Eval: A software package for analysis of genome annotations

Evan Keibler
Washington University in St Louis

Michael R. Brent
Washington University School of Medicine in St. Louis

Follow this and additional works at: https://digitalcommons.wustl.edu/open_access_pubs

Part of the Medicine and Health Sciences Commons

Recommended Citation
https://digitalcommons.wustl.edu/open_access_pubs/136

This Open Access Publication is brought to you for free and open access by Digital Commons@Becker. It has been accepted for inclusion in Open Access Publications by an authorized administrator of Digital Commons@Becker. For more information, please contact engeszer@wustl.edu.
Software

Eval: A software package for analysis of genome annotations
Evan Keibler and Michael R Brent*

Address: Department of Computer Science and Engineering, Washington University, St. Louis, MO 63130, USA
Email: Evan Keibler - evan@cse.wustl.edu; Michael R Brent* - brent@cse.wustl.edu
* Corresponding author

Abstract

Summary: Eval is a flexible tool for analyzing the performance of gene annotation systems. It provides summaries and graphical distributions for many descriptive statistics about any set of annotations, regardless of their source. It also compares sets of predictions to standard annotations and to one another. Input is in the standard Gene Transfer Format (GTF). Eval can be run interactively or via the command line, in which case output options include easily parsable tab-delimited files.

Availability: To obtain the module package with documentation, go to http://genes.cse.wustl.edu and follow links for Resources, then Software. Please contact brent@cse.wustl.edu

Introduction

Automated gene annotation systems are typically based on large, complex probability models with thousands of parameters. Changing these parameters can change a system's performance as measured by the accuracy with which it reproduces the exons and gene structures in a standard annotation. While traditional sensitivity and specificity measures convey the accuracy of gene predictions [1,2], more information is often required for gaining insight into why a system is performing well or poorly. A deep analysis requires considering many features of a prediction set and its relation to the standard set, such as the distribution of number of exons per gene, the distribution of predicted exon lengths, and accuracy as a function of GC percentage. Such statistics can reveal which parameter sets are working well and which need tuning. We are not aware of any publicly available software systems that have this functionality. We therefore developed the Eval system to support detailed analysis and comparison of the large data sets generated by automated gene annotation systems [e.g., [3]].

Features

Statistics

Eval can generate a wide range of statistics showing the similarities and differences between a standard annotation set and a prediction set. It reports traditional performance measures, such as gene sensitivity and specificity, as well as measures focusing on specific features, including initial, internal, and terminal exons, and splice donor and acceptor sites (see Table 1 for a sampling of these statistics; for a complete list of all calculated statistics see online documentation). These specific measures can show why an annotation system is performing well or poorly on the traditional measures. They can also reveal specific weaknesses or strengths of the system – for example, that it is good at predicting the boundaries of genes but has problems with exon/intron structure because it does poorly on splice donor sites. Eval can also compute statistics on a single set of gene annotations (either predictions or standard annotations). These statistics reveal the average characteristics of the genes, such as their coding and genomic lengths, exon and intron lengths, number of
exons, and so on. This is useful when tuning the parameters of annotation systems for optimal performance.

Plots
Eval can also produce two types of plots. One type is a histogram showing the distribution of a statistic. Histograms are useful for determining whether the annotation system is producing specific types of genes and exons in the expected proportions. For example, suppose that the average number of exons per gene in an automated annotation is slightly below that of a standard annotation. Comparing the two distributions can reveal whether that difference is due to an insufficient fraction of predictions with extremely large exon counts or an insufficient fraction with slightly above-average exon counts (Fig. 1a). The other type of plot categorizes exons or genes by their length or GC content and shows the statistic for each category. For example, plotting transcript sensitivity as a function of transcript length might reveal that an annotation system is performing poorly on long genes but well on short ones (Fig. 1b). Further analysis would be needed to determine whether this effect is due to intron length or exon count.

Multi-way comparisons (Venn diagrams)
Eval can also determine the similarities and differences among multiple annotation sets. For example, it can build clusters of genes or exons which share some property, such as being identical to each other or overlapping each other. Building clusters of identical genes from two gene predictors and a standard annotation can show how similar the predictors are in their correctly and incorrectly predicted genes. For example, it could reveal that the two programs predict the same or completely separate sets of correct and incorrect genes. If they predict correct gene sets with a small intersection and incorrect gene sets with a large intersection then they could be combined to create a system which has both a higher sensitivity and specificity than either one alone. Table 2 shows a different example - clustering of identical exons from the aligned human RefSeq mRNAs, TWINSCAN [3,4] predictions, and GENSCAN [5] predictions.

Extraction of subsets
Eval can also extract subsets of genes that meet specific criteria for further analysis. Sets of genes that match another gene set by any of the following criteria can be selected: exact match, genomic overlap, CDS overlap, all introns match, one or more introns match, one or more exons match, start codon match, stop codon match, start and
stop codon match. Boolean combinations of these criteria can also be specified. For example, the set of RefSeq genes that are predicted correctly by System1 but not by System2 can be extracted from annotations of the entire human genome with just a few commands. Once extracted, gene sets can be inspected individually using standard visualization tools.

**Implementation**
Eval is written in Perl and uses the Tk Perl module for displaying its graphical user interface. It is intended to run on Linux based systems, although it also runs under Windows. It requires the gnuplot utility to display the graphs it produces, but it can create the graphs as text files without this utility. The package comes with both command line and graphical interface. The command line interface provides quick access to the functions, while the graphical
interface provides easier, more efficient access when running multiple analyses on the same data sets.

Annotations are submitted to Eval in GTF file format http://genes.cse.wustl.edu/GTF2.html, a community standard developed in the course of several collaborative genome annotations projects [6,7]. As such it can be run on the output of any annotation system. The Eval package contains a GTF validator which verifies correct GTF file format and identifies common syntactic and semantic errors in annotation files. It also contains Perl libraries for parsing, storing, accessing, and modifying GTF files and comparing sets of GTF files.

Although it is written in Perl, the Eval system runs relatively quickly. A standard Eval report comparing all TWINSCAN [3,4] genes predicted on the human genome to the aligned human RefSeqs processes ~40,000 transcripts and ~300,000 exons and completes in under five minutes on a machine with a 1.5 GHz Athlon processor and 2 GB of RAM.

Acknowledgments
This work was supported in part by grant DBI-0091270 from the National Science foundation to MRB and grant HG02278 from the National Institutes of Health to MRB.

References