Published online 17 September 2009 *Nucleic Acids Research, 2009, Vol. 37, No. 20* ***6643–6654***
doi:10.1093/nar/gkp698

FIGfams: yet another set of protein families

Folker Meyer[1,2,*], Ross Overbeek[3] and Alex Rodriguez[2]

[1]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, [2]Computation Institute, University of Chicago/Argonne National Laboratory, Chicago, USA and [3]Fellowship for the Interpretation of Genomes, Burr Ridge 60527, IL, USA

Received June 8, 2009; Revised August 5, 2009; Accepted August 6, 2009

ABSTRACT

We present FIGfams, a new collection of over 100 000 protein families that are the product of manual curation and close strain comparison. Using the Subsystem approach the manual curation is carried out, ensuring a previously unattained degree of throughput and consistency. FIGfams are based on over 950 000 manually annotated proteins and across many hundred Bacteria and Archaea. Associated with each FIGfam is a two-tiered, rapid, accurate decision procedure to determine family membership for new proteins. FIGfams are freely available under an open source license. These can be downloaded at ftp://ftp.theseed.org/FIGfams/. The web site for FIGfams is http://www.theseed.org/wiki/FIGfams/

INTRODUCTION

Progress in DNA sequencing technology has led to an abundance of nucleotide sequences in community databases (1). As the pace of sequencing increases (2) so does the importance of creating tools to accurately describe the protein functions encoded in the DNA sequences. These descriptions, or 'annotations', are created by using a variety of bioinformatics tools and databases. One of our most valuable clues to decipher functions of unknown proteins is their comparison with existing proteins (3).

Several groups are curating large sets of existing genomes (4–6), and even more groups are focusing their curation efforts on sets of proteins (6–12). The common denominator of all these approaches is that they need to rely on automatic propagation of 'correct' annotations using bioinformatics techniques. The reason for this is that the number of newly sequenced proteins clearly exceeds the available manpower when following the established 'one-protein-at-a-time' annotation approach.

The same issue explains why few authors of genome data sets are able to spend time curating the genome data sets they deposited in Genbank. For the majority of genomes, new discoveries are not used to update after the time of initial submission. As a result, even as our knowledge of protein function is developing, the existing genome data sets are often out of date. This situation presents a serious dilemma for the analysis of new sequences because comparison with existing data is the source for annotations of new data, the results of which are then submitted to a number of repositories (3).

Most of the tools for analyzing new sequence data use BLAST (13) or more sophisticated bioinformatics tools such as HMMs (7,14–16), PSSMs (17), or integrations of multiple tools (18). Both with BLAST-type searches and with more complex representations, the construction and use of protein families are central to most accurate annotation efforts (6–8,19); see ref. (3) for a discussion.

The common requirement for these approaches is that curation of initial protein sets [in the case of TIGRFAMS (7)] or assignment of protein family functions [in the case of PIRfams or OrthoMCL (20)] needs to be performed by an expert in the protein being analyzed. As with manual curation of complete genomes, however, manpower for the creation of these core data sets has been the limiting factor so far. Curation of data has restricted the number of protein families; for example, in the manually curated TIGRFAM core set, only 1972 *TIGR equivalogs* exist in Release 8.0 of the TIGRFAMs (21).

This bottleneck for manually curating the protein families can be overcome by using the *Subsystem* approach (22) for the construction and maintenance of protein families. Subsystem-based curation provides a scalable alternative to the traditional manual curation efforts for protein families.

Subsystems and FIGfams

Basically, a *Subsystem* is a collection of abstract functional roles and a spreadsheet mapping those functional roles to genes across multiple genomes. The spreadsheet has

*To whom correspondence should be addressed. Email: folker@anl.gov

functional roles as columns, and each row corresponds to a single genome. Each cell contains the genes in the corresponding genome that implement the functional row given by the column. Together, the *Subsystem* and the *Subsystem spreadsheet* are referred to as a *populated Subsystem*. The current collection of manually curated Subsystems includes over 800 subsystems containing over 6400 functional roles, to which >950 000 genes are connected; see ref. (5) for details.

The FIGfam effort may be thought of as constructing the infrastructure needed to automatically project the manual annotations maintained within the Subsystem collection.

Defining FIGfams

FIGfams are sets of isofunctional homologues (23). In other words each FIGfam is supposed to contain a set of proteins that are end-to-end homologous and share a common function. The current release (10.0) contains roughly 107 000 families, from careful manual curation using Subsystems (22) and automatic annotation of closely related strains. The families from closely related strains are based on sequence similarity and conserved genomic context. Figure 1 gives an overview of FIGfam creation and the use of FIGfams for automatic annotation.

More formally, each FIGfam can be defined as a four-tuple: (*ID*, *protein-set, decision-procedure, family-function*), where

(1) The *ID* is a stable, unique identifier that describes the family and allows linking to a web site describing the protein family;
(2) The *protein-set* is a set of protein sequences that are similar over essentially their entire length (i.e. they share a common domain structure; we allow for slight differences in the C-terminal because the correct determination of start codons is still somewhat imperfect and would artificially split protein sets otherwise belonging to the same family) and are believed to implement a common function;
(3) The *decision-procedure* is a decision procedure that, given a new protein sequence as input, decides whether the new sequence should be considered as 'part of the same family'; and
(4) The *family-function* is the function believed to be implemented by all members of the protein-set.

Creation and maintenance of FIGfams

The construction of FIGfams is based on forming *protein-sets* in cases in which it can more or less reliably be asserted that sequences implement identical functions. Currently, there are two scenarios for creating a FIGfam: one based on subsystems and the other based on closely related strains.

The FIGfams are constructed by inferring which pairs of genes must be placed in the same FIGfam (see below for detailed discussion in each of the scenarios) and then forming the set of FIGfams as the maximum set of protein-sets consistent with the pairwise constraints.

Families constructed from subsystems. Two proteins will be placed in the same FIGfam if they are similar over their entire length and they occur within the same column of a Subsystem (Figure 2). Genes within the same column of a Subsystem implement a common function.

Two genes within the same column of a Subsystem will be placed in distinct FIGfams when one is multifunctional and the other is not.

A low degree of sequence similarity (*e*-value of 10^{-10}) will lead to the creation of multiple FIGfams with the same function.

These rules firmly ground the FIGfams in the manual curation effort maintaining the Subsystems. If at any point it appears that two proteins are part of a single protein-set but are believed to implement distinct functions, the solution is to make sure that Subsystems exist to which the proteins are attached. That is, if there is a solid reason to believe that the proteins implement different functions, the way to force this to occur within FIGfams is to make sure that the manual effort reflects the reasoning that the functions are distinct.

Families constructed from closely related strains. If two or more sequenced genomes are from closely related strains, it is usually possible to trivially establish a reliable correspondence between 90 and 95% of the genes within the genomes. This is illustrated by the display of corresponding regions from the genomes shown in Figure 3. Of course, one needs an automated tool that uses specific rules to detect reliable correspondences, and many have been constructed.

We use a simple tool to implement this process; a description is provided in the Appendix. We note that as more genomes are sequenced from closely related strains, the number of correspondences will rapidly grow.

Curating FIGfams over time—connecting changes in Subsystems to changes in FIGfams

The current set of FIGfams will rapidly become outdated as the characterization of specific proteins continues to improve. New experimental results reported in the literature and careful manual annotation of the current Subsystems naturally force changes and additions to the FIGfam collection. A central feature of the existing collection is that families will automatically be split, merged, and added in response to the addition of new Subsystems or corrections of errors in the existing collection. In a field experiencing such rapid advance, this automated coupling of changes in Subsystems to derived changes in FIGfams is vital.

Once each month, the existing FIGfams are scanned looking for cases in which a family contains two proteins such that both proteins occur in Subsystems and the functions of the proteins are not identical. Such a situation forces a split of the FIGfam, which can be achieved automatically. Similarly, if two families are found to each contain proteins that occur in Subsystems

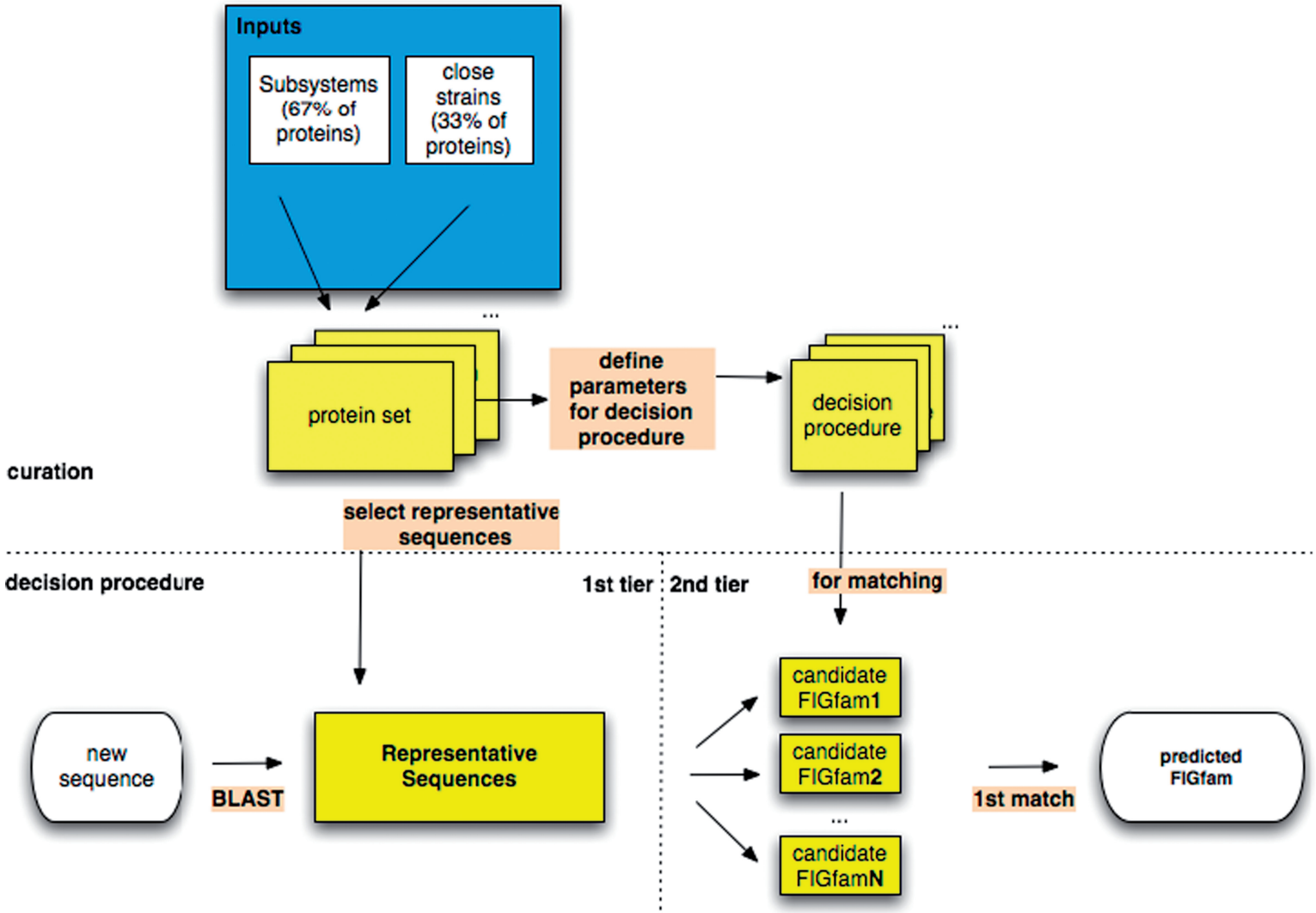


Figure 1. Overview of FIGfam creation and use. From two different input sets, FIGfam protein sets are defined. Subsequently, parameters for decision procedures and representative proteins are computed. On the decision procedure side, a new sequence is classified via BLAST searching the representative database (first tier); then the decision procedure associated with each candidate FIGfam is executed (second tier); and finally the first match for any candidate FIGfam is reported back.

Organism	Domain	Variant	argA/J	argB	argC	argD	argE	argF/I	argG	argH	ArgJ	ArgR
	Bacteria		All	All	All	All	All	All	All	All	All	All
Nocardia farcinica IFM 10152 (247156.1)	Bacteria	1	1942	1943	1941	1944	4560	1945	1967	1968	1942	1946
Rubrobacter xylanophilus DSM 9941 (266117.6)	Bacteria	1	2525	2526	2524	2527	1493 2176	2528	2521	897	2525	2523
Bacillus anthracis str. 'Ames Ancestor' (261594.1)	Bacteria	1	4320	4319	4321	4318	2417 766	4317	4815	3608 4814	4320	4361
Bacillus cereus ATCC 14579 (226900.1)	Bacteria	1	3958	3957	5486	3956	2123	358 3955	4448	3395 4447	3958	356
Bacillus halodurans C-125 (272558.1)	Bacteria	1	2899	2898	2900	2897	1059 2678	2894	3187	3186	2899	
Listeria monocytogenes str. 1/2a F6854 (267409.1)	Bacteria	1	2095	2094	2096	2093	1809	2092	1114	1115	2095	1263
Lactobacillus plantarum WCFS1 (220668.1)	Bacteria	1	441	442	440	443	1597 2351	444	657	658	441	1166 1331
Rhodopseudomonas palustris CGA009 (258594.1)	Bacteria	1	3604 589	626	2476	3712 4750	2313 3015	4749	389	4720	589	
Pelagibacter ubique HTCC1062 (335992.3)	Bacteria	1	878	721	1238	812	466	813	57	522	878	
Geobacter metallireducens GS-15 (269799.3)	Bacteria	1	1027	234	636	235		236	237	240	1027	

Figure 2. FIGfams from a Subsystem. The manual curation of the Arginine Biosynthesis Subsystem led to the creation of multiple FIGfams. The colored background indicates FIGfam membership. A single column can contain multiple FIGfams.

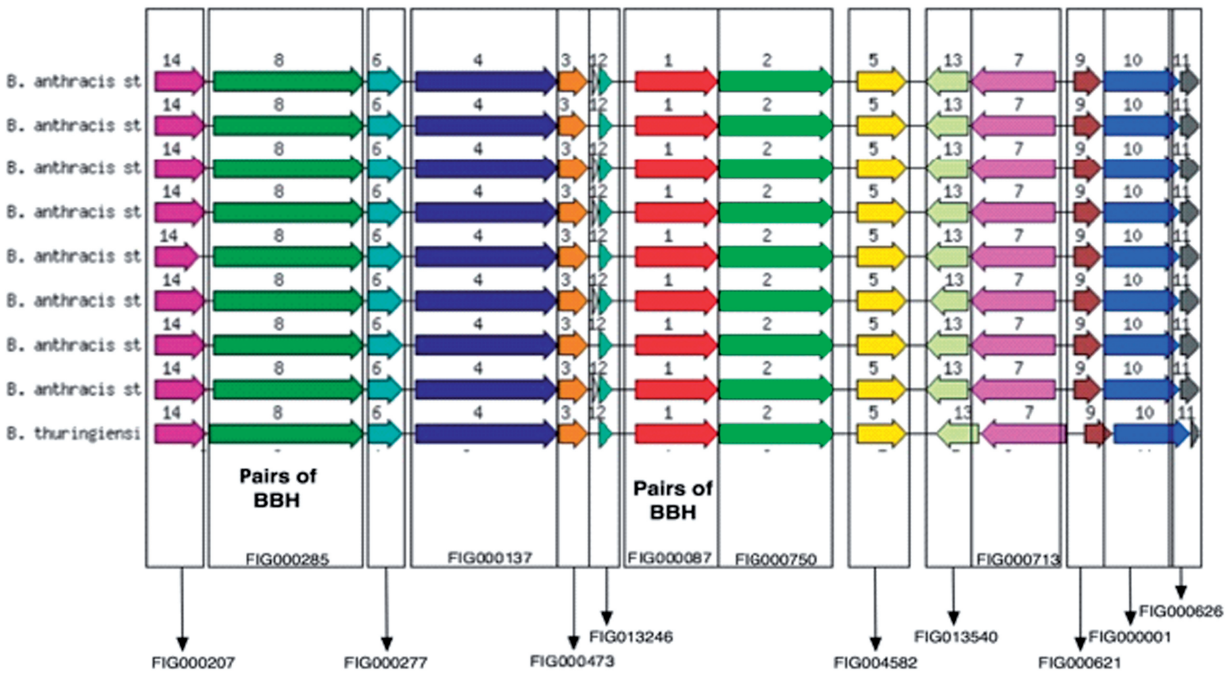


Figure 3. FIGfams constructed from closely related strains. The graphic depicts the chromosomal neighborhood of histidyl-tRNA synthetase in closely related *Bacilli*. The same color indicates a set of similar proteins (bidirectional best hits) that form a FIGfam. Each of the FIGfams has a different functional role; see Table 1 in the Supplementary Data.

and if these proteins are globally similar and implement identical functions, then the families are automatically merged. As new Subsystems are implemented, we find cases in which globally similar proteins that all connect to Subsystems and implement the same function are not yet members of any FIGfam. If they can be added to existing FIGfams, they are; if not, new FIGfams are automatically created.

COMPARISON OF FIGfams, TIGRFAMs AND PIRSFs

Many groups have attempted the curation of protein families over time; here we discuss the differences and similarities among FIGfams and two other prominent efforts. All three efforts curate families of homeomorphic proteins, requiring full-length sequence similarity and common domain structure within each family. These requirements make them different from efforts such as the PFAM database (14) that provide protein domains.

The technologies used for curation are very different, resulting in vastly different throughput of the various protein family curation strategies. In the case of FIGfams roughly 23 000 protein families are the results of manual curation; that number is in stark contrast to 1 900 TIGRFAM equivalogs with manually curated kernel (or 'SEED') alignments.

Our understanding is that curation of TIGRFAMs starts with manual creation of a SEED alignment and that an HMM is then created from that alignment. The subsequent curation effort is the definition of thresholds that allow the HMM-based decision procedure to reliably detect new members of the protein family. It should be noted that the TIGRFAMs distribution contains a large number of non-equivalog based HMMs that are not a result of manual curation of on a protein family level, they represent broader classes of proteins.

The PIRSF concept involves the formation of a shallow hierarchy (superfamilies, containing families, containing subfamilies). The goal is somewhat different from, and perhaps more ambitious than, that present in FIGfams. The PIRSF hierarchy attempts to group things into a hierarchy based on physical properties, realizing that significant shifts in physical properties usually correlate closely with functional properties. The FIGfams are based on the *Subsystems* view (22) in which a bacterial organism is composed of a set of functional Subsystems, and each active variant of a Subsystem is thought of as a set of functional roles. Proteins implement one or more functional roles. Grouping sets of functional roles induces the shallow hierarchy imposed by Subsystems.

Both notions involve protein families made up of proteins that are globally (i.e. full-length) similar. In most cases, the lowest-level PIRSF family (either a family or a subfamily) is composed of proteins that are believed to implement a common function. Hence, we believe there exists a close correspondence between the families produced by the two efforts, and the correspondence will improve as uncertainties are gradually eliminated. At this point the differences in perspective become most apparent in the way families are constructed. In the FIGfam effort, the major concern is to avoid placing two proteins with different functions into the same set. This leads to many small protein families (and many distinct families that contain closely similar sequences). In a somewhat oversimplified view, the PIRSF families are large groupings of homologous proteins in which the precise, distinct functions of subfamilies are gradually worked out, whereas the FIGfams start with no groups and conservatively gather proteins of identical function. To provide perspective on what this means, we note that the FIGfams collection now includes over 100 000 families, over half of which contain three or fewer members, whereas the PIRSF contains 32 000 families.

All three groups maintain sets of proteins and suggest a function for all members of that family. For TIGRfams, the set of proteins is used for the kernel (or 'SEED') alignment subsequent used to create an HMM. For FIGfams and PIRSF, complete sets of proteins are maintained for each family. Table 1 lists further differences and similarities.

In the case of the HMM based protein families it should be noted that HMM based methods have been shown to have excellent results in the detection of remote homologies, however this comparison does not evaluate

Table 1. Comparison of protein family creation and maintenance

	FIGfams	PIRSFs	TIGRFAM
Family creation	Via Subsystem curation and close strains	Via automatically generated sets of homeomorphic[a] proteins incorporating protein domain knowledge	Via manually curated kernel (or 'SEED') alignment
Extending an existing family	Include new genomes in Subsystem	Automatic placement in homeomorphic family[b]	Adjust threshold for trusted HMM score
Creating new families	Via new Subsystem creation	Via automated process (see above)	New SEED and HMM
Curation of function (for all proteins in set)[c]	Via Subsystem inclusion	Define family function for set	Not applicable
Scope	Bacteria and Archaea	Universal	Bacteria and Archaea
Number of families	107 233	33 599	3603
Families with proteins with manually curated function	20 699	327[d]	1920
Number of proteins in manually curated families	970 682	6040[d]	Not applicable[d]

[a]Homeomorphic = full-length homologuous with common domain architecture.
[b]Info from http://pir.georgetown.edu/pirwww/about/doc/tutorials/pirsftutorial.ppt.
[c]TIGRFAM protein sets are not curated; only SEED sets and HMM thresholds are curated.
[d]Info from ftp://ftp.pir.georgetown.edu/pir_databases/pirsf/data/pirsf_full_validated_oo.readme.

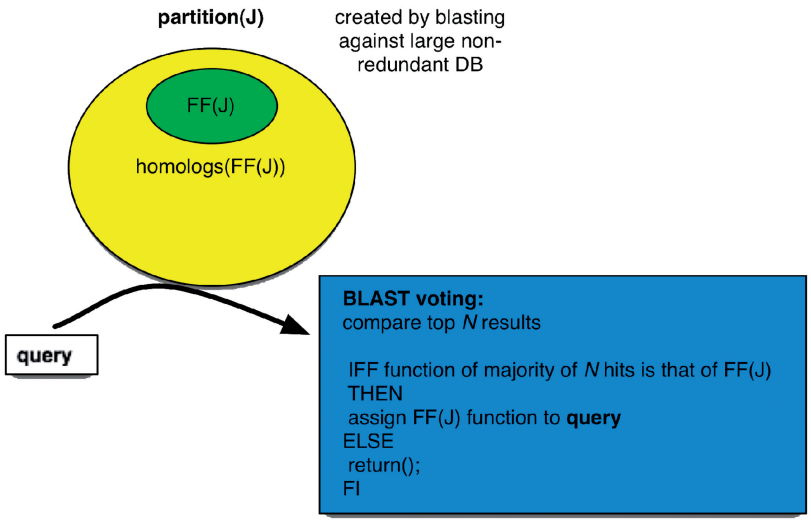


Figure 4. BLAST voting explained. For a given FIGfam *J* the set of all homologs and the FIGfam members form a *partition* (*J*). (We use an *e*-value cut-off of 0.01 and a minimal similarity of 30% for inclusion of sequences in the set of homologs.) For new sequences, we perform a BLAST searches against the partition and assign the family function for FIGfam *J* if the majority of the top *N* (either 3 or 10 depending on the size of the FIGfam) are annotated with the family function. Note that no function is returned if the majority of BLAST hits do not share the family function.

this ability. We also did not evaluate the option of careful manual curation of threshold values for HMMs that to avoid false positives.

FIGfam DECISION PROCEDURES

The third component of each FIGfam, the *decision-procedure*, is used to answer the question 'For a new sequence X, should X be considered part of the FIGfam?'

Various technologies exist to implement this decision procedure, ranging from a simple 'take the best BLAST hit' approach to more sophisticated approaches using machine-learning technology such as hidden Markov models and position-specific scoring matrices.

The current implementation of the FIGfam decision is two-tiered. A global fast screening procedure will create a set of candidate FIGfams for a target sequence *X*. A slower, more accurate decision procedure is associated with the individual family.

In order to provide maximum throughput, the initial screening is implemented by using a database of representative sequences for all FIGfams. Each FIGfam is associated with a set of representative sequences. We rely on comparing all sequences within a FIGfam to each other via BLAST (cut-off $1e^{-10}$) to form the set. A single randomly chosen sequence will be used in this database to represent all sequences that are within a $1e^{-10}$ distance.

Each candidate FIGfam has its own decision procedure: we currently implement two distinct procedures. Manual curation is used to assign decision procedures, currently most families use the BLAST voting procedure.

Similarity bounds decision procedure—a bounds list is generated for each member of the protein family by using the learning data. The bounds list is essentially a threshold for trusted BLAST scores, in which the user can safely assign a functional role if the BLAST score falls below a designated threshold. The decision procedure goes through the closest BLAST hits in the family (from the sequence being considered for membership), and the individual members are examined to see whether the hits fall within the 'safe' threshold. If there are ever more 'safe' hits than those that are not, the process ends successfully. Otherwise, the sequence cannot reliably be assigned to the set.

BLAST voting decision procedure—the top 10 and 20 BLAST results are voted on to select the functional role with the most hits. When two or more functional roles have an equal number of votes as the top choice, no assignment is given. Figure 4 provides details on the BLAST voting algorithm.

We have evaluated a series of decisions procedures when designing and implementing FIGfams. In the remainder of this section, we revisit some of the issues that led to the existing implementation. The first test targets pure runtime performance, the second tests for robust classification performance in the face of noisy data, and the third compares performance with two related protein family efforts.

Test 1: a simple case—finding ribosomal protein L33p

FIGfam FIG000053 has a family function of LSU ribosomal protein L33p. The decision procedure for this family is straightforward. The central issue in choosing a decision procedure is just performance.

Specifically, we evaluated the use of an HMM as opposed to the use of BLAST using the set of sequences (1313 sequences) from the FIGfam FIG000053. The BLAST test was performed by first BLASTing the sequences against a set of representative sequences of the FIGfams. Subsequently, the resulting FIGfams in a threshold were further evaluated by BLASTing against

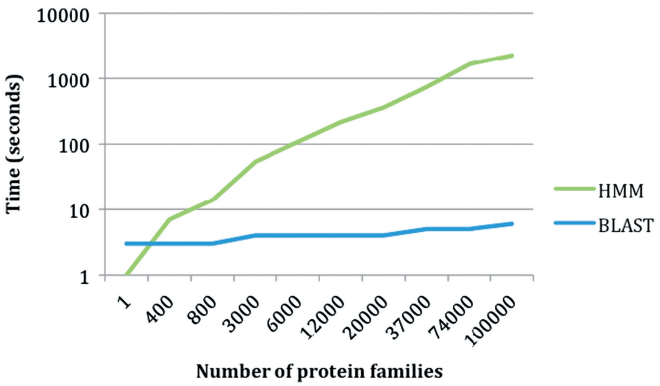


Figure 5. Comparison of the time (in seconds) spent searching growing numbers of FIGfams via HMMs and BLAST. Time for searching via HMMs increases with the number of families; for the two-tiered BLAST-based decision procedure, the time is constant. The time required to perform the search with BLAST remains at <10 s. These computations were performed on a current desktop machine running Linux (3 GHz Intel CPU, 4 GB RAM); with faster CPUs the ratios will remain stable.

the specific FIGfams. The HMM test was performed by doing a search against the set of HMMs for each candidate protein. Currently, there are over 100 000 FIGfams, which hampers the ability to perform an HMM search on the FIGfams.

Both methods generated the same prediction for the test sequences. Runtime requirements were dramatically different, however, with the HMM procedure taking significantly more resources. As shown in Figure 5 the time required for the HMM case depends on the number of families searched, whereas the time required for the two-tiered FIGfam strategy shows only minimal variation with increasing numbers of protein families searched. Note that the initial rapid screening stage does not show significant variations with growing numbers of families.

Test 2: HisAb—using different decision procedures to distinguish between two similar proteins

Curation of data is an error-prone process, and any decision procedure employed to recognize new family members (or predict functions for novel genes) is likely to include erroneously annotated false positive members. We therefore have devised a test to 'poison' carefully verified protein families with errors.

Several decision procedures were tested using protein sequences from a protein family implementing the function *Histidyl-tRNA synthetase (EC 6.1.1.21)* and a second family implementing *ATP phosphoribosyltransferase regulatory subunit (EC 2.4.2.17)*. These two families contain protein sequences that are similar to one another [see (24) for details], which has led to numerous errors in annotation over the past few years.

We call the union of these two families the *HisAb* set, and we consider in detail the issue of how well different decision procedures separate the entries of *HisAb* into the two protein families. The *HisAb* set offers a framework for comparison in that the sequences are closely homologous, and we believe that we have accurately annotated (manually) the entire set of sequences. The FIGfam FIG000087 implements *Histidyl-tRNA synthetase (EC 6.1.1.21)*, and FIG000865 implements *ATP phosphoribosyltransferase regulatory subunit (EC 2.4.2.17)*.

One obvious way to evaluate each decision procedure would be to take each sequence from the *HisAb* set, delete the sequence from the family containing it, and then examine the results of asking, for each of the two families, 'Does the sequence belong in this family?' For each of these single-sequence experiments there are four possible outcomes: the decision procedure can place the sequence into FIG000087, FIG000865, both, or neither. If we perform this experiment for each sequence in each of the two families for each of the decision procedures we wish to evaluate, we gain some insight into the relative merits of the set of decision procedures (we display the results of this experiment below). However, we can also investigate the situation in which some percentage of the sequences in *HisAb* has been assigned to the wrong protein family. This more closely resembles the real situation for most paralogous families, and we believe that it offers a more comprehensive way to evaluate the relative merits of the decision procedures.

Overall, the decision procedures that performed the best in the presence of misannotated sequences were the BLAST voting algorithms (top 1, top 20 BLAST results). The number of BLAST hits to vote on was directly proportional to the size of the protein family being tested. The HMM decision procedure was outperformed by all other decision procedures, and it was also more time consuming.

Test methodology. Each decision procedure was tested by using a jack-knife approach, where a sequence was used for testing the decision procedure, while the rest of the sequences were used as the learning data to create the model. This process was iterated several times over the number of total sequences in the learning data. In addition to experimenting with each decision procedure using the gold standard, errors were introduced to the gold standard assignments by switching a sequence's assigned functional role in the learning data. The goal is to view how each decision procedure behaves in the presence of errors in the annotations using a controlled environment. Each decision procedure was tested with 0, 10, 20, 30 and 40% annotation errors in the learning data. The accuracy, sensitivity and specificity measurements were calculated in order to compare the results of the different decision procedures for each of the protein families tested. The sensitivity measures how well a binary classification test correctly identifies a condition. The specificity measures how well a binary classification test correctly identifies the negative cases, or those cases that do not meet the condition under study. The specificity, sensitivity, and accuracy measures were calculated by counting the number of true positives (tp), true negatives, (tn), false positives (fp) and false negatives (fn).

Test results. For FIG000087, the sensitivity of all three procedures is identical without errors present. As errors

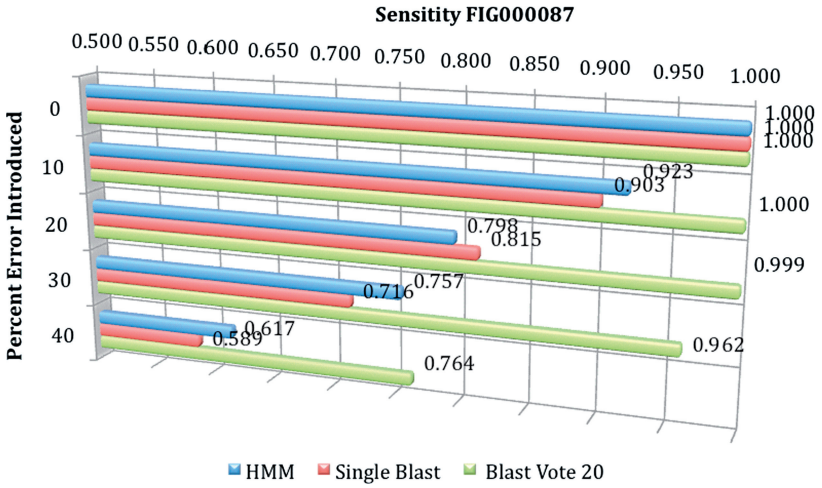
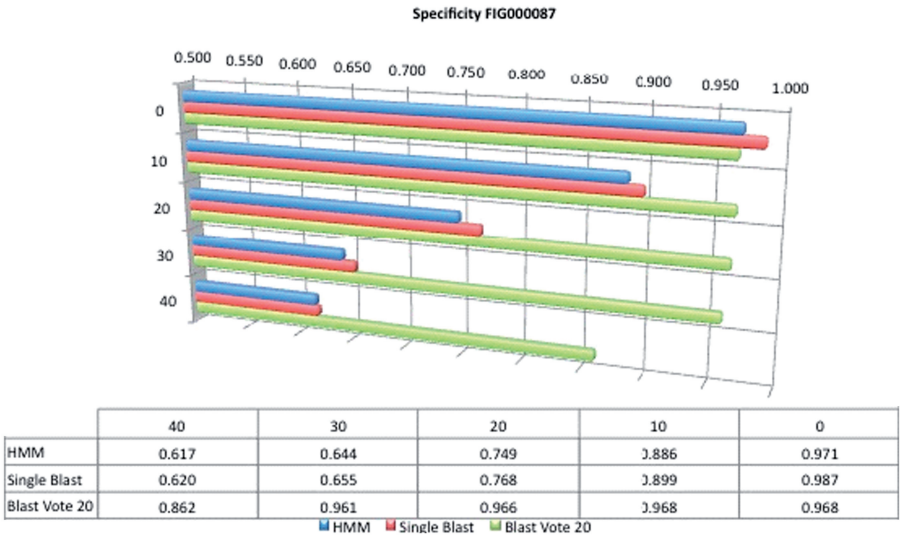


Figure 6. Sensitivity results in the presence of varying error rates for FIG000087. The BLAST voting procedure clearly outperforms the other procedures, losing virtually no sensitivity at 20% error.



	40	30	20	10	0
HMM	0.617	0.644	0.749	0.886	0.971
Single Blast	0.620	0.655	0.768	0.899	0.987
Blast Vote 20	0.862	0.961	0.966	0.968	0.968

Figure 7. The specificity for protein family FIG000087 in the presence of varying error rates. All three procedures perform equally well in the presence of no errors, but performance drastically drops once errors are introduced into the protein family data. The BLAST voting procedure clearly outperforms the other procedures with >0.96 specificity in the presence of 30% errors.

are introduced into the data set, the BLAST voting procedure clearly outperforms the other procedure with almost no loss of sensitivity at 20% errors and a 0.96 sensitivity rate at 30% errors (Figure 6).

The specificity, or rate of false positive predictions, is another important performance measure for a classification tool. Again the BLAST voting procedure clearly outperforms the simple BLAST and the HMM (Figure 7).

In some cases the decision procedure associated with a FIGfam is not the BLAST voting procedure. Instead we use the similarity bounds procedure described earlier. Figure 8 shows the performance characteristics of this procedure. Similarity bounds provide very good specificity (>0.914 for 50% errors in the data), but the sensitivity degrades badly with increasing error rates. The complete results are available in the appendix.

Discussion of test 2. The BLAST voting procedure clearly outperforms HMMs in the chosen example. Since we do not include this procedure with all FIGfams, however, we also show data for the similarity bounds procedure. This is a very conservative procedure with very poor sensitivity. The decision to use the similarity bounds procedure when faced with the potential of introducing false positives is one taken by the human curator of the FIGfams to minimize the noise introduced into the predictions.

Test 3: using HisAb as a gold standard for comparing FIGfam, TIGRFAM and PIR HMM assignments

The decision procedures from different groups such as PIRSF [1] and TIGRFAM [4] were tested to evaluate the

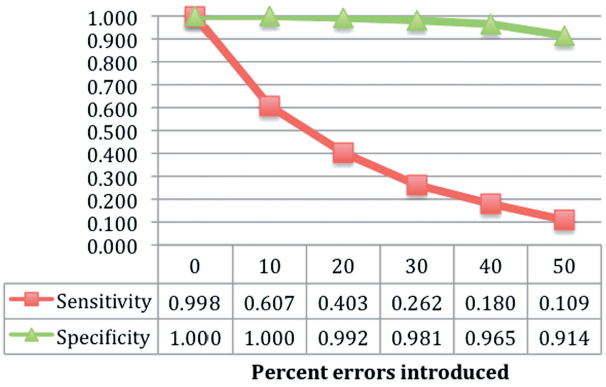


Figure 8. The similarity bounds procedure has very good specificity in the face of errors, but the sensitivity degrades rapidly in the presence of errors.

accuracy of the three protein families and the associated algorithms. We believe that in the case of *HisAb* we can use the annotations provided in ref. (25) as a gold standard.

Table 2 provides the number of protein families intersecting the contents of the *HisAb* set associated with each database. TIGRFAM release 7.0 provides the public with a set of protein families covering a range of functional roles. TIGRFAM's preferred decision procedure is a set of HMMs that provide a trusted and noise-cutoff score indicating the ranges for which the results can either be trusted or not as the specified functional role. The HMMER package was used to make an assignment. PIRSF (July 2007 release) also provides a set of HMMs along with a decision procedure for its protein families. The PIR decision procedure uses the available HMMs together with BLAST results to assign a sequence to a PIRSF group.

The FIGFAMs, TIGRFAMs and PIRSF protein families provide a number of related families that intersect with the contents of *HisAb*. The same *HisAb* sequences used in the previous section were used to compare the accuracy, sensitivity and specificity against the decision procedures of the three protein family groups. The main interest is to test whether the manually curated *HisAb* sequences were correctly characterized as any of the respective *HisAb* families in FIGfam, TIGRFAM, and PIRSF. A similar procedure was used to count the *tp*, *fp*, *tn* and *fn* as before. The same equations as in the above section were used to calculate the accuracy, sensitivity and specificity of the *HisAb* families. The specificity comparisons between the *HisAb* protein families from FIGfams, TIGRFAMS and PIRSF show that none tries to overpredict the sequence's functional role (keep false positives to a minimum). However, using the FIGFAM decision procedures resulted in more functional roles assigned correctly (sensitivity). A summary of the comparisons is shown in Figure 9.

Test 4: comparing runtime and coverage of FIGfams, TIGRFAMS and PIRSF

The use case for the protein families is in annotating novel sequences; here we study the percentage of proteins in five

Table 2. *HisAb* protein families from FIGfam, TIGRFAM and PIRSF

Protein family group	Name	Description
FIGfam	FIG000087	Histidyl-tRNA synthetase (EC 6.1.1.21)
	FIG000865	ATP phosphoribosyltransferase regulatory subunit (EC 2.4.2.17)
TIGRFAM	hisS	histidyl-tRNA synthetase
	hisS_second	ATP phosphoribosyltransferase, regulatory subunit
PIRSF	PIRSF001549	histidyl-tRNA synthetase (validated)
	PIRSF006650	ATP phosphoribosyltransferase
	PIRSF000486	ATP phosphoribosyltransferase

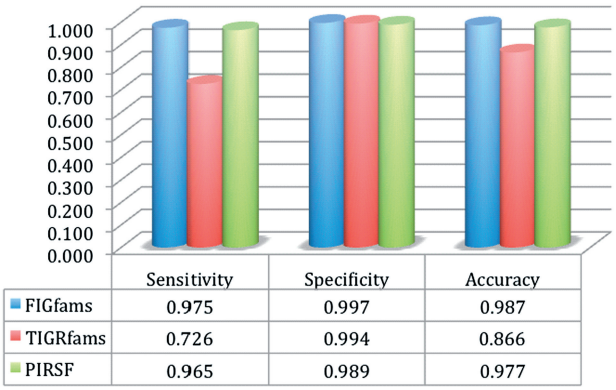


Figure 9. Comparisons of FIGfams, TIGRFAMS and PIRSF *HisAb* protein families. The high specificity levels on each protein group indicate that no group tries to overpredict the functional roles of the sequences. Both FIGfams and PIRSF performed well in annotating the correct function. The TIGRFAM predictions used the HMM trusted and noise-cutoff scores to assign the functions. The top hit of most TIGRFAM's non-assigned sequences was one of the *HisAb* families; however, TIGRFAM failed to annotate it as a member of the *HisAb* family because it missed the noise-cutoff score.

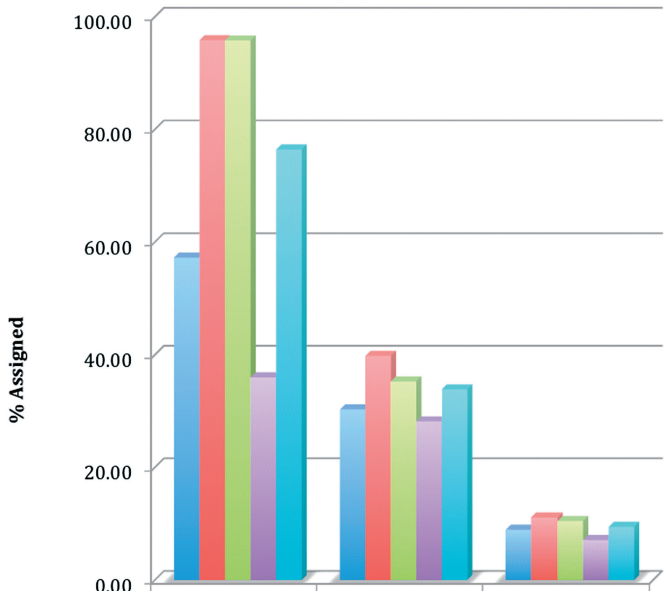
Table 3. List of genomes analyzed

Genome	Number of proteins
Bacillus subtilis subsp. subtilis str. 168	4105
Escherichia coli K12	4133
Staphylococcus aureus subsp. aureus COL	2618
Synechocystis sp. PCC 6803	3572
Vibrio cholera cholerae O1 biovar eltor str. N16961	3835

complete microbial genome sequences (Table 3) that were automatically assigned a function and the time require to compute the annotations.

For FIGfams, we used the built-in method, in this case the BLAST voting procedure; for TIGRfams, we used the HMMs provided with the cut-off values; and for PIRSF, we used the decision procedure provided by PIR.

The most interesting aspect of the comparison for this test is the vast difference in the number of proteins assigned by the different technologies to protein families as indicated in Figure 10. The difference in coverage is at



	FIGfams	TIGRFams	PIRSF
Bacillus subtilis subsp. subtilis str. 168	57.25	30.28	8.92
Escherichia coli K12	95.84	39.80	11.08
Staphylococcus aureus subsp. aureus COL	95.80	35.26	10.47
Synechocystis sp. PCC 6803	35.97	28.16	7.08
Vibrio cholera cholerae O1 biovar eltor str. N16961	76.43	33.82	9.41

Figure 10. Percentage of proteins covered by FIGfams, TIGRFAMS and PIRSF using the appropriate decision procedures for five genomes.

least 10-fold and in some cases 20-fold, resulting from the larger number of FIGfams. Also taking into account the runtimes for the three decision procedures (Figure 11), we see that the FIGfam decision procedures are 10- to 30-fold faster than the existing procedures.

Because of the lack of a gold standard, the comparisons here are not for the actual correctness of the annotations; instead we are comparing the fractions of proteins annotated with the protein families and the runtime required. We provide a complete list of all assignments made in the appendix.

SUMMARY AND DISCUSSION

The propagation of errors from the sequence databases has been a significant problem in genome annotation and other areas. Several techniques have been used to handle the results of the noise in the databases. We present a novel solution to the problem by providing a set of protein families that can be used for automatic annotation based on a set of consistent, manually derived, high-quality annotations. The fact that Subsystems cover 50% of the known bacterial and archaeal protein space makes FIGfams a very useful resource. By allowing for variable decision procedures on a per family basis, we have ensured rapid processing at a rate that enables the annotation of several genomes per day on a current desktop machine.

The primary benefits of our approach are as follows:

- FIGfams are fast, reliable, and robust to noise in the data. Moreover, as more diverse genomes are sequenced and annotated, the speed and accuracy of FIGfam-based annotation will increase.
- The time to classify a single protein averages around 10 s on a modest desktop machine, allowing processing of ~8640 proteins per day on a single machine.
- In the examples shown in this manuscript and our other tests, the BLAST voting procedure, the most frequently used decision procedure for FIGfams, performs at least as well as simple BLAST and HMM-based procedures for the propagating the annotations of conserved proteins (test 1), or distinguishing between two closely related proteins (test 2). If errors are present in the data set, it outperforms the other procedures in the examples we tested (test 3).
- FIGfam performance is optimized to minimize false positive assignments.
- As Subsystems cover more and more of the known protein space, the FIGfams will increase in value over time.
- New results from the literature are incorporated into the FIGfams via Subsystem curation, guaranteeing that the FIGfams remain up to date.

As the number of proteins in FIGfam increases, automatic annotation pipelines such as RAST (26) will be able

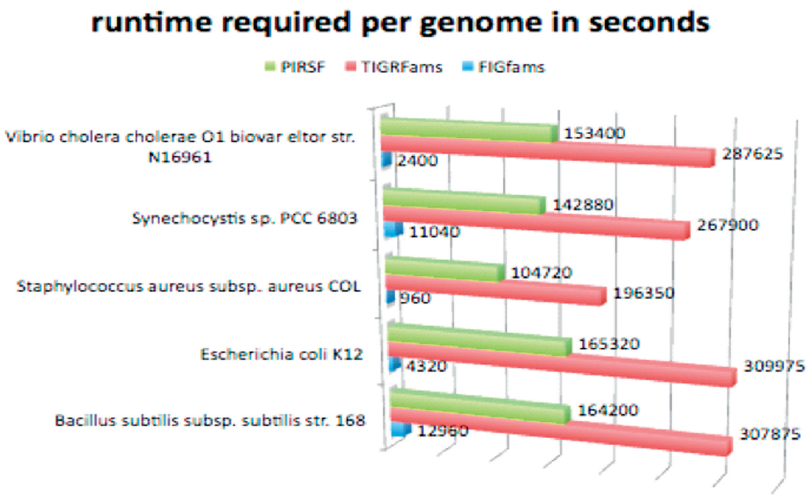


Figure 11. Runtime requirements (in seconds) for annotating a genome with the decision procedure associated with the protein families.

to reduce the number of genes subjected to costly in-depth database searches. Thus, by linking accurate, mass creation of protein annotations and protein family construction using Subsystems we have achieved a novel approach offering both high productivity and high accuracy in protein family creation.

AVAILABILITY

The FIGfams have been used as a central component in the RAST server (27) (http://RAST.nmpdr.org), a system that provides rapid, accurate annotation of prokaryotic genomes. They are also used in MG-RAST (28), a public server focusing on the annotation of metagenomic data (http://metagenomics.nmpdr.org).

Release 10 of FIGfams is made freely available to anyone for any use. It contains 1 414 035 proteins grouped into 106 775 families. All families can be downloaded from ftp.theseed.org/FIGfams/.

Running the FIGfam decision procedure locally requires a Linux/Unix/OS-X operating system and Perl 5.6 or greater.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

The authors wish to thank Gail Pieper for helping with editing the manuscript. We thank Veronika Vonstein and her team of expert Subsystem annotators.

FUNDING

Part of this project has been funded with federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Contract No. HHSN266200400042C. Argonne National Laboratory's work was supported under US Department of Energy contract DE-AC02-06CH11357. Funding for open access charge: US Department of Energy contract DE-AC02-06CH11357.

Conflict of interest statement. None declared.

REFERENCES

1. Benson,D.A., Karsch-Mizrachi,I., Lipman,D.J., Ostell,J. and Wheeler,D.L. (2007) GenBank. *Nucleic Acids Res.*, **35**, D21–D25.
2. Margulies,M., Egholm,M., Altman,W.E., Attiya,S., Bader,J.S., Bemben,L.A., Berka,J., Braverman,M.S., Chen,Y.J., Chen,Z. *et al.* (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**, 376–380.
3. Overbeek,R., Bartels,D., Vonstein,V. and Meyer,F. (2007) Annotation of bacterial and archaeal genomes: improving accuracy and consistency. *Chem. Rev.*, **107**, 3431–3447, [Epub 21 July, 2007].
4. Pruitt,K.D., Tatusova,T. and Maglott,D.R. (2007) NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res.*, **35**, D61–D65.
5. McNeil,L.K., Reich,C., Aziz,R.K., Bartels,D., Cohoon,M., Disz,T., Edwards,R.A., Gerdes,S., Hwang,K., Kubal,M. *et al.* (2007) The national microbial pathogen database resource (NMPDR): a genomics platform based on subsystem annotation. *Nucleic Acids Res.*, **35**, D347–D353.
6. Kanehisa,M., Araki,M., Goto,S., Hattori,M., Hirakawa,M., Itoh,M., Katayama,T., Kawashima,S., Okuda,S., Tokimatsu,T. *et al.* (2008) KEGG for linking genomes to life and the environment. *Nucleic Acids Res.*, **36**, D480–D484.
7. Selengut,J.D., Haft,D.H., Davidsen,T., Ganapathy,A., Gwinn-Giglio,M., Nelson,W.C., Richter,A.R. and White,O. (2007) TIGRFAMs and genome properties: tools for the assignment of molecular function and biological process in prokaryotic genomes. *Nucleic Acids Res.*, **35**, D260–D264.
8. Wu,C.H., Nikolskaya,A., Huang,H., Yeh,L.S., Natale,D.A., Vinayaka,C.R., Hu,Z.Z., Mazumder,R., Kumar,S., Kourtesis,P. *et al.* (2004) PIRSF: family classification system at the protein information resource. *Nucleic Acids Res.*, **32**, D112–D114.
9. Tatusov,R.L., Fedorova,N.D., Jackson,J.D., Jacobs,A.R., Kiryutin,B., Koonin,E.V., Krylov,D.M., Mazumder,R., Mekhedov,S.L., Nikolskaya,A.N. *et al.* (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41.

10. Li,L., Stoeckert,C.J. Jr. and Roos,D.S. (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res.*, **13**, 2178–2189.
11. Pearl,F.M., Bennett,C.F., Bray,J.E., Harrison,A.P., Martin,N., Shepherd,A., Sillitoe,I., Thornton,J. and Orengo,C.A. (2003) The CATH database: an extended protein family resource for structural and functional genomics. *Nucleic Acids Res.*, **31**, 452–455.
12. Andreeva,A., Howorth,D., Chandonia,J.M., Brenner,S.E., Hubbard,T.J., Chothia,C. and Murzin,A.G. (2008) Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res.*, **36**, D419–D425.
13. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
14. Bateman,A., Coin,L., Durbin,R., Finn,R.D., Hollich,V., Griffiths-Jones,S., Khanna,A., Marshall,M., Moxon,S., Sonnhammer,E.L. *et al.* (2004) The Pfam protein families database. *Nucleic Acids Res.*, **32**, D138–D141.
15. Mi,H., Lazareva-Ulitsky,B., Loo,R., Kejariwal,A., Vandergriff,J., Rabkin,S., Guo,N., Muruganujan,A., Doremieux,O., Campbell,M.J. *et al.* (2005) The PANTHER database of protein families, subfamilies, functions and pathways. *Nucleic Acids Res.*, **33**, D284–D288.
16. Andreeva,A., Howorth,D., Brenner,S.E., Hubbard,T.J., Chothia,C. and Murzin,A.G. (2004) SCOP database in 2004: refinements integrate structure and sequence family data. *Nucleic Acids Res.*, **32**, D226–D229.
17. Marchler-Bauer,A., Anderson,J.B., Cherukuri,P.F., DeWeese-Scott,C., Geer,L.Y., Gwadz,M., He,S., Hurwitz,D.I., Jackson,J.D., Ke,Z. *et al.* (2005) CDD: a Conserved Domain Database for protein classification. *Nucleic Acids Res.*, **33**, D192–D196.
18. Mulder,N.J., Apweiler,R., Attwood,T.K., Bairoch,A., Bateman,A., Binns,D., Bork,P., Buillard,V., Cerutti,L., Copley,R. *et al.* (2007) New developments in the InterPro database. *Nucleic Acids Res.*, **35**, D224–D228.
19. Schneider,M., Tognolli,M. and Bairoch,A. (2004) The Swiss-Prot protein knowledgebase and ExPASy: providing the plant community with high quality proteomic data and tools. *Plant Physiol. Biochem.*, **42**, 1013–1021.
20. Chen,F., Mackey,A.J., Stoeckert,C.J. Jr and Roos,D.S. (2006) OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Res.*, **34**, D363–D368.
21. Haft,D.H., Selengut,J.D. and White,O. (2003) The TIGRFAMs database of protein families. *Nucleic Acids Res.*, **31**, 371–373.
22. Overbeek,R., Begley,T., Butler,R.M., Choudhuri,J.V., Chuang,H.Y., Cohoon,M., de Crecy-Lagard,V., Diaz,N., Disz,T., Edwards,R. *et al.* (2005) The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes. *Nucleic Acids Res.*, **33**, 5691–5702.
23. Jensen,R.A. (2001) Orthologs and paralogs—we need to get it right. *Genome Biology*, **2**, INTERACTIONS1002.
24. Sissler,M., Delorme,C., Bond,J., Ehrlich,S.D., Renault,P. and Francklyn,C. (1999) An aminoacyl-tRNA synthetase paralog with a catalytic role in histidine biosynthesis. *Proc. Natl Acad. Sci. USA*, **96**, 8985–8990.
25. Vega,M.C., Zou,P., Fernandez,F.J., Murphy,G.E., Sterner,R., Popov,A. and Wilmanns,M. (2005) Regulation of the hetero-octameric ATP phosphoribosyl transferase complex from Thermotoga maritima by a tRNA synthetase-like subunit. *Mol. Microbiol.*, **55**, 675–686.
26. Aziz,R.K., Bartels,D., Best,A.A., DeJongh,M., Disz,T., Edwards,R.A., Formsma,K., Gerdes,S., Glass,E.M., Kubal,M. *et al.* (2008) The RAST server: rapid annotations using Subsystems technology. *BMC Genomics*, **19**, 386.
27. Aziz,R.K., Bartels,D., Best,A.A., DeJongh,M., Disz,T., Edwards,R.A., Formsma,K., Gerdes,S., Glass,E.M., Kubal,M. *et al.* (2008) The RAST Server: rapid annotations using subsystems technology. *BMC Genomics*, **9**, 75.
28. Meyer,F., Paarmann,D., D'Souza,M., Olson,R., Glass,E.M., Kubal,M., Paczian,T., Rodriguez,A., Stevens,R., Wilke,A. *et al.* (2008) The metagenomics RAST server—a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, **9**, 386.