGV;

$gene\_filename =~ m/(\S+)\.genes/;

my $dest = $1;

mkdir $dest;

open(CHROMFILE,"$gene\_filename") or die ("Unable to open $gene\_filename.\n");

my @orfs=<CHROMFILE>;

close(CHROMFILE);

foreach my $orf(@orfs)

{

$orf =~ m/(\S+)\s+(\S+)/;

my $orfname = $1;

my $nucleotides = $2;

open(TEMPGENE,">tempgene.fa") or die ("Unable to create tempgene");

print TEMPGENE $nucleotides;

close(TEMPGENE);

system "transeq -sequence tempgene.fa -outseq $dest/$orfname.fasta";

}

## split\_to\_fasta.pl Perl script

#######################################

# split\_to\_fasta.pl

# input recs: <key><delim><sequence>

# output : rec1 = ><key>

# rec2 = <sequence>

#######################################

my $infile = 'component\_sequences.txt';

my $outfile = 'component\_sequences.fa';

my $delim = '\|';

open(IN, $infile) or die("Unable to open $infile\n");

my @lines = <IN>;

close(IN);

open(OUT,">",$outfile) or die ("Unable to open $outfile\n");

foreach my $line(@lines)

{

my @rec = split($delim,$line);

if (scalar(@rec) > 1)

{

print OUT ">$rec[0]\n";

print OUT "$rec[1]\n";

}

}

close(OUT);

exit(0);

## blast\_stats.sh

for i in $(ls -d \*/);do

for j in $(ls ${i}\*.blastp.txt);do

k=`echo $j | cut -d '.' -f 1,2`

perl extract\_header.pl < $j > ${k}.stats

done

done

## extract\_header.pl

use Switch 'fallthough';

my @lines = <STDIN>;

my $phase = 0;

my @rec = {};

my $rec\_string;

my %recs = ();

foreach my $line(@lines)

{

switch($phase){

case 0 {

if ( $line =~ m/\>\s(\S+)/) # orf id

{

$phase = 1;

$rec[0] = $1;

}

}

case 1 {

if ( $line =~ m/Length\=(\S+)/ ){

$rec[scalar(@rec)] = $1;

$phase = 2;

}

}

case 2 {

if ( $line =~ m/Score\s\=\s(\S+)/){

$rec[scalar(@rec)] = $1;

$line =~ m/Expect\s\=\s(\S+),/;

$rec[scalar(@rec)] = $1;

$phase = 3;

}

}

case 3 {

if ( $line =~ m/Identities\s\=\s\S+\s\((\S+)\%/){

$rec[scalar(@rec)] = $1;

$line =~ m/Positives\s\=\s\S+\s\((\S+)\%/;

$rec[scalar(@rec)] = $1;

$line =~ m/Gaps\s\=\s\S+\s\((\S+)\%\)/;

$rec[scalar(@rec)] = $1;

$rec\_string = join("\t",@rec);

$recs{$rec\_string} = 1;

$phase = 0;

@rec = {};

}

}

}

}

foreach my $record(keys %recs){

print "$record\n";

}

## create\_populate\_blast\_statistics.sh

# create populate\_blast\_statistics.sql

echo -- populate\_blast\_statistics.sql > populate\_blast\_statistics.sql

for i in $(ls -d \*/);do

for j in $(ls ${i}\*.stats);do

echo 'truncate table tmp\_blast\_statistics;' >> populate\_blast\_statistics.sql

echo "\\copy tmp\_blast\_statistics (targ\_comp, query\_length, score, expect, identities, positives, gaps) from '${PWD}/${j}'" >> populate\_blast\_statistics.sql

echo "update tmp\_blast\_statistics set tax\_id=5691, chromosome='$i',orf='${j:${#i}}';" >> populate\_blast\_statistics.sql

echo 'insert into blast\_statistics select \* from tmp\_blast\_statistics;' >> populate\_blast\_statistics.sql

done

done

## blast\_statistics.sql

create table blast\_statistics

(

tax\_id bigint,

chromosome character varying(50),

orf character varying(50),

targ\_comp int,

query\_length smallint,

score numeric,

expect numeric,

identities numeric,

positives numeric,

gaps numeric

);

create table tmp\_blast\_statistics

(

tax\_id bigint,

chromosome character varying(50),

orf character varying(50),

targ\_comp int,

query\_length smallint,

score numeric,

expect numeric,

identities numeric,

positives numeric,

gaps numeric

);

## orf\_sequence.sql

CREATE TABLE orf\_sequence

(

tax\_id bigint,

chromosome character varying(50),

orf\_name character varying(50),

orf\_aa text

);

## tax\_norm\_threshold.sql

-- Table: tax\_norm\_threshold

-- DROP TABLE tax\_norm\_threshold;

CREATE TABLE tax\_norm\_threshold

(

tax\_id bigint NOT NULL,

threshold numeric,

organism character varying(150),

CONSTRAINT tax\_norm\_threshold\_pk PRIMARY KEY (tax\_id)

)WITH (

OIDS=FALSE

);

|  |  |  |
| --- | --- | --- |
| **tax\_id** | **threshold** | **organism** |
| **5833** | **36.89212** | **Plasmodium falciparum** |
| **5664** | **30.63388** | **Leishmania major** |
| **5691** | **29.04432** | **Trypanosoma brucei** |
| **5693** | **29.04432** | **Trypanosoma cruzi** |
| **272561** | **35.31996** | **Chlamidia trachomatis** |
| **5811** | **29.0478** | **Toxoplasmosis gondii** |

Table : tax\_norm\_threshold values.

## Query for new malaria targets

select distinct td.chembl\_id,td.tax\_id,organism

FROM blast\_statistics bs

, target\_components tc

, target\_dictionary td

, tax\_norm\_threshold tnt

WHERE bs.score > tnt.threshold

And tnt.tax\_id = bs.tax\_id

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and tc.tid = td.tid

and not exists (select 1

from exclude\_organisms eo

where eo.tax\_id = td.tax\_id

)

and upper(td.organism) not like '%PLASMODIUM%'

and bs.tax\_id = 5833

group by td.chembl\_id, td.tax\_id, organism

order by chembl\_id;



Table : Drugs and targets found for Malaria showing cross species organism and mechanism.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **tax\_id** | **organism** | **pref\_name** | **chembl\_id** | **mechanism\_of\_action** |
| 6231 | Nematoda | ALBENDAZOLE | CHEMBL1483 | Tubulin inhibitor |
| 2 | Bacteria | AMIKACIN SULFATE | CHEMBL1164318 | Bacterial 70S ribosome inhibitor |
| 1773 | Mycobacterium tuberculosis | CAPREOMYCIN SULFATE | CHEMBL2218913 | 70S ribosome inhibitor |
| 2 | Bacteria | CHLORAMPHENICOL | CHEMBL130 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | CHLORAMPHENICOL PALMITATE | CHEMBL1506 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | CHLORAMPHENICOL SODIUM SUCCINATE | CHEMBL1200729 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | CHLORTETRACYCLINE HYDROCHLORIDE | CHEMBL2146063 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | CLINDAMYCIN HYDROCHLORIDE | CHEMBL1200588 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | CLINDAMYCIN PALMITATE HYDROCHLORIDE | CHEMBL1237086 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | CLINDAMYCIN PHOSPHATE | CHEMBL3184512 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | DALFOPRISTIN | CHEMBL1200937 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | DEMECLOCYCLINE HYDROCHLORIDE | CHEMBL1200474 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | DIRITHROMYCIN | CHEMBL3039471 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | ERYTHROMYCIN | CHEMBL532 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | ERYTHROMYCIN ESTOLATE | CHEMBL2218877 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | ERYTHROMYCIN ETHYLSUCCINATE | CHEMBL1200688 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | ERYTHROMYCIN GLUCEPTATE | CHEMBL1200657 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | ERYTHROMYCIN LACTOBIONATE | CHEMBL1200506 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | ERYTHROMYCIN STEARATE | CHEMBL1200510 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | GENTAMICIN SULFATE | CHEMBL3039594 | Bacterial 70S ribosome inhibitor |
| 5550 | Trichophyton | GRISEOFULVIN | CHEMBL562 | Tubulin inhibitor |
| 2 | Bacteria | KANAMYCIN SULFATE | CHEMBL1446 | Bacterial 70S ribosome inhibitor |
| 1313 | Streptococcus pneumoniae | LINCOMYCIN HYDROCHLORIDE | CHEMBL1201097 | 70S ribosome inhibitor |
| 1313 | Streptococcus pneumoniae | LINEZOLID | CHEMBL126 | 70S ribosome inhibitor |
| 6231 | Nematoda | MEBENDAZOLE | CHEMBL685 | Tubulin inhibitor |
| 2 | Bacteria | MECLOCYCLINE SULFOSALICYLATE | CHEMBL261772 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | METHACYCLINE HYDROCHLORIDE | CHEMBL2146123 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | MINOCYCLINE HYDROCHLORIDE | CHEMBL1200881 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | NEOMYCIN SULFATE | CHEMBL1275977 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | NETILMICIN SULFATE | CHEMBL1200872 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | OXYTETRACYCLINE | CHEMBL1517 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | OXYTETRACYCLINE CALCIUM | CHEMBL2068727 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | OXYTETRACYCLINE HYDROCHLORIDE | CHEMBL1607480 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | PAROMOMYCIN SULFATE | CHEMBL2206196 | Bacterial 70S ribosome inhibitor |
| 1773 | Mycobacterium tuberculosis | PYRAZINAMIDE | CHEMBL614 | 70S ribosome inhibitor |
| 2 | Bacteria | QUINUPRISTIN | CHEMBL1200649 | Bacterial 70S ribosome inhibitor |
| 1280 | Staphylococcus aureus | RETAPAMULIN | CHEMBL409542 | 70S ribosome inhibitor |
| 1314 | Streptococcus pyogenes | RETAPAMULIN | CHEMBL409542 | 70S ribosome inhibitor |
| 2 | Bacteria | RIFABUTIN | CHEMBL444633 | Bacterial DNA-directed RNA polymerase inhibitor |
| 2 | Bacteria | RIFAMPICIN | CHEMBL374478 | Bacterial DNA-directed RNA polymerase inhibitor |
| 2 | Bacteria | RIFAPENTINE | CHEMBL1660 | Bacterial DNA-directed RNA polymerase inhibitor |
| 2 | Bacteria | RIFAXIMIN | CHEMBL1617 | Bacterial DNA-directed RNA polymerase inhibitor |
| 2 | Bacteria | SPECTINOMYCIN HYDROCHLORIDE | CHEMBL1200407 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | STREPTOMYCIN SULFATE | CHEMBL453087 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | TELITHROMYCIN | CHEMBL1136 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | TETRACYCLINE | CHEMBL1440 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | TETRACYCLINE HYDROCHLORIDE | CHEMBL454950 | Bacterial 70S ribosome inhibitor |
| 2 | Bacteria | TETRACYCLINE PHOSPHATE COMPLEX | CHEMBL1201071 | Bacterial 70S ribosome inhibitor |
| 6231 | Nematoda | THIABENDAZOLE | CHEMBL625 | Tubulin inhibitor |
| 2 | Bacteria | TIGECYCLINE | CHEMBL376140 | Bacterial 70S ribosome inhibitor |
| 287 | Pseudomonas aeruginosa | TOBRAMYCIN | CHEMBL1747 | 70S ribosome inhibitor |
| 287 | Pseudomonas aeruginosa | TOBRAMYCIN SULFATE | CHEMBL1200780 | 70S ribosome inhibitor |
| 1313 | Streptococcus pneumoniae | TROLEANDOMYCIN | CHEMBL564085 | 70S ribosome inhibitor |
| 1773 | Mycobacterium tuberculosis | VIOMYCIN SULFATE | CHEMBL1200526 | 70S ribosome inhibitor |

Table : Trypanosoma brucei drugs.

|  |  |  |
| --- | --- | --- |
| **pref\_name** | **chembl\_id** | **mechanism\_of\_action** |
| ALBENDAZOLE | CHEMBL1483 | Tubulin inhibitor |
| AMIKACIN SULFATE | CHEMBL1164318 | Bacterial 70S ribosome inhibitor |
| CAPREOMYCIN SULFATE | CHEMBL2218913 | 70S ribosome inhibitor |
| CHLORAMPHENICOL | CHEMBL130 | Bacterial 70S ribosome inhibitor |
| CHLORAMPHENICOL PALMITATE | CHEMBL1506 | Bacterial 70S ribosome inhibitor |
| CHLORAMPHENICOL SODIUM SUCCINATE | CHEMBL1200729 | Bacterial 70S ribosome inhibitor |
| CHLORTETRACYCLINE HYDROCHLORIDE | CHEMBL2146063 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN HYDROCHLORIDE | CHEMBL1200588 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN PALMITATE HYDROCHLORIDE | CHEMBL1237086 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN PHOSPHATE | CHEMBL3184512 | Bacterial 70S ribosome inhibitor |
| DALFOPRISTIN | CHEMBL1200937 | Bacterial 70S ribosome inhibitor |
| DEMECLOCYCLINE HYDROCHLORIDE | CHEMBL1200474 | Bacterial 70S ribosome inhibitor |
| DIRITHROMYCIN | CHEMBL3039471 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN | CHEMBL532 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN ESTOLATE | CHEMBL2218877 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN ETHYLSUCCINATE | CHEMBL1200688 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN GLUCEPTATE | CHEMBL1200657 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN LACTOBIONATE | CHEMBL1200506 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN STEARATE | CHEMBL1200510 | Bacterial 70S ribosome inhibitor |
| GENTAMICIN SULFATE | CHEMBL3039594 | Bacterial 70S ribosome inhibitor |
| GRISEOFULVIN | CHEMBL562 | Tubulin inhibitor |
| KANAMYCIN SULFATE | CHEMBL1446 | Bacterial 70S ribosome inhibitor |
| LINCOMYCIN HYDROCHLORIDE | CHEMBL1201097 | 70S ribosome inhibitor |
| LINEZOLID | CHEMBL126 | 70S ribosome inhibitor |
| MEBENDAZOLE | CHEMBL685 | Tubulin inhibitor |
| MECLOCYCLINE SULFOSALICYLATE | CHEMBL261772 | Bacterial 70S ribosome inhibitor |
| METHACYCLINE HYDROCHLORIDE | CHEMBL2146123 | Bacterial 70S ribosome inhibitor |
| MINOCYCLINE HYDROCHLORIDE | CHEMBL1200881 | Bacterial 70S ribosome inhibitor |
| NEOMYCIN SULFATE | CHEMBL1275977 | Bacterial 70S ribosome inhibitor |
| NETILMICIN SULFATE | CHEMBL1200872 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE | CHEMBL1517 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE CALCIUM | CHEMBL2068727 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE HYDROCHLORIDE | CHEMBL1607480 | Bacterial 70S ribosome inhibitor |
| PAROMOMYCIN SULFATE | CHEMBL2206196 | Bacterial 70S ribosome inhibitor |
| PYRAZINAMIDE | CHEMBL614 | 70S ribosome inhibitor |
| QUINUPRISTIN | CHEMBL1200649 | Bacterial 70S ribosome inhibitor |
| RETAPAMULIN | CHEMBL409542 | 70S ribosome inhibitor |
| RIFABUTIN | CHEMBL444633 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAMPICIN | CHEMBL374478 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAPENTINE | CHEMBL1660 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAXIMIN | CHEMBL1617 | Bacterial DNA-directed RNA polymerase inhibitor |
| SPECTINOMYCIN HYDROCHLORIDE | CHEMBL1200407 | Bacterial 70S ribosome inhibitor |
| STREPTOMYCIN SULFATE | CHEMBL453087 | Bacterial 70S ribosome inhibitor |
| TELITHROMYCIN | CHEMBL1136 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE | CHEMBL1440 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE HYDROCHLORIDE | CHEMBL454950 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE PHOSPHATE COMPLEX | CHEMBL1201071 | Bacterial 70S ribosome inhibitor |
| THIABENDAZOLE | CHEMBL625 | Tubulin inhibitor |
| TIGECYCLINE | CHEMBL376140 | Bacterial 70S ribosome inhibitor |
| TOBRAMYCIN | CHEMBL1747 | 70S ribosome inhibitor |
| TOBRAMYCIN SULFATE | CHEMBL1200780 | 70S ribosome inhibitor |
| TROLEANDOMYCIN | CHEMBL564085 | 70S ribosome inhibitor |
| VIOMYCIN SULFATE | CHEMBL1200526 | 70S ribosome inhibitor |

## Trypanosoma brucei drugs query.

select distinct md.pref\_name, md.chembl\_id, mechanism\_of\_action

FROM blast\_statistics bs

, tax\_norm\_threshold t

, target\_components tc

, target\_dictionary td

, drug\_mechanism dm

, molecule\_dictionary md

, binding\_sites bind

WHERE bs.score > t.threshold

and bs.tax\_id = t.tax\_id

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and bs.tax\_id = 5691

and bs.tax\_id = t.tax\_id

and tc.tid = td.tid

and bind.tid = tc.tid

and dm.site\_id = bind.site\_id

and md.molregno = dm.molregno

and not exists (select 1

from exclude\_organisms eo

where eo.tax\_id = td.tax\_id

)

order by md.pref\_name;

## Trypanosoma brucei targets query

select max(bs.score) score, chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

from

(select distinct td.chembl\_id, td.target\_type, td.organism, td.tax\_id, bs.targ\_comp

from blast\_statistics bs

, target\_components tc

, target\_dictionary td

, tax\_norm\_threshold tnt

where bs.tax\_id=5691

and bs.tax\_id = tnt.tax\_id

and bs.score > tnt.threshold

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and tc.tid = td.tid

and not exists (select 1

from exclude\_organisms xo

where td.tax\_id = xo.tax\_id

)

) targets

,blast\_statistics bs

where bs.targ\_comp = targets.targ\_comp

group by chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

order by score desc;

Table : Trypanosoma brucei targets



## Trypanosoma cruzi drugs query.

select distinct md.pref\_name, md.chembl\_id, mechanism\_of\_action

FROM blast\_statistics bs

, tax\_norm\_threshold t

, target\_components tc

, target\_dictionary td

, drug\_mechanism dm

, molecule\_dictionary md

, binding\_sites bind

WHERE bs.score > t.threshold

and bs.tax\_id = t.tax\_id

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and bs.tax\_id = 5693

and bs.tax\_id = t.tax\_id

and tc.tid = td.tid

and bind.tid = tc.tid

and dm.site\_id = bind.site\_id

and md.molregno = dm.molregno

and not exists (select 1

from exclude\_organisms eo

where eo.tax\_id = td.tax\_id

)

order by md.pref\_name;

## Trypanosoma cruzi targets query

select max(bs.score) score, chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

from

(select distinct td.chembl\_id, td.target\_type, td.organism, td.tax\_id, bs.targ\_comp

from blast\_statistics bs

, target\_components tc

, target\_dictionary td

, tax\_norm\_threshold tnt

where bs.tax\_id=5693

and bs.tax\_id = tnt.tax\_id

and bs.score > tnt.threshold

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and tc.tid = td.tid

and not exists (select 1

from exclude\_organisms xo

where td.tax\_id = xo.tax\_id

)

) targets

,blast\_statistics bs

where bs.targ\_comp = targets.targ\_comp

group by chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

order by score desc;

Table : Trypanosoma cruzi targets



Table : Leishmania drugs.

|  |  |  |
| --- | --- | --- |
| **pref\_name** | **chembl\_id** | **mechanism\_of\_action** |
| ALBENDAZOLE | CHEMBL1483 | Tubulin inhibitor |
| AMIKACIN SULFATE | CHEMBL1164318 | Bacterial 70S ribosome inhibitor |
| CAPREOMYCIN SULFATE | CHEMBL2218913 | 70S ribosome inhibitor |
| CHLORAMPHENICOL | CHEMBL130 | Bacterial 70S ribosome inhibitor |
| CHLORAMPHENICOL PALMITATE | CHEMBL1506 | Bacterial 70S ribosome inhibitor |
| CHLORAMPHENICOL SODIUM SUCCINATE | CHEMBL1200729 | Bacterial 70S ribosome inhibitor |
| CHLORTETRACYCLINE HYDROCHLORIDE | CHEMBL2146063 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN HYDROCHLORIDE | CHEMBL1200588 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN PALMITATE HYDROCHLORIDE | CHEMBL1237086 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN PHOSPHATE | CHEMBL3184512 | Bacterial 70S ribosome inhibitor |
| DALFOPRISTIN | CHEMBL1200937 | Bacterial 70S ribosome inhibitor |
| DEMECLOCYCLINE HYDROCHLORIDE | CHEMBL1200474 | Bacterial 70S ribosome inhibitor |
| DIRITHROMYCIN | CHEMBL3039471 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN | CHEMBL532 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN ESTOLATE | CHEMBL2218877 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN ETHYLSUCCINATE | CHEMBL1200688 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN GLUCEPTATE | CHEMBL1200657 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN LACTOBIONATE | CHEMBL1200506 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN STEARATE | CHEMBL1200510 | Bacterial 70S ribosome inhibitor |
| GENTAMICIN SULFATE | CHEMBL3039594 | Bacterial 70S ribosome inhibitor |
| GRISEOFULVIN | CHEMBL562 | Tubulin inhibitor |
| KANAMYCIN SULFATE | CHEMBL1446 | Bacterial 70S ribosome inhibitor |
| LINCOMYCIN HYDROCHLORIDE | CHEMBL1201097 | 70S ribosome inhibitor |
| LINEZOLID | CHEMBL126 | 70S ribosome inhibitor |
| MEBENDAZOLE | CHEMBL685 | Tubulin inhibitor |
| MECLOCYCLINE SULFOSALICYLATE | CHEMBL261772 | Bacterial 70S ribosome inhibitor |
| METHACYCLINE HYDROCHLORIDE | CHEMBL2146123 | Bacterial 70S ribosome inhibitor |
| MINOCYCLINE HYDROCHLORIDE | CHEMBL1200881 | Bacterial 70S ribosome inhibitor |
| NEOMYCIN SULFATE | CHEMBL1275977 | Bacterial 70S ribosome inhibitor |
| NETILMICIN SULFATE | CHEMBL1200872 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE | CHEMBL1517 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE CALCIUM | CHEMBL2068727 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE HYDROCHLORIDE | CHEMBL1607480 | Bacterial 70S ribosome inhibitor |
| PAROMOMYCIN SULFATE | CHEMBL2206196 | Bacterial 70S ribosome inhibitor |
| PYRAZINAMIDE | CHEMBL614 | 70S ribosome inhibitor |
| QUINUPRISTIN | CHEMBL1200649 | Bacterial 70S ribosome inhibitor |
| RETAPAMULIN | CHEMBL409542 | 70S ribosome inhibitor |
| RIFABUTIN | CHEMBL444633 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAMPICIN | CHEMBL374478 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAPENTINE | CHEMBL1660 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAXIMIN | CHEMBL1617 | Bacterial DNA-directed RNA polymerase inhibitor |
| SPECTINOMYCIN HYDROCHLORIDE | CHEMBL1200407 | Bacterial 70S ribosome inhibitor |
| STREPTOMYCIN SULFATE | CHEMBL453087 | Bacterial 70S ribosome inhibitor |
| TELITHROMYCIN | CHEMBL1136 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE | CHEMBL1440 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE HYDROCHLORIDE | CHEMBL454950 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE PHOSPHATE COMPLEX | CHEMBL1201071 | Bacterial 70S ribosome inhibitor |
| THIABENDAZOLE | CHEMBL625 | Tubulin inhibitor |
| TIGECYCLINE | CHEMBL376140 | Bacterial 70S ribosome inhibitor |
| TOBRAMYCIN | CHEMBL1747 | 70S ribosome inhibitor |
| TOBRAMYCIN SULFATE | CHEMBL1200780 | 70S ribosome inhibitor |
| TROLEANDOMYCIN | CHEMBL564085 | 70S ribosome inhibitor |
| VIOMYCIN SULFATE | CHEMBL1200526 | 70S ribosome inhibitor |

## Leishmania drugs query.

select distinct md.pref\_name, md.chembl\_id, mechanism\_of\_action

FROM blast\_statistics bs

, tax\_norm\_threshold t

, target\_components tc

, target\_dictionary td

, drug\_mechanism dm

, molecule\_dictionary md

, binding\_sites bind

WHERE bs.score > t.threshold

and bs.tax\_id = t.tax\_id

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and bs.tax\_id = 5664

and bs.tax\_id = t.tax\_id

and tc.tid = td.tid

and bind.tid = tc.tid

and dm.site\_id = bind.site\_id

and md.molregno = dm.molregno

and not exists (select 1

from exclude\_organisms eo

where eo.tax\_id = td.tax\_id

)

order by md.pref\_name;

## Leishmania major targets query

select max(bs.score) score, chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

from

(select distinct td.chembl\_id, td.target\_type, td.organism, td.tax\_id, bs.targ\_comp

from blast\_statistics bs

, target\_components tc

, target\_dictionary td

, tax\_norm\_threshold tnt

where bs.tax\_id=5664

and bs.tax\_id = tnt.tax\_id

and bs.score > tnt.threshold

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and tc.tid = td.tid

and not exists (select 1

from exclude\_organisms xo

where td.tax\_id = xo.tax\_id

)

) targets

,blast\_statistics bs

where bs.targ\_comp = targets.targ\_comp

group by chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

order by score desc;

Table :Leishmania major targets



Table : Chlamidia drugs.

|  |  |  |
| --- | --- | --- |
| **pref\_name** | **chembl\_id** | **mechanism\_of\_action** |
| AMIKACIN SULFATE | CHEMBL1164318 | Bacterial 70S ribosome inhibitor |
| CAPREOMYCIN SULFATE | CHEMBL2218913 | 70S ribosome inhibitor |
| CHLORAMPHENICOL | CHEMBL130 | Bacterial 70S ribosome inhibitor |
| CHLORAMPHENICOL PALMITATE | CHEMBL1506 | Bacterial 70S ribosome inhibitor |
| CHLORAMPHENICOL SODIUM SUCCINATE | CHEMBL1200729 | Bacterial 70S ribosome inhibitor |
| CHLORTETRACYCLINE HYDROCHLORIDE | CHEMBL2146063 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN HYDROCHLORIDE | CHEMBL1200588 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN PALMITATE HYDROCHLORIDE | CHEMBL1237086 | Bacterial 70S ribosome inhibitor |
| CLINDAMYCIN PHOSPHATE | CHEMBL3184512 | Bacterial 70S ribosome inhibitor |
| DALFOPRISTIN | CHEMBL1200937 | Bacterial 70S ribosome inhibitor |
| DEMECLOCYCLINE HYDROCHLORIDE | CHEMBL1200474 | Bacterial 70S ribosome inhibitor |
| DIRITHROMYCIN | CHEMBL3039471 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN | CHEMBL532 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN ESTOLATE | CHEMBL2218877 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN ETHYLSUCCINATE | CHEMBL1200688 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN GLUCEPTATE | CHEMBL1200657 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN LACTOBIONATE | CHEMBL1200506 | Bacterial 70S ribosome inhibitor |
| ERYTHROMYCIN STEARATE | CHEMBL1200510 | Bacterial 70S ribosome inhibitor |
| GENTAMICIN SULFATE | CHEMBL3039594 | Bacterial 70S ribosome inhibitor |
| KANAMYCIN SULFATE | CHEMBL1446 | Bacterial 70S ribosome inhibitor |
| LINCOMYCIN HYDROCHLORIDE | CHEMBL1201097 | 70S ribosome inhibitor |
| LINEZOLID | CHEMBL126 | 70S ribosome inhibitor |
| MECLOCYCLINE SULFOSALICYLATE | CHEMBL261772 | Bacterial 70S ribosome inhibitor |
| METHACYCLINE HYDROCHLORIDE | CHEMBL2146123 | Bacterial 70S ribosome inhibitor |
| MINOCYCLINE HYDROCHLORIDE | CHEMBL1200881 | Bacterial 70S ribosome inhibitor |
| NEOMYCIN SULFATE | CHEMBL1275977 | Bacterial 70S ribosome inhibitor |
| NETILMICIN SULFATE | CHEMBL1200872 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE | CHEMBL1517 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE CALCIUM | CHEMBL2068727 | Bacterial 70S ribosome inhibitor |
| OXYTETRACYCLINE HYDROCHLORIDE | CHEMBL1607480 | Bacterial 70S ribosome inhibitor |
| PAROMOMYCIN SULFATE | CHEMBL2206196 | Bacterial 70S ribosome inhibitor |
| PYRAZINAMIDE | CHEMBL614 | 70S ribosome inhibitor |
| QUINUPRISTIN | CHEMBL1200649 | Bacterial 70S ribosome inhibitor |
| RETAPAMULIN | CHEMBL409542 | 70S ribosome inhibitor |
| RIFABUTIN | CHEMBL444633 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAMPICIN | CHEMBL374478 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAPENTINE | CHEMBL1660 | Bacterial DNA-directed RNA polymerase inhibitor |
| RIFAXIMIN | CHEMBL1617 | Bacterial DNA-directed RNA polymerase inhibitor |
| SPECTINOMYCIN HYDROCHLORIDE | CHEMBL1200407 | Bacterial 70S ribosome inhibitor |
| STREPTOMYCIN SULFATE | CHEMBL453087 | Bacterial 70S ribosome inhibitor |
| TELITHROMYCIN | CHEMBL1136 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE | CHEMBL1440 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE HYDROCHLORIDE | CHEMBL454950 | Bacterial 70S ribosome inhibitor |
| TETRACYCLINE PHOSPHATE COMPLEX | CHEMBL1201071 | Bacterial 70S ribosome inhibitor |
| TIGECYCLINE | CHEMBL376140 | Bacterial 70S ribosome inhibitor |
| TOBRAMYCIN | CHEMBL1747 | 70S ribosome inhibitor |
| TOBRAMYCIN SULFATE | CHEMBL1200780 | 70S ribosome inhibitor |
| TROLEANDOMYCIN | CHEMBL564085 | 70S ribosome inhibitor |
| VIOMYCIN SULFATE | CHEMBL1200526 | 70S ribosome inhibitor |

## Chlamidia drugs query.

select distinct md.pref\_name, md.chembl\_id, mechanism\_of\_action

FROM blast\_statistics bs

, tax\_norm\_threshold t

, target\_components tc

, target\_dictionary td

, drug\_mechanism dm

, molecule\_dictionary md

, binding\_sites bind

WHERE bs.score > t.threshold

and bs.tax\_id = t.tax\_id

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and bs.tax\_id = 272561

and bs.tax\_id = t.tax\_id

and tc.tid = td.tid

and bind.tid = tc.tid

and dm.site\_id = bind.site\_id

and md.molregno = dm.molregno

and not exists (select 1

from exclude\_organisms eo

where eo.tax\_id = td.tax\_id

)

order by md.pref\_name;

## Chlamidia targets query

select max(bs.score) score, chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

from

(select distinct td.chembl\_id, td.target\_type, td.organism, td.tax\_id, bs.targ\_comp

from blast\_statistics bs

, target\_components tc

, target\_dictionary td

, tax\_norm\_threshold tnt

where bs.tax\_id=272561

and bs.tax\_id = tnt.tax\_id

and bs.score > tnt.threshold

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and tc.tid = td.tid

and not exists (select 1

from exclude\_organisms xo

where td.tax\_id = xo.tax\_id

)

) targets

,blast\_statistics bs

where bs.targ\_comp = targets.targ\_comp

group by chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

order by score desc;

Table : Chlamidia trochamatis targets



## Queries to find which Trypanosoma cruzi targets are not Trypanosoma brucei targets

First, we create to separate table for each species targets:

CREATE TABLE brucei\_targets as

select max(bs.score) score, chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

from

(select distinct td.chembl\_id, td.target\_type, td.organism, td.tax\_id, bs.targ\_comp

from blast\_statistics bs

, target\_components tc

, target\_dictionary td

, tax\_norm\_threshold tnt

where bs.tax\_id=5691

and bs.tax\_id = tnt.tax\_id

and bs.score > tnt.threshold

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and tc.tid = td.tid

and not exists (select 1

from exclude\_organisms xo

where td.tax\_id = xo.tax\_id

)

) targets

,blast\_statistics bs

where bs.targ\_comp = targets.targ\_comp

group by chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

order by score desc;

CREATE TABLE cruzi\_targets as

select max(bs.score) score, chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

from

(select distinct td.chembl\_id, td.target\_type, td.organism, td.tax\_id, bs.targ\_comp

from blast\_statistics bs

, target\_components tc

, target\_dictionary td

, tax\_norm\_threshold tnt

where bs.tax\_id=5693

and bs.tax\_id = tnt.tax\_id

and bs.score > tnt.threshold

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and tc.tid = td.tid

and not exists (select 1

from exclude\_organisms xo

where td.tax\_id = xo.tax\_id

)

) targets

,blast\_statistics bs

where bs.targ\_comp = targets.targ\_comp

group by chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

order by score desc;

chembl\_20=> select c.\*

chembl\_20-> from cruzi\_targets c

chembl\_20-> where not exists (select 1

chembl\_20(> from brucei\_targets b

chembl\_20(> where b.chembl\_id = c.chembl\_id);

score | chembl\_id | target\_type | organism | tax\_id | targ\_comp

-------+------------+----------------+--------------------------------+--------+-----------

211 | CHEMBL6057 | SINGLE PROTEIN | Saccharomyces cerevisiae S288c | 559292 | 4591

47.4 | CHEMBL3313 | SINGLE PROTEIN | Torpedo californica | 7787 | 1633

(2 rows)

## Toxoplasmosis Gondii drugs query

select distinct md.pref\_name, md.chembl\_id, mechanism\_of\_action

FROM blast\_statistics bs

, tax\_norm\_threshold t

, target\_components tc

, target\_dictionary td

, drug\_mechanism dm

, molecule\_dictionary md

, binding\_sites bind

WHERE bs.score > t.threshold

and bs.tax\_id = t.tax\_id

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and bs.tax\_id = 5811

and bs.tax\_id = t.tax\_id

and tc.tid = td.tid

and bind.tid = tc.tid

and dm.site\_id = bind.site\_id

and md.molregno = dm.molregno

and not exists (select 1

from exclude\_organisms eo

where eo.tax\_id = td.tax\_id

)

order by 1



## Toxoplasmosis Gondii targets query

select max(bs.score) score, chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

from

(select distinct td.chembl\_id, td.target\_type, td.organism, td.tax\_id, bs.targ\_comp

from blast\_statistics bs

, target\_components tc

, target\_dictionary td

, tax\_norm\_threshold tnt

where bs.tax\_id=5811

and bs.tax\_id = tnt.tax\_id

and bs.score > tnt.threshold

and bs.expect < .001

and bs.targ\_comp = tc.component\_id

and tc.tid = td.tid

and not exists (select 1

from exclude\_organisms xo

where td.tax\_id = xo.tax\_id

)

) targets

,blast\_statistics bs

where bs.targ\_comp = targets.targ\_comp

group by chembl\_id, target\_type, organism, targets.tax\_id, targets.targ\_comp

order by score desc



# Example Workbook for Toxoplasmosis gondii ETL

## Glimmer

for i in $(ls Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.\*.fa);do

~/glimmer3.02/scripts/g3-iterated.csh ${i} ${i}.genes

Done

## Extract

for i in $(ls Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.\*.fa);do

extract -2 ${i} ${i}.genes.coords > genes/${i}.genes

done

## chrom\_genes\_to\_proteins

From genes directory:

for i in $(ls \*.genes);do

perl ../chrom\_genes\_to\_proteins.pl ${i}

done

ls -d \*/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrIa.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrIX.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrVIII.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrIb.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrV.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrX.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrII.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrVI.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrXI.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrIII.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrVIIa.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrXII.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrIV.fa/

Toxoplasma\_gondii.ToxoDB-7.1.dna.chromosome.chrVIIb.fa/

## Rename the directories more simply.

for i in $(ls -d \*/);do

to\_name=`echo ${i} | cut -d \. -f 6 `

$c=`mv ${i} $to\_name`

$c

done

## BLAST

To the *genes* directory copy:

do\_blast\_all.sh

do\_blast.sh

blast\_stats.sh

extract\_header.pl

[jsinger@localhost 12:55:03 ~/genome/toxoplasmosis/genes] for i in $(ls -d \*/);do

> echo ${i}

> done

chrIa/

chrIb/

chrII/

chrIII/

chrIV/

chrIX/

chrV/

chrVI/

chrVIIa/

chrVIIb/

chrVIII/

chrX/

chrXI/

chrXII/

for i in $(ls -d \*/);do cp do\_blast.sh ${i}; done

./do\_blast\_all.sh

chrIa/

chrIb/

chrII/

chrIII/

chrIV/

chrIX/

chrV/

chrVI/

chrVIIa/

chrVIIb/

chrVIII/

chrX/

chrXI/

chrXII/

## Extract BLAST statistics from BLAST reports.

From *genes* directory:

./blast\_stats.sh

## Create populate\_blast\_statistics.sql

Manual step: update script to update the tax\_id value to 5811 for Toxoplasmosis gondii, then run the script.

~/genome/toxoplasmosis/genes] ../create\_populate\_blast\_statistics.sh

## Upload BLAST stats to postgres database on work VM.

Start the PostgreSQL server on the work VM:

[postgres@localhost 12:43:20 ~] ./startdb &

[1] 22776

[postgres@localhost 12:43:37 ~] server starting

[1]+ Done ./startdb

[postgres@localhost 12:43:39 ~] exit

logout

[jsinger@localhost 12:43:45 ~/genome/toxoplasmosis/genes]

Connect to the database:

psql -U postgres -d chembl\_targets

psql (8.4.20)

Type "help" for help.

Chembl\_targets=#

\i ‘populate\_blas psql -U postgres

psql (8.4.20)

Type "help" for help.

\i ‘populate\_blast\_statistics.sql’

.

.

chembl\_targets=# select count(\*)

chembl\_targets-# from blast\_statistics

chembl\_targets-# where tax\_id=508771;

count

--------

664776

(1 row)

## Export data to a file on the work VM.

\copy (select \* from blast\_statistics where tax\_id=508771) to '~/blast\_statistics\_508771.txt'

## Download data to Windows integration environment.

$ scp jsinger@192.168.1.12:~/blast\_statistics\_508771.txt .

jsinger@192.168.1.12's password:

blast\_statistics\_508771.txt 100% 43MB 42.6MB/s 00:00

## Import data into MyChEMBL database.

chembl\_20=> \copy blast\_statistics from 'C:/jeremy/toxoplasmosis/blast\_statistics\_508771.txt'

COPY 664776