Fungal Genome Annotation Standard Operating Procedure (SOP)

Introduction

The JGI annotation process for fungal genomes uses an automated annotation pipeline, a set of quality control metrics manually inspected by annotators, and community curation of predicted genes and annotations. Annotation includes both structural and functional characterization of protein-coding and non-coding genes and other genomic features stored in JGI genome databases and accessible through MycoCosm (www.jgi.doe.gov/fungi), a fungal genomics resource for comparative analysis and community annotation.

Requirements

The JGI annotation pipeline takes fasta files of a genome assembly, ESTs and EST assemblies, several optional inputs (e.g., curated repeat libraries, gene predictor parameters, custom-built gene models, a list of organisms for comparison etc), configuration parameters and tags to customize the annotation pipeline if necessary. The pipeline produces annotated gene models and other genomic features that are deposited in JGI genomic databases. Web-based visualization and analysis tools connected to these databases and pipeline execution log files allow real-time monitoring and control of both the annotation process and the data it generates. Data- and user-management tools facilitate the data release process and data distribution to individual users.

Procedure

The key steps of the annotation pipeline include gene prediction, functional annotation, and comparative analysis.

1. Gene Prediction

Gene prediction is a critical step in the annotation of eukaryotic genomes because of their complex gene structure and genome organization and involves several stages: repeat-masking, gene prediction assisted with ESTs and homologs from other fungi using different prediction methods, and validation of predicted gene models with several lines of evidence.

Repeat-masking. Before gene prediction, assembly scaffolds are masked using RepeatMasker [Smit et al. 2010] and a genome-specific library of repeats composed of the standard RepBase library[Jurka et al. 2005], most frequent (>150times) repeats recognized by RepeatScout [Price et al, 2005], and manually curated libraries of transposons when available.

Mapping ESTs and proteins. All ESTs for a given organism, either sequenced in-house or retrieved from GenBank or collaborator collections, are inspected for quality, trimmed, clustered, and assembled into consensus sequences using sequencing platform-specific EST clustering pipelines (external to this pipeline). These ESTs and consensus sequences are mapped to the assembly using BLAT, filtered using thresholds of 95% nucleotide identity and 80% coverage over EST length [Kent, 2000] and used in gene modeling, model selection and validation processes.

Proteins from publicly available fungal genomes are grouped taxonomically and blasted against masked genome assembly using BLASTx with e-value 1e-5. These alignments serve as seeds for homology-based gene predictors.

Gene predictors. Using the repeat-masked assembly, several gene prediction programs falling into three general categories are used: 1) *ab initio* - FGENESH [Salamov and Solovyev 2000]; GeneMark [Ter-Hovhannisyan et al, 2008] trained for given genome, 2) *homology-based* - FGENESH+[Salamov and Solovyev 2000]; Genewise [Birney et al, 2004] seeded by BLASTx alignments against GenBank's database of non-redundant proteins (NR: http://www.ncbi.nlm.nih.gov/BLAST/), and 3) *EST-based* - EST_map (http://www.softberry.com/) seeded by EST contigs.

Training gene predictors. This process employs an automatically generated set of full-length genes derived from EST clusters and protein-homology models produced by Genewise and Fgenesh+ and screened for completeness, quality and redundancy. This set is randomly split into training and test subsets in proportion 4:1. The genes from the training subset provide hexamer frequencies derived from CDS while intron structure informs exon/intron transition probabilities of the Fgenesh parameter file. These newly derived parameters are tested on the test subtest in parallel to the parameters created earlier for other fungi, and the best performing parameter set assessed by specificity and sensitivity of exon predictions is used for given genome. If either specificity or sensitivity of best prediction drops below 50%, the process requires expert training. In addition, we use the self-training version of GeneMark software, which also captures intron structure specific to fungal genomes.

Improving gene models. Since all gene predictors predict only coding parts of genes (CDS), we use the program estExt (I. Grigoriev, unpublished) that employs ESTs and EST clusters overlapping with predicted gene models to extend them into coding or untranslated regions (UTR) and to correct their gene structures if they disagree with EST splicing patterns. GeneWise models not supported with ESTs are extended by finding in frame upstream start and downstream stop codons in assembly sequences. GeneWise models that include frameshifts are labeled as potential pseudogenes.

Filtering gene models. The annotation pipeline produces multiple overlapping gene models by different gene predictors at each locus. All models having significant similarity to repeat-type proteins and repeat Pfam domains are excluded from further analysis. To select the best representative gene model we employ a heuristic approach (A. Salamov, unpublished) based on a combination of protein homology and EST support. Homology information is based on alignments with the best BLASTp hit from protein databases, where only alignments with BLASTp score > 50 and that cover at least 25% of length of gene models are considered. EST support is based on correlation coefficient (CC), a measure commonly used to estimate the accuracy of predicted gene models relative to experimentally known, validated gene models (f.e. Burset & Guigo). For this case 'known' genes are substituted by mapped ESTs, which overlap with a particular gene model. So for a given gene model EST support is measured as an average of all CC computed for every EST overlapping with that model. CC value ranges from 1 for perfect match between ESTs and predicted gene model to -1 for complete disagreement. Each gene model is assigned the following empirical score: S = Sblast * (cov1*cov2 + CC), where Sblast is combined BLASTp score of alignments between given gene model and protein homolog, cov1 and cov2 are alignment coverages for the model and homolog respectively (0 <= cov1,cov2 <= 1) and CC is the correlation coefficient between the model and overlapping ESTs. At each locus, a model with the highest score is selected, and all other models, which have at least 5% overlap with the selected model are discarded.

The selected gene models form the GeneCatalog, which is subject to further genome analysis, manual curation, and, ultimately, GenBank submission.

Non-coding genes. In addition to protein coding genes, the pipeline predicts non-coding genes. tRNAs are predicted using tRNAscan-SE [Lowe and Eddy 1997]. Infernal software suite (S.Eddy, Janelia Farms) is used to predict members of known non-coding RNA families from RFAM database, which combines information of primary sequences and secondary structure of RNAs. In addition, regions of genomic conservation determined by VISTA alignments [Ratnere & Dubchak, 2009] and supported by expression data may suggest additional non-coding genes.

2. Functional Annotation.

All predicted proteins are functionally annotated using SignalP [Nielsen et al. 1997] for signal sequences, TMHMM [Melen et al. 2003] for transmembrane domains, InterProScan [Quevillon et al, 2005] for integrated collection of functional and structure protein domains, and protein alignments to NCBI nr, SwissProt (http://www.expasy.org/sprot/), KEGG [Kanehisa et al. 2006] for metabolic pathways, and KOG [Koonin et al. 2004] for eukaryotic clusters of orthologs. KEGG hits are used to assign EC numbers (http://www.expasy.org/enzyme/) and map to metabolic

pathways. Interpro and SwissProt hits are used to map gene ontology (GO) terms (Ashburner et al, 2000; http://www.geneontology.org/).

Besides linking predicted proteins to various annotation sources, a defline for each protein is inferred from the top BLASTp protein hit when amino acid sequence identity and coverage > 80% or from InterPro domain with e-value<1e-15. When deflines do not meet GenBank requirements, it is replaced with ' hypothetical protein'.

Predicted protein sets are also run through external pipelines built by collaborators for specific gene families such as CAZy, FOLy, transportDB, etc

3. Comparative Analysis

Each annotated genome is analyzed in the context of comparative analysis using several tools such as VISTA, synteny, dot-plots, functional profiles (KOG, EC, GO), and gene clusters with visualization tools enabling comparative analysis of gene structure, domain composition, and genomic neighborhood for each gene cluster.

The masked assembly is aligned to assemblies from a pre-defined group of related organisms using VISTA. These alignments are translated into regions of DNA conservation displayed on the genome browsers and also visualized using interactive dot-plots and synteny analysis tools.

Functional annotation of individual genes in the GeneCatalog is summarized according to different classification schemas (eg, KOG, KEGG, GO) and presented as functional profiles (gene counts in each functional category) for compared genomes.

Multi-gene families are predicted with the Markov clustering algorithm (MCL [Enright et al. 2002]), which clusters proteins based on BLASTp alignment scores between them. Gene clusters are annotated using PFAM domains detected in cluster member sequences. The display of gene structure, synteny and domains composition assists validation of gene families.

Quality control of annotated genomes includes analysis of GeneCatalog composition, support by different lines of evidence such as ESTs and homology, statistical characteristics of gene models such as gene length, number of exons, intron size etc., and comparative analysis with previously annotated genomes. This also involves manual curation of a random selection of predicted genes. Quality assessment of annotated genome involves a multi-tier process that include (i) assessment by annotator, (ii) peer review, (iii) community annotation, and (iv) GenBank review.

Implementation

The fungal pipeline uses a framework of pipeline infrastructure tools to monitor and control abstract pipelines running on Linux clusters. These tools enable automatic setup of the pipeline, visualization of the pipeline run with interactive control by annotators, APIs to external tools for gene cluster analysis and Vista alignments, and automatic web portal construction and configuration. Detailed error detection and fault identification procedures are provided for the annotators to troubleshoot problems.

Pipeline/Portal setup process. A Perl script controls all the input data for annotation, creates directories and copies files to appropriate locations, and builds and launches the pipeline also sending work requests to cluster, Vista and Portal subsystems.

Annotation Pipeline process uses the Pipeline infrastructure to run a network of programs that implement the database and perform analyses required to create a genome portal. Extensive checking is done to insure that programs are run correctly and log files are created for possible multiple runs of the pipeline. Pipelines can run on all the JGI clusters and can easily be moved from cluster to cluster as the workload changes and a cluster becomes busy.

Cluster Analysis system is notified by the annotation pipeline when data is available for multiple compared genomes. At the end, the annotation pipeline verifies completeness of the work and connects the clustering results to the organism database.

Vista Analysis system is notified by the annotation pipeline when the masked genome is made available, which also launches whole-genome synteny analysis. At the end of the pipeline, Vista results are connected to the organism database automatically.

Web Portal construction and configuration is initiated by the annotation pipeline, which also adds data and services to the portal as corresponding analysis steps are executed to allow immediate display of results. At the end of the annotation pipeline the portal is notified to move the organism database from the staging server to the production server.

Pipeline infrastructure tools. The pipeline construction web interface enables graphical construction of abstract pipelines that are parametized by symbolic on/off tags to include/exclude parts (e.g. specific gene modelers) and by text substitution macros that control command lines generated within the pipeline. Pipeline sections are defined in a template which defines a program execution subgraph. Templates are nested to provide modularity in pipeline specification. The pipeline monitoring/control web interface allows annotators to view the pipeline graph with color coded status of pipeline modules. A mouse click on a program node in the graph instantly displays the log file for that program allowing quick problem determination. Parts of the pipeline can be suspended or moved to different queues on the cluster. The status of hardware can be also displayed.

References:

Ashburner, M. et al. Gene ontology: tool for the unification of biology. Nature Genetics 25, 25-9 (2000).

Bendtsen, J. D., Nielsen, H., von Heijne, G. & Brunak, S. Improved prediction of signal peptides: SignalP 3.0. Journal of Molecular Biology 340, 783-795 (2004).

Birney, E., Clamp, M. & Durbin, R. GeneWise and Genomewise. Genome Research 14, 988-995 (2004).

Enright AJ, Van Dongen S & Ouzounis CA. An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research 30(7):1575-1584 (2002)

Ter-Hovhannisyan V, Lomsadze A, Chernoff YO, Borodovsky M. Gene prediction in novel fungal genomes using an ab initio algorithm with unsupervised training. Genome Res. 2008 Dec;18(12):1979-90. Epub 2008 Aug 29.

Jurka J, Kapitonov VV, Pavlicek A, Klonowski P, Kohany O & Walichiewicz J. Repbase Update, a database of eukaryotic repetitive elements. Cytogenet. Genome Res. 110:462–467 (2005).

Kanehisa M, G. S., Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, Katayama T, Araki M, Hirakawa M. From genomics to chemical genomics: new developments in KEGG. Genome Biology 5, R7 (2006).

Kent WJ. BLAT--the BLAST-like alignment tool. Genome Res. 12(4):656-664 (2002).

Koonin, E. V. et al. A comprehensive evolutionary classification of proteins encoded in complete eukaryotic genomes. Genome Biology 5, R7 (2004).

Lowe TM, Eddy SR. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Res. 25(5):955-964 (1997)

Melen K, Krogh A & von Heijne G. Reliability measures for membrane protein topology prediction algorithms. J. Mol. Biol 327(3):735-744 (2003).

Nielsen H, Engelbrecht J, Brunak S & von Heijne, G. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Protein Engineering, 10:1-6, (1997).

Price AL, Jones NC, Pevzner PA. De novo identification of repeat families in large genomes. Bioinformatics. 2005 Jun;21 Suppl 1:i351-8.

Quevillon, E. et al. InterProScan: protein domains identifier. Nucleic Acids Res 33, W116-20 (2005).

Ratnere I, Dubchak I. Obtaining comparative genomic data with the VISTA family of computational tools. Curr Protoc Bioinformatics. 2009 Jun;Chapter 10:Unit 10.6.

Salamov, A. A. & Solovyev, V. V. Ab initio gene finding in Drosophila genomic DNA. Genome Res. 10, 516-22 (2000).

Smit, AFA, Hubley, R & Green, P. *RepeatMasker Open-3.0*. 1996-2010