

PATRIC Standard Operating Procedures for Genomic Sequence Annotation Pipeline (GSAP)

Author: Eric E. Snyder
Version: 1.1, October 2, 2006 9:32 AM
Supersedes version: 1.0, August 18, 2006 5:02 PM

Purpose

This SOP describes the Genomic Sequence Annotation Pipeline (GSAP), an automated procedures for DNA-level annotation of microbial genomes. The purpose of GSAP is simply to execute standard sequence analysis programs; no interpretation of the output of these programs is done by this pipeline. It should be seen as the first step in a larger pipeline that also includes automated and manual curation procedures. preparation for manual follow-up by the project's curation staff. The goal of this document is to familiarize personnel with the configuration, execution and output of GSAP.

Application

Although this SOP is narrowly focused on its application to the NIAID/BRC PATRIC project, it will be helpful to other groups interested in using the Curation Infrastructure developed by the Genomics Domain Area Team at VBI. The system is flexible and can be readily adapted to analyze the genomes of other organisms, or other types of sequence data.

Participants

This SOP is typically performed by a software developer working with a bioinformatician to define the analyses to be executed by the pipeline. This information is captured in a "script" (a GAPML document) which can be used to process a wide variety of genomes from other organisms (assuming the appropriate tools and parameters are chosen).

Inputs

To execute the processes described herein, access to a number of databases and applications is presumed. These include:

1. GAP application (described in section 1, below)
2. GAPML template and/or GAPML documentation [8] (also described in section 1)
3. Instance of Curation Infrastructure (CI) Database ("PATRIC")
4. CI Website connected to CI Database

Entry Criteria

In order to analyze a genome using GSAP, the sequence of interest (in GenBank format, preferably with legacy annotation) must have been loaded into the CI database using the GBparser.pl plugin. The genome must be assigned to an organism category and be identified as a "reference" or "associated" genome.

Steps

As described in detail in sections 1 - 2, the GSAP GAPML document must be appropriately configured. Then, the GAP application is run from a UNIX command line with the corresponding GAPML file supplied as an argument.

Outputs

As GAP is running, the following diagnostic information is printed:

1. Accession number of input sequence
2. Tool currently executing

3. Availability of web service being called
4. Return status of web service (whether it returned a valid output)
5. Return status of data-loading plugin (whether the data was successfully uploaded to database)

This data from STDOUT and STDERR are directed to files for post-GAP analysis.

In addition, a log file is created which logs the sequence and the start and end times for the execution of each tool plus elapsed time for each input sequence.

Exit Criteria

Following pipeline execution, the error logs are scanned for failed sequences or tools. In the event of such a failure, the process is repeated for the remaining data until the complete dataset has been analyzed.

Performance Measures

After each GSAP run, queries are executed to gather data for data integrity checking. For example, all predicted CDSs should have length%3 = 0. Histograms of CDS feature length are plotted to identify outliers for manual inspection. In addition, queries are run to compare predicted CDSs to legacy annotation (GenBank or RefSeq) and to each other. This allows curators to assess the quality of previous annotation with respect to GSAP and generate a baseline against which to compare the results of manual annotation. The performance of individual tools can also be assessed to aid in their interpretation.

Abstract

The Genomic Sequence Analysis Pipeline (GSAP) is the first step in the PATRIC curation process. The pipeline executes a pre-defined series of sequence analysis programs on the genome of interest and generates a collection of predicted features aimed at identifying the genes encoded therein. The features can be used by curators directly or after processing by an automated curation system.

1. GAP and GAPML

Pipeline services in the CyberInfrastructure Group are constructed using a Java-based, domain-independent pipeline application known as GAP (Generic Analysis Pipeline). Specific pipeline instances are defined using GAPML (Generic Analysis Pipeline Markup Language), an XML-based pipeline description language developed for use with GAP. (For a complete description of the use of GAP and GAPML, please see: [8].) Each GAPML document begins with the specification of a source of input data for the pipeline. Data can be read from a file or directly from the PATRIC database. In the latter instance, the specification contains connection information (URL, database name, username, password) and SQL query. Next follows a series of specifications for executing analytical applications. These applications can be run locally via a shell but more commonly involve web service calls. In either case, parameter names and values are defined which will be passed to the program when the pipeline is executed. Finally, each tool specification is followed by a corresponding directive instructing the pipeline program how to process and upload the tool's output into the database.

2. GSAP Configurations

GSAP is a type of GAPML document that defines the programs to be executed on a genome sequence. Each GSAP pipeline is configured according to the nature of the input genome and the specific analyses that are to be performed. There are four types of genomes in the PATRIC project, each of which requires a different GSAP configuration. From a biological standpoint, the bacterial and viral genomes differ dramatically in size and in the types of expected features. From the perspective of annotation strategy, two additional genome types are defined. Each organism category contains one or more *reference genomes* (RGs). These genomes are selected in collaboration with community experts to be the best characterized and/or most representative members of the category. The remaining members are referred to as *associated genomes* (AGs). Consequently, there are four general types of genomes in this project and therefore four corresponding GSAP configurations, as illustrated in Table 1.

Table 1: GSAP Configuration Types

Genome Type	Reference	Associated
Bacterial	B-RG	B-AG
Viral	V-RG	V-AG

The functional differences between these pipelines are discussed in sections 4 and 5.

3. Applications Used in Pipelines

Table 4 shows a complete list of GSAP applications as a function of the configuration shown in Table 1.

4. Pipeline Analysis of Reference Genomes

The primary difference between the GSAP used to annotate a RG and that used for an AG lies in the way that genes are predicted. The RG is annotated first and thus equal weight is given to *ab initio* gene predictions, alignments to sequences from organisms outside the RG's organism category and to primary GenBank annotation or RefSeq annotation from NCBI. The sequence features generated by the pipeline are interpreted by the curators who then edit the gene coordinates (if necessary) using the Gene Edit Page on the PATRIC web site and indicate the evidence used in their decision making (provenance). The "finalized" genes from the RG are then translated to create the first version of the Reference Protein List (RPL) for the organism category. The RPL will be used in subsequent pipelines to identify genes from AGs, as described in section 5.

4.1 Bacterial Genomes

Given a new collection of organisms, the reference genomes are analyzed first, using a GSAP configuration that emphasizes the generation of sequence features that will be interpreted by curators to make "curated" gene predictions. Table 4 shows the applications run by GSAP and their execution order (column "RG") for bacterial reference genomes, GSAP B-RG, using the nomenclature in Table 1. The gene prediction programs are executed first, followed by the "start site correction" programs TICO and RBSfinder. Taken with the primary annotation, this generates the feature types shown in Table 2.

Table 2: GSAP B-RG Features

	Feature Derivation	Type
1	Original GenBank entry or NCBI RefSeq project	various
2	Glimmer	CDS
3	Glimmer + TICO	CDS
4	Glimmer + RBSfinder	CDS
5	GeneMark	CDS
6	GeneMark + TICO	CDS
7	GeneMark + RBSfinder	CDS
8	tRNAscan-SE	Gene
9	rRNAscan	Gene

Steps 1 and 2 of GSAP B-RG produce 9 types of gene-related features. The next step is to identify putative gene intervals (PGIs) using a program call `pgi.pl`. The object of the program is to partition the genome sequence into segments such that each segment contains a single gene, subject to the condition that genes (excluding stop codons) do not overlap and that longer genes are preferred over shorter ones. These segments plus 200 bp on either end are used as input to a BLASTX search against the NCBI non-redundant protein database. While some groups prefer to simply align gene predictions or ORFs, this

approach ensures that the entire genome is covered in a tiling of aligned segments that include approximately one gene each, reducing the probability that alignments to one gene in a segment will “swamp” out alignments to an adjacent gene in the same segment.

4.2 Viral Genomes

The PATRIC viruses are small RNA viruses that infect mammalian hosts. The Coronaviruses have the largest genomes at ~30 kb; the remaining viruses are in the 7 – 9 kb range. Consequently, GSAP for viral genomes is much simpler than their bacterial counterparts, using a more limited set of analytical tools and no PGI identification step. Their mammalian host specificity requires that the pipeline to use different versions of Glimmer and GeneMark, optimized for performance on human genes. Since TICO and RBSfinder search for signals peculiar to prokaryotic genes, these tools are not used. While ribosomal and transfer RNAs are found in a handful of viruses, they have not been observed in the small genomes of the PATRIC viruses. Consequently, tRNAscan-SE and rRNAscan are not run on these genomes.

5. Pipeline Analysis of Associated Genomes

Within a given organism category, analysis of AGs begins only after the RG(s) have been curated. This allows creation of the Reference Protein List (RPL), a major asset that facilitates and increases the accuracy of AG curation. Because the organisms within a category are so closely related, it is relatively straightforward to find the orthologs of genes identified during RG curation. After this initial step (described in more detail below), the remaining applications in GSAP are run to identify additional features, including novel genes (as described in Table 4).

5.1 Bacterial Genomes

For bacterial AGs, a high-throughput approach is taken to gene identification and concomitant identification of orthologs. Each reference protein is aligned to a 6-frame translation of the associated genome using TBLASTN. The highest scoring HSP (by E-value) is then checked against the inclusion criteria shown in Table 3. If the inclusion criteria are met, a “reference genome alignment” feature is created.

Table 3: Parameters for Ortholog Identification in Bacteria

Parameter	Cutoff	Rationale
E	< 10 ⁻²⁰	Retain only significant alignments
Coverage, % of query	> 80%	Ensure alignment covers whole protein, not domain(s) in isolation
AA Similarity, as defined by BLAST	> 95%	Minimize evolutionary divergence

5.2 Viral Genomes

In contrast to the methods described thus far, which all rely on applications being executed using GAP, the curation of viral AGs uses GATU [10] (Genome Annotation Transfer Utility). This Java application, which is executed outside of the PATRIC curation infrastructure, was built for ortholog identification and annotation transfer between small viral genomes, using BLAST and Smith-Waterman [7]. When the job is complete, results are dumped in a standard format and loaded into GUS using custom plugin.

6. Conclusion and Follow-Up

After the completion of GSAP execution, the CI database will have been populated with a new set of features produced by the pipeline. At this point, basic data integrity checks can be run to ensure, for example, that all CDS features contain no stop codons and have length a multiple of three. Statistics can also be generated by ad hoc queries to create length distributions of the various feature types to help identify outliers and pathological cases. Once any problems have been identified and corrected, the genome's DNA features should be submitted to the next SOP, automated DNA-level curation (ADC).

7. References

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. Basic local alignment search tool. *J Mol Biol* **215**, 403-10 (1990).
2. Azad, R.K. & Borodovsky, M. Probabilistic methods of identifying genes in prokaryotic genomes: connections to the HMM theory. *Brief Bioinform* **5**, 118-30 (2004).
3. Delcher, A.L., Harmon, D., Kasif, S., White, O. & Salzberg, S.L. Improved microbial gene identification with GLIMMER. *Nucleic Acids Res* **27**, 4636-41 (1999).
4. Gribskov, M., Devereux, J. & Burgess, R.R. The codon preference plot: graphic analysis of protein coding sequences and prediction of gene expression. *Nucleic Acids Res* **12**, 539-49 (1984).
5. Lowe, T.M. & Eddy, S.R. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res* **25**, 955-64 (1997).
6. Lukashin, A.V. & Borodovsky, M. GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res* **26**, 1107-15 (1998).
7. Smith, T.F. & Waterman, M.S. Identification of common molecular subsequences. *J Mol Biol* **147**, 195-7 (1981).
8. Soneja, J. GAPML Documentation. (Blacksburg, 2005).
9. Suzek, B.E., Ermolaeva, M.D., Schreiber, M. & Salzberg, S.L. A probabilistic method for identifying start codons in bacterial genomes. *Bioinformatics* **17**, 1123-30 (2001).
10. Tcherepanov, V.T., Ehlers, A. & Upton, C. Genome Annotation Transfer Utility (GATU): Rapid annotation of viral genomes using a closely related reference genome. *BMC Genomics* **7**, 150 (2006).
11. Tech, M., Pfeifer, N., Morgenstern, B. & Meinicke, P. TICO: a tool for improving predictions of prokaryotic translation initiation sites. *Bioinformatics* **21**, 3568-9 (2005).

Table 4: Applications and Data Used in DNA-Level Genome Curation + GSAP Order of Execution for Different Genome Types

	Program	Parameter Descriptions	Flags + Values	Database	Data/Task Description	Order of Execution				Notes
						Bacterial		Viral		
						RG	AG	RG	AG	
1	(1 ^o annotation)			GenBank / RefSeq	Primary annotation from current GenBank record or NCBI RefSeq project, when available (preferred)					1
2	TBLASTN [1]	expectation value one line DB descriptions to print one line DB alignments to print turn off query complexity filter show GI no. in definition line	-e 10 ⁻²⁰ -v 5 -b 5 -F F -I T	Reference Protein List	CDS prediction and ortholog identification		1		1	2
3	Glimmer [3]	use 1 st codon as start min overlap length min overlap percentage use independent prob scores gene threshold score use weak scores on long genes	+f -o 30 -p 10 +r -t 90 -w 935		CDS prediction	1	2			3
	long-orfs (used for glimmer training)	max overlap length max overlap percentage	-o 30 -p 10							
4	GeneMarkHMMp [2,6]	use 1 st codon as start train on primary annotations		-	CDS prediction	1	2			4
5	GeneMarkHMMp	use 1 st codon as start use heuristic model based on [G+C]		-	CDS prediction			1	2	5
6	tRNAscan-SE [5]	search for bacterial tRNAs	-B	tRNA HMMs	tRNA gene prediction	1	2			
7	rRNAscan (BLASTN)	expectation value	-e 10 ⁻⁵	rRNA	Ribosomal RNA gene prediction	1	2			6
8	TICO [11]	upstream search window downstream search window	-su 250 -sd 250	-	Start site correction	2	3			

		upstream extracted window downstream extracted window minimum gene length smoothing parameter ROC-flag	-exu 30 -exd 30 -minlength 60 -sig 0.5 -roc 1							
9	RBSFinder [9]			-	Start site correction, ribosome binding site prediction	2	3			
10	PGI.pl	give preference to 1 ^o annotation multiplier for preferred annotation	-p RefSeq -m 1000		BLAST pre-processor	3	4	2	3	7
11	BLASTX [1]	expectation value one line DB descriptions to print one line DB alignments to print turn off query complexity filter show GI no. in definition line substitution matrix	-e 10 ⁻⁵ -v 5 -b 5 -F F -I T -M BLOSUM45	NRAA	Protein similarity search	4	5	3	4	8
13	Start site consensus				Identify putative translational start sites using PWM					9
14	Codon usage				Generate scanning window plot of codon usage					10
15	Codon bias [4]				Generate scanning window plot of codon bias					11

1. These annotations come from the source sequences (usually GenBank or RefSeq records) rather than an application; however, the CDS features are treated like other gene predictions when processed by the "putative gene interval" identification program (pgi.pl).
2. After curation of the reference genome(s) for a particular organism category, the CDS features are translated to create the Reference Protein List. When annotating associated genomes, this step is used to identify putative orthologs. Each reference protein is aligned to a 6-frame translation of the associated genome using TBLASTN. The highest scoring HSP (by E-value) is then checked against the inclusion criteria, typically 90% amino acid-level similarity (as defined by BLAST) extending over at least 85% of the length of the query sequence. If the inclusion criteria are met, the coordinates of the alignment on the associated genome are processed further to identify valid start and stop codons in the vicinity. The 5'-end is identified using either TICO or a log-likelihood position-weight matrix for the start codon consensus for the organism in question. The weight matrix is scanned upstream of the alignment to find the best start site that maintains the reading frame (within 500 bp). If no site is found, scanning continues 3' of the alignment's 5'-end until a valid site is found (matrix score > 0). A similar scanning approach is used to identify the stop codon associated with the alignment. If valid termini cannot be found, the alignment is targeted for inspection by the curation staff.
3. For genomes with multiple chromosomes, Glimmer is trained on all of them, although the program is executed on individual chromosomes.

4. For bacterial genomes, GeneMark is trained using the organism category's reference genome. When the reference genome is analyzed, GeneMark is trained using RefSeq annotations. For other genomes in the same category, GeneMark is trained using the reference genome and CDSs based on PATRIC's manual curation.
5. Due to the small size of our viral genomes, GeneMarkHMM is trained using "heuristic models" based on the G+C content of the genome being analyzed.
6. rRNAscan implements a stringent BLASTN search against a database of known prokaryotic ribosomal RNA species.
7. PGI.pl (Eric E. Snyder, unpublished) uses a dynamic programming algorithm to segment the genome into fragments based on gene predictions and primary (GenBank) annotation for the purpose of follow-up similarity searches using BLASTX. We find this method superior to simply BLASTing ORFs because adjacent presumptive non-coding sequence is also included. This increases the probability of accurate gene identification, particularly in the presence of sequencing error. Because of the small size of the viral genomes in the project, this step is not required for viral annotation.
8. TimeLogic BLAST is an implementation of the BLAST algorithm for use with TimeLogic's peristaltic array hardware.
9. The start site consensus (SSC) score at each position in the sequence is calculated using a log-likelihood position-weight matrix (PWM) scoring program (Eric E. Snyder, unpublished). The matrices represent 3 bp on either side of the start codon (for a total of 9 positions) and were calculated from occurrence matrices compiled from the primary annotation of each (bacterial) RG and applied to all of the genomes in their respective categories. Given the small size of the viral genomes, a PWM based on a compilation of human start sites was used for all viral categories. Technically, the SSC program is not executed by GSAP; rather, the program is executed on the fly by the visualization component of the Gene Edit Page (GEP), where the curators do their work.
10. As with the SSC, codon usage and bias plots are not part of GSAP; they are calculated on the fly by the GEP visualization. Codon usage statistics are compiled based on the CDS features from the primary annotation of the RG for each bacterial organism category. For each codon, a log-likelihood ratio (LLR) score is calculated from the natural log of the ratio of observed codon frequency to the expected codon frequency based on the organism's overall base composition (G+C content). The codon usage plot is calculated for the 3 forward and 3 reverse reading frames using a scanning window approach. For each frame, the sum of the LLR for each codon in the window is taken and normalized by dividing by the number of codons in the window. Thus the score of the window is proportional to the likelihood that the sequence interval (and frame) is coding.
11. The codon bias (also known as codon preference) plot is calculated in a manner analogous to the codon usage plot using the methods of Gribskov, *et al.*[4].