Scalable and reproducible genome analysis in the age of next-generation genome sequencing

by

Daniel Scott Standage

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Bioinformatics and Computational Biology

Program of Study Committee:
Volker Brendel, Co-major Professor
Amy Toth, Co-major Professor
Karin Dorman
Xiaoqui Huang
Jonathan Wendel

Iowa State University
Ames, Iowa
2016

TABLE OF CONTENTS

LIST OF TABLES v

LIST OF FIGURES vi

ACKNOWLEDGEMENTS vii

CHAPTER 1. OVERVIEW 1

Introduction 1

Dissertation Organization 5

CHAPTER 2. GENOME, TRANSCRIPTOME, AND METHYLOME SEQUENCING OF A PRIMITIVELY EUSOCIAL WASP REVEAL A GREATLY REDUCED DNA METHYLATION SYSTEM IN A SOCIAL INSECT 6

Introduction 6

Hypothesis 6

Second Hypothesis 7

Criteria Review 7

CHAPTER 3. PARSEVAL: PARALLEL COMPARISON AND ANALYSIS OF GENE STRUCTURE ANNOTATIONS 8

Abstract 8

Background 9

Implementation 11

Overview 11

Gene locus identification 12
Gene structure representation 13
Comparative analysis of annotations 15
Reporting comparison scores 16
Results and Discussion 18
Use case: predictions vs. gold standard 18
Use case: two sets of annotations 19
Benchmarks 21
Performance evaluation in comparison to Eval software 23
Conclusions 24
Availability and requirements 24
Authors contributions 25
Acknowledgements 25

CHAPTER 4. ILOCI: SCALABLE GENOME ANNOTATION FOR PROVISIONAL GENOME ASSEMBLIES 26

Introduction 26
Hypothesis 28

CHAPTER 5. GENEANNOLOGY: SCALABLE AND REPRODUCIBLE GENOME ANALYSIS WITH GENE ANNOTATION VERSION CONTROL 29

Introduction 29
Hypothesis 29
Second Hypothesis 30
Criteria Review 30

CHAPTER 6. SUMMARY AND DISCUSSION 31

Introduction 31

Hypothesis 31

Second Hypothesis 31

Criteria Review 33

APPENDIX A. ADDITIONAL MATERIAL 35

APPENDIX B. STATISTICAL RESULTS 36

BIBLIOGRAPHY 37

LIST OF TABLES

Table 2.1 Moon Data 6

Table 4.1 This table shows a standard empty table 27

Table 5.1 Moon Data 29

Table 6.1 This table shows almost nothing but is a sideways table and takes up a whole page by itself 32

LIST OF FIGURES

Figure 2.1 Durham Centre 7

Figure 4.1 This table shows a standard empty figure 28

Figure 5.1 Durham Centre 30

Figure 6.1 Durham Centre— Another View 34

ACKNOWLEDGEMENTS

To go here.

CHAPTER 1. OVERVIEW

Introduction

In the 2000s, the advent of new nucleotide sequencing strategies based on ion semiconductors (Ion Torrent), pyrosequencing (454), and sequencing-by-synthesis (Illumina) provided new tools for studying genomes of both model and non-model organisms at unprecedented scale, resolution, and cost effectiveness. These technologies continue to evolve, the latest innovations involving single-molecule long read sequencing (Pacific Biosciences SMRT and Oxford Nanopore). By the 2010s, these so-called *next-generation sequencing (NGS)* technologies had made genome sequencing accessible to essentially any scientist with even a modest research budget. This "democratization of sequencing" has precipitated a tremendous increase in the number of published genome projects and draft genome sequences (http://www.ncbi.nlm.nih.gov/genome/browse/), as well as genome-scale data sets profiling gene expression, chromatin accessibility, transcription initiation, and a multitude of other genomic characteristics.

During this same time frame, however, genome analysis has been democratized to a much lesser extent. The availability of high-quality model reference genomes has changed very little, and many scientists struggle to effectively manage these newly acquired data and critically evaluate downstream research products. The proliferation of new algorithms and software tools for analyzing NGS data is a mixed blessing for scientists who now have both the flexibility and the burden of selecting suitable tool(s) for a particular analysis.

The complexity and difficulty of *de novo* genome assembly [1, 2, 3, 4], annotation [5, 6, 7], and analysis (cite pig genes paper? RNA-seq trimming paper? others?) has been reported in various recent studies and community projects. The consistent, resounding message from this body of work is that data quality varies considerably across data sets and even within a single data set. When reference genome assemblies are fragmented, when annotated genome features vary in reliability, when raw -omics data contain a potentially large amount of technical bias or other artifacts, and when the performance of state-of-the-art algorithms is difficult to predict on new data, doing principled and reproducible science requires a framework for disciplined quality control and data evaluation. The focus of this dissertation has been the development of such a framework, and associated software tools, as motivated by research problems I encountered in genomics research projects.[1]

My first task as a graduate student was to evaluate the performance of a new automated genome annotation workflow, involving primarily the comparison of its outputs to an existing reference annotation. Similar comparisons were also required in subsequent projects, particularly while re-annotating an insect genome using an improved transcriptome assembly and additional protein evidence. Quantitative measures of agreement between gene predictions and a reliable reference had long been established (cite Burset/Guigo), and are easily adapted for comparison of two alternative sources of annotation of unknown relative quality. However, software capable of computing these accuracy/similarity statistics at the time (cite Eval, GFPE) were unsatisfactory for our needs. Designed more for algorithm and model refinement than for biological interpretation, these tools report a huge number of statistics aggregated over all data inputs, offering exquisite detail into overall performance but no detail at the level of individual loci. To address these limitations, I created the ParsEval tool to provide locus-level reports in addition to overall aggregate statistics. The strategy utilized by ParsEval for

[1] I have only made it to here with my latest round of revisions.

partitioning the genome into units that can be independently analyzed not only offered significant improvements in runtime and memory usage over previous tools, but also provided the foundation for developing a more generalized genome analysis framework.

Another one of my earliest projects as a graduate student was to assist with *de novo* assembly of the genome of the paper wasp *Polistes dominula*. I subsequently ended up taking charge of the assembly effort, as well as genome annotation, transcriptome profiling, and comparative genomics analysis. These latter analyses get to the heart of the interest of *Polistes* as a model system for studying the evolution of social organization and behavior. Previous to this study, no published genomic data was available for the vespid wasps, one of three major lineages of social *Hymenoptera* (along with bees and ants), and accordingly we were eager to characterize the similarities and differences between the wasps and their hymenopteran relatives. We were also interested in what expression data from adult queens and workers could reveal about genes related to caste differentiation in *Polistes*.

Although our *Polistes* assembly and annotation compared favorably to any other published hymenopteran genome, all the data accessible to us was subject to the same data quality inconsistencies that characterize any NGS-based genome project. Answering questions related to gene expression and genome composition therefore required careful consideration of, for example, how precisely to handle overlapping gene models, and how to distinguish differences rooted in biology from technical artifacts. It was in addressing these issues that we extended the *gene locus* concept introduced by ParsEval and developed *interval loci (iLoci)* as a generalized organizational framework for genome analyses. *iLoci* define an unambiguous parsing of an annotated genome into distinct regions, each encapsulating the genomic context of a gene or intergenic space, providing a complete and granular representation of the genome. Following the publication of the *P. dominula* genome paper, in which *iLoci* played a central role, we investigated general applications of *iLoci*, describing their stability across assembly and annotation versions and their util-

ity for characterizing genome organization, within a single genome and between multiple genomes in a clade of species.

One of the earliest applications of *iLoci* I envisioned was as an organizational principle for version control of annotations. Continual refinement of our genome assembly and annotation made progress on our research more difficult at times, and this issue is certainly not unique to our work. The honeybee, for example, had three official annotation versions in concurrent use during our work on the *Polistes* genome, with some studies even making their own unpublished refinements to an annotation (citations). What seemed to be lacking was a precise way to refer to a particular gene or genomic region, as annotated at a particular time, and to make statements about its expression, or conservation, or methylation status, or any number of additional characteristics. The idea of tracking annotations over time is not new (cite AED paper), and some well-supported communities provide tools for mapping annotations from an older assembly version to updated assembly (cite liftover, RATT). However, *iLoci* provide an alternative solution to these issues, and with the additional benefits previously described provide a complete framework for organization, quality control, and reproducibility for provisional genome projects. At an early stage in my training, I prototyped a tool called *GeneAnnoLogy* for maintaining an annotation version history, leveraging existing version control tools to track changes to individual *iLoci* over time. I later built in features that facilitate filtering *iLoci* based on their length, nucleotide composition, annotation quality, or any number of additional characteristics that can be computed from or attached to the annotation. The *GeneAnnoLogy* tool encapsulates the culmination of concepts and principles investigated in this dissertation.

Dissertation Organization

This dissertation is organized into six chapters. Chapter 1 provides an overview of the dissertation, a motivation for the work, and a brief discussion of relevant literature. Chapters 2 through 5 are presented as complete manuscripts: chapter 2 is a research paper published in *Molecular Ecology* describing the genome, transcriptome, and methylome of the paper wasp *Polistes dominula*, highlighting its reduced DNA methylation system, several hundred genomic loci with caste-related differential expression, and the lack of any detectable caste-related alternative splicing in the adult organism; chapter 3 is a paper published in *BMC Bioinformatics* describing *ParsEval*, a tool for comparing two alternative sources of annotation for a genome sequence; chapter 4 is a methodology to be submitted to *BMC Bioinformatics/Genomics*[2], describing the use of *interval loci (iLoci)* as an organizational framework facilitating reproducible genome analysis; chapter 5 is a manuscript slated for submission to *A Journal* describing *GeneAnnoLogy*, an *iLocus*-based tool for quality control and version control of genome annotations. Chapter 6 provides brief concluding remarks and suggestions for further research.

[2]Will be submitted within a couple of weeks.

CHAPTER 2. GENOME, TRANSCRIPTOME, AND METHYLOME SEQUENCING OF A PRIMITIVELY EUSOCIAL WASP REVEAL A GREATLY REDUCED DNA METHYLATION SYSTEM IN A SOCIAL INSECT

A manuscript published in *Molecular Ecology*.

Introduction

Here initial concepts and conditions are explained and several hypothesis are mentioned in brief.

Of course, data on this as seen in Table 5.1 is few and far between.

Table 2.1 Moon Data

Element	Control	Experimental
Moon Rings	1.23	3.38
Moon Tides	2.26	3.12
Moon Walk	3.33	9.29

Hypothesis

Here one particular hypothesis is explained in depth and is examined in the light of current literature.

Or graphically as seen in Figure 5.1 it is certain that my hypothesis is true.

Figure 2.1 Durham Centre

Parts of the hypothesis

Here one particular part of the hypothesis that is currently being explained is examined and particular elements of that part are given careful scrutiny.

Second Hypothesis

Here one particular hypothesis is explained in depth and is examined in the light of current literature.

Parts of the second hypothesis

Here one particular part of the hypothesis that is currently being explained is examined and particular elements of that part are given careful scrutiny.

Criteria Review

Here certain criteria are explained thus eventually leading to a foregone conclusion.

CHAPTER 3. PARSEVAL: PARALLEL COMPARISON AND ANALYSIS OF GENE STRUCTURE ANNOTATIONS

A paper published in *BMC Bioinformatics*: doi:10.1186/1471-2105-13-187.

Abstract

Background: Accurate gene structure annotation is a fundamental but somewhat elusive goal of genome projects, as witnessed by the fact that (model) genomes typically undergo several cycles of re-annotation. In many cases, it is not only different versions of annotations that need to be compared but also different sources of annotation of the same genome, derived from distinct gene prediction workflows. Such comparisons are of interest to annotation providers, prediction software developers, and end-users, who all need to assess what is common and what is different among distinct annotation sources. We developed ParsEval, a software application for pairwise comparison of sets of gene structure annotations. ParsEval calculates several statistics that highlight the similarities and differences between the two sets of annotations provided. These statistics are presented in an aggregate summary report, with additional details provided as individual reports specific to non-overlapping, gene-model-centric genomic loci. Genome browser styled graphics embedded in these reports help visualize the genomic context of the annotations. Output from ParsEval is both easily read and parsed, enabling systematic identification of problematic gene models for subsequent focused analysis.

Results: ParsEval is capable of analyzing annotations for large eukaryotic genomes

on typical desktop or laptop hardware. In comparison to existing methods, ParsEval exhibits a considerable performance improvement, both in terms of runtime and memory consumption. Reports from ParsEval can provide relevant biological insights into the gene structure annotations being compared.

Conclusions: Implemented in C, ParseEval provides the quickest and most feature-rich solution for genome annotation comparison to date. The source code is freely available (under an ISC license) at http://parseval.sourceforge.net/.

Background

It was only a decade ago when annotating a eukaryotic genome required years of extensive collaboration and millions of dollars of investment. Since then, the tremendous rate at which the cost of DNA sequencing has been dropping as well as increased accessibility to gene prediction software are placing genome sequencing and annotation well within the reach of most single investigator biology laboratories. As a result, proliferation of distinct annotation sets corresponding to the same genomic sequences is becoming increasingly common. Annotation sets for a particular genome can accumulate in a variety of scenarios. When developing gene prediction software, it is common to test the software on a genomic region for which a high-quality reference is available, running and re-running the software and comparing the resulting predictions against the reference. Community groups providing annotation for species- or clade-specific genomes typically release updated annotations following the initial release. Affordable transcriptome sequencing provides individual labs with data to specifically improve annotations for particular genes of interest, for example with respect to alternative splicing. In each of these scenarios, multiple annotations associated with a common set of genomic sequences require comparative assessment.

A variety of comparison methods exist, but none can fully address the growing needs

of the community (see Table 1). Manual comparison approaches can trivially be ruled out as slow, tedious, error prone, and hopelessly unscalable. Although genome browsers have had a huge impact by making gene annotations accessible to a wide variety of scientists, they likewise do little to provide the automation and precision needed in whole-genome annotation comparisons. Large genome sequencing projects and centers have certainly developed in-house scripts and pipelines over the years to address this need. However, these pipelines are typically not standardized, not openly shared, and do not migrate well.

Tools such as the Eval package [8] and the GFPE program [9] represent some of the earliest efforts to provide a reusable, easy-to-use annotation comparison tool to the community. Eval in particular stands out based on the amount of detail provided by its reported comparison statistics and by the ability to visualize the distributions of these statistics. Eval takes as input annotation files in Gene Transfer Format (GTF) and calculates a rich set of descriptive statistics summarizing the differences between the annotations. Because whole-genome annotations typically include thousands (or tens of thousands) of genes, these statistics are intended to condense the information into a comprehensive yet concise summary (at the resolution of entire sequences or sets of sequences), facilitating targeted improvement of gene prediction software. Unfortunately, this condensing process discards large amounts of valuable information at the resolution of individual gene loci, making the tool unsuitable for analyses that target a particular gene, sets of genes, or gene loci with characteristics of interest from within a larger set of genes. Such locus-resolution comparisons are useful not only to software developers and annotation producers who need to know whether their software has distinct advantages or disadvantages, e.g., favoring long over shorter gene models on average, or failing in untranslated region (UTR) prediction, but they are of primary interest for specialists concerned with a particular gene family or pathway.

Motivated by a need for genome-scale evaluations with locus-scale detail, we devel-

oped ParsEval, a program for comparing and analyzing distinct sets of gene structure annotations for the same input sequences. The program is designed to incorporate all of the benefits of existing methods while addressing their shortcomings. ParsEval identifies differences in exon/intron assignments and in coding sequence (CDS) and UTR designations, at both feature-level (exon, CDS segment, UTR segment) and nucleotide-level resolution. The output consists of a set of commonly used statistics that provide quantitative measures of agreement when comparing predicted gene structures against a standard reference [10, 11, 6]. This output is presented in a detailed report for each gene locus, supplemented with genome browser styled graphics to enable additional visual assessment and analysis of the annotations. The statistics are also presented in a single summary report that aggregates the statistics across all loci, providing a condensed high-level view of the similarity between the two sets of annotations. For gene loci that include alternatively spliced genes or overlapping genes (or both), ParsEval determines the optimal matching of reference transcripts to prediction transcripts, and additionally reports any novel transcript predictions that have been identified.

Implementation

Overview

ParsEval is a gene annotation comparison and analysis tool, designed with a focus on speed, resource efficiency, and portability. The program takes as input a pair of gene structure annotations corresponding to the same sequence (in GFF3 format [12]), analogous to two separate annotation tracks one might see in a genome browser. For comparison purposes, the first set of annotations is treated as the *reference* while the other is treated as the *prediction*, although ParsEval makes no assumptions regarding the respective quality of the two annotation sets. The output of the program is a set of reports containing common comparison statistics intended to highlight relevant similarities and

differences between the two sources of annotation.

ParsEval first loads the annotation data into memory, identifies start and end coordinates for gene loci, and associates each gene annotation with a single locus. Next, the program does a comparative assessment of the gene annotations for each locus, calculating and storing a variety of informative similarity statistics. Finally, ParsEval generates reports providing a detailed readout of these statistics.

Implemented in ANSI C, ParsEval is fast, memory efficient, and portable, designed to run on all POSIX-compliant UNIX systems (Linux, Mac OS X, Cygwin, Solaris, etc.). Most of the analysis code is implemented with shared memory parallelization, providing additional performance gains when running on multicore processors that are becoming increasingly common in commodity hardware. ParsEval's only external dependency is the GenomeTools library [13], which provides an API for generating annotation graphics with AnnotationSketch [14], as well as implementations of a variety of data parsers and dynamic data structures.

Gene locus identification

Comparative analysis of two sets of gene annotations requires determining how annotations from one set correspond to annotations from the other, as well as the genomic coordinates (the *gene locus*) that should be considered in each comparison. For rare cases in which a single reference annotation and a single prediction annotation line up perfectly, determining the gene locus and the corresponding genes is trivial. However, in most cases this task is complicated a variety of factors. For example, a single gene prediction workflow may annotate multiple genes at a single location, so one must determine how to associate these annotations with corresponding annotations from an alternative source. Furthermore, when one or more gene annotations from one source overlap with multiple annotations from another source, one must determine how to compare these gene annotations and which coordinates to include in the comparison.

One common approach involves designating one set of annotations as the *reference* set and then using the coordinates of each reference gene annotation to define a distinct gene locus to serve as the basis for subsequent comparison (see Figure 1). However, this approach is unfavorable for several related reasons. First, reference gene annotations that overlap are handled separately, when it makes more sense to associate them with the same locus and handle them together. Second, it forces a quality judgment between the two sets of annotations when their relative quality is often unknown. The two sets of annotations likely include complementary information, and unless there is a clear distinction in quality between the two, choosing one as a reference discards clearly related information from the other. Third, relevant information from predicted gene models that extend beyond the boundaries of the corresponding reference annotation is ignored.

Although ParsEval uses the terms *reference* and *prediction* to distinguish between the two sets of annotations, both are considered equally when identifying gene loci. Each gene annotation corresponds to a node in an interval graph G. There is an edge between two nodes G_i and G_j if the corresponding gene annotations overlap (see Figure 2). Each connected component in G then corresponds to a distinct gene locus, which we define as the smallest genomic region containing every gene annotation associated with the corresponding subgraph. Defining a gene locus in this way makes no assumptions as to the relative quality of the two sets of annotations, and ensures that no potentially relevant data are discarded. Furthermore, according to this definition each gene locus is independent, enabling the subsequent comparative analysis tasks to run in parallel.

Gene structure representation

To facilitate analysis at each gene locus, ParsEval converts GFF3 annotations for each gene into a character string representing the annotated gene structure (a *model vector*). This model vector is similar to a sequence in Fasta format, except instead of using the alphabet $\{A, C, G, T\}$ to represent chemical composition at each nucleotide,

the alphabet $\{C, F, G, I, T\}$ representing gene structure is used: C for coding sequence, F for 5'-UTR, T for 3'-UTR, I for introns, and G for intergenic sequence. Using this alphabet, each transcript can be represented by a single model vector. ParsEval uses these model vectors when comparing reference and prediction gene annotations.

In many cases, a single pair of model vectors (one for the reference, one for the prediction) is sufficient to fully represent annotated gene structure at a given locus. This is certainly true when both the reference and the prediction annotate a single gene with a single mRNA product at the locus. But even if the reference (or the prediction) annotates multiple genes or transcripts, non-overlapping annotations can be encoded in the same model vector and compared simultaneously with corresponding annotations from the other data set. However, if either the reference or the prediction contains annotations for overlapping transcripts, either because of alternative splicing or because of overlapping gene models, a single pair of model vectors is insufficient to represent the complete annotated gene structure at that locus. In these more complicated cases, the reference or the prediction or both will be associated with multiple model vectors. Thus, the algorithmic requirement is to represent all annotated transcript structures in the locus using the smallest number of model vectors.

This problem reduces to a common problem in graph theory known as the *maximal clique enumeration problem* [15]. We treat each transcript as a node in an undirected graph and place an edge between two nodes if the corresponding transcripts do not overlap (unlike the locus identification step, reference annotations and prediction annotations are handled separately in this step). Each maximal clique (maximal fully-connected subgraph) in this graph corresponds to a set of transcripts that do not overlap and can therefore be collapsed into a single model vector. ParsEval uses the Bron-Kerbosch algorithm [15] to enumerate all maximal transcript cliques, first for the reference and then for the prediction. A model vector is generated for each clique, after which ParsEval compares all reference model vectors with all prediction model vectors.

Comparative analysis of annotations

Given a pair of equal-length model vectors representing a pair of gene structure annotations at a given locus, ParsEval computes a variety of comparison statistics to measure the level of agreement between the pair of annotations. Calculated at different levels of resolution, these statistics provide a detailed assessment of similarity between the reference and the prediction. At the resolution of distinct annotation features, ParsEval calculates the sensitivity and specificity as described in [10], the F1 score as described in [11], and the annotation edit distance as described in [6, 16]. These statistics are calculated for exons, CDS segments, and UTR segments. Note that for a prediction feature to be considered a true positive, ParsEval requires both the start and end coordinates to match the reference perfectly.

At the nucleotide-level resolution, ParsEval also calculates the sensitivity, specificity, F1 score, and annotation edit distance, as well as the simple matching coefficient and the correlation coefficient as described in [10]. These statistics are calculated for coding nucleotides (CDS) and untranslated exonic nucleotides (UTR). Overall identity at the nucleotide level, of which the simple matching coefficient is a generalization, is also computed.

For complex loci requiring multiple comparisons, the locus report includes an aggregate summary of the similarity statistics at the locus level in addition to the reports for each individual comparison. This locus-level summary also includes the splice complexity statistic [6], which ParsEval computes and reports for both the reference and the prediction at the locus level.

Based on the computed statistics, each comparison is classified in terms of similarity. A comparison is classified as a *perfect match* if the model vectors (and by implication the annotated gene structures) are identical. A comparison is classified as a *CDS structure match* if the comparison is not a perfect match, but there is perfect agreement in terms of CDS structure. A comparison is classified as an *exon structure match* if there are

differences in the coding sequence that nevertheless preserve exon structure (as resulting from different start and/or stop codons). A comparison is classified as a *UTR structure match* if there are differences in CDS and exon structure, but the UTR structures are identical. All other comparisons are classified as *non-matches*.

Note that, as with feature-level statistics, match classifications require perfect agreement. For instance, a pair of annotations may have very similar CDS structures, and this will be reflected in the nucleotide-level CDS statistics. However, if the CDS structures are not precisely identical, the comparison will not be classified as a *CDS structure match*.

As comparison statistics are computed on a locus-by-locus basis, ParsEval also maintains a running total of all comparison counts (such as true positives and false positives) from which the statistics are computed. When all loci have been considered, each comparison statistic is then recomputed using these running totals to provide an overall assessment of similarity.

Reporting comparison scores

For each gene locus, comparison statistics are calculated for each corresponding pair of reference and prediction model vectors. If multiple comparisons are required at a locus, however, statistics are not reported for each comparison. The comparisons are ranked using the previously described similarity statistics and are reported so as to ensure each transcript (or transcript clique) is considered at most one time. In cases where there is an unequal number of reference and prediction transcripts (or transcript cliques) associated with a particular locus, some will be labeled as novel or unmatched transcripts, and corresponding statistics are not included in ParsEval's reports.

ParsEval presents the comparison statistics in a collection of reports. The first is a single summary report providing the aggregated statistics for a high-level assessment of similarity, as is standard for tools of this kind. Additionally, ParsEval produces a

dedicated comparison report for each individual locus. The detail provided by these locus-level reports is extremely valuable, and ParsEval is the only tool of its kind that preserves and reports comparisons at this level. By default, ParsEval generates these reports in an easy-to-parse and easy-to-read text format. However, ParsEval can also generate the reports as hyperlinked HTML files to facilitate browsing and network-based distribution. Furthermore, ParsEval can supplement HTML reports with embedded PNG graphics providing a genome-browser-like view of each locus' genomic context and enabling visual assessment of the annotations.

If more targeted reporting is desired, ParsEval also provides some filtering features. Using a simple optional configuration file, the user can exclude some gene loci from the reports based on a variety of features: locus length, number of genes, number of transcripts, number of transcripts per gene, number of exons, and CDS length. No comparisons are performed for loci that are filtered out, and thus do not contribute to the reported aggregate summary statistics and comparison classifications.

To facilitate integration of comparison reports with popular genome browsers such as GBrowse [17] and PlantGDB [18], ParsEval can generate an additional output file (in GFF3 format) containing the coordinates of each gene locus. These genome browsers commonly allow users to anonymously create private custom tracks with uploaded data, which provides the quickest mechanism for integration. Once a track is populated with the uploaded locus data, the user can configure the track configuration so that each locus feature in the track is hyperlinked to the corresponding ParsEval report stored, for example, on that user's local machine (see Figure 3). Alternatively, if a more permanent and public solution is desired, a user with administrative privileges for the genome browser can follow standard procedures for populating a new track with the GFF3 data, and then configure the track so that locus features are linked to network-accessible ParsEval reports.

Results and Discussion

We present several use cases to demonstrate ParsEval's capabilities, benchmark its performance, and compare its utility relative to existing methods. The input data for these demonstrations were obtained from a variety of public databases with different respective formatting conventions. Accordingly, all data files were processed and converted to a uniform format before analysis. A detailed description of this conversion process, along with all code and commands used, are provided in the Supplemental Data as well as in ParsEval's source code distribution.

Unless otherwise noted, all use cases and benchmarks described herein were run on a fairly modest desktop computer: a Mac Pro with two 2.8 GHx quad-core Intel Xeon processors and 4 GB of RAM. ParsEval's performance for these demonstrations should therefore be fairly representative of the performance one might expect when running on commodity laboratory or personal hardware.

Use case: predictions vs. gold standard

High-quality gene structure annotations derived from a combination of computational and experimental evidence, and possibly improved with expert manual curation, are indispensably used as "gold standards" for measuring the accuracy of a novel gene prediction method or entire new annotation workflows. Identifying differences between the new method's predictions and such gold standard reference can help identify areas in which the novel method provides or needs improvement. Reports from ParsEval are effective for quickly and clearly identifying such differences.

To demonstrate ParsEval in this context, we reproduced a comparison that was originally published to assess the performance of the AUGUSTUS gene prediction program [19]. In the original study, AUGUSTUS was tested on the *h178* data set [20], a set of 178 human genomic sequences, each containing a single gene, for which annotations were

available from the EMBL database release 50 [21]. Gene predictions from AUGUSTUS were compared the annotations from EMBL, and sensitivity and specificity scores were calculated at the nucleotide level, the exon level, and the gene level.

We obtained the *h178* data set (sequences and EMBL r50 annotations) from [22]. We then used the latest version of AUGUSTUS (2.5.5) to generate gene predictions for the 178 sequences. The data files were reformatted and then compared using ParsEval. Running on a desktop computer, ParsEval generated graphical reports in less than a minute. The summary report provided immediate access to a variety of similarity metrics, including those reported in the original assessment. The sensitivity and specificity values reported by ParsEval are comparable to those reported in the original AUGUSTUS manuscript (see Table 2). Differences in the comparison metrics can likely be explained by improvements to the AUGUSTUS program since publication, although the exact reason is elusive since the original AUGUSTUS software is no longer accessible.

Use case: two sets of annotations

When working with genome annotations, there is an increasing variety of cases in which no gold standard is available for comparison. For example, gene annotations for many model species are available from a variety of sources (i.e., UCSC versus Ensembl). The respective quality of these different annotation sets is not always clear, but comparison is still a necessary and fundamental task. Another example relates to genome projects that typically offer multiple releases of gene annotations between each major genome assembly release. Although newer releases may offer marginal improvements over the older ones, neither one can truly be considered a high-quality standard reference for comparison. An additional example relates to the increased affordability of genome sequencing and the number of new and exotic species for which genome sequence is available. Gene annotation software is based on complex statistical models containing many parameters, and it is not always initially clear which parameter values to use up

front. Therefore, when annotating a newly sequenced genome, it is common to extract a subset of the genome on which to perform repeated optimization runs to determine the parameter values that should be used subsequently to annotate the entire genome.

In each of these scenarios, multiple annotation sets must be compared, despite having no intuition as to the relative quality of the respective annotations. ParsEval was designed precisely for this type of analysis. Reports from ParsEval provide both an overall summary and locus-level detail, enabling the user to make informed decisions about annotations for individual loci, as well as for annotation sets as a whole.

As a demonstration of ParsEval's capability in this context, we downloaded two recent gene annotation releases (releases 64 and 65) for *Mus musculus* from the Ensembl database [23]. We compared these annotations using ParsEval, which required approximately 3 minutes of runtime on a desktop computer. A brief review of ParsEval's summary report shows that a total of 20,362 gene loci were identified using these annotations (see Table 3 for a complete breakdown). Of these gene loci, 6,725 had only annotations from release 64.

23,590 comparisons were performed by ParsEval, of which 22,333 (94.7%) were perfect matches between releases 64 and 65. A small number (83, 0.4%) of comparisons were classified as UTR structure matches. For the remaining 1,174 comparisons (5.0%) that were classified as non-matches, transcripts from release 64 contained an average of 16.47 exons, whereas transcripts from release 65 contained an average of 8.11 exons. A brief review of a handful of selected loci showed that many long transcripts (with many exons) that had been present in release 64 were absent in release 65.

This use case is an ideal demonstration of ParsEval's capabilities. Although the authors have no prior experience working with these particular data sets, a cursory examination ParsEval's reports clearly draw attention to an important fact—between release 64 and 65, changes to Ensembl's annotation pipeline (perhaps different values for parameters that influence joining/splitting annotations, or implementation of stricter

filters for gene length) affected approximately 5% of the gene annotations. Not only does ParsEval provide this information in a summarized form, it also provides detailed locus reports enabling users to scrutinize the results on a gene-by-gene basis. This breadth and detail of information is of great benefit to a wide variety of scientists and will empower them to more fully understand the available data and make informed decisions regarding alternative sources of annotation.

Benchmarks

To demonstrate its speed, scalability, and efficiency, we benchmarked ParsEval by analyzing pairs of whole-genome gene structure annotations for four common model organisms representing a wide range of eukaryotic diversity: *Arabidopsis thaliana* (thale cress), *Drosophila melanogaster* (fruit fly), *Glycine max* (soybean), and *Homo sapiens* (human) (see Table 4). To give a detailed demonstration of its performance, ParsEval was run 24 times for each species—3 technical replicates while varying the output mode (text and HTML/PNG) and the number of dedicated processors (1, 2, 4, and 8). Reported runtimes were obtained by taking the mean of the 3 corresponding replicates.

Performance in text output mode

ParsEval demonstrated optimal performance when running in text output mode, with runtimes ranging between about 30 seconds to about 4 minutes. Running ParsEval in parallel on multiple processors provided noticeable improvement in runtime for *Drosophila* and human, although no improvement was seen for *Arabidopsis* and soybean. It is likely that for loci with relatively small and simple gene structures, ParsEval's runtime is bound more by serial I/O related tasks than by actual analytical computations, which would explain why no improvement was observed for the plant species.

Performance in HTML output mode with PNG graphics

Running ParsEval in HTML/PNG output mode increased the runtimes by an order of magnitude, although parallel processing kept these runtimes within a reasonable range (about a half hour for the most intensive comparison) with observed speedup factors ranging from 3 to 5 when using all 8 processors. Because these improvements in runtime were observed for all species, it is likely that ParsEval's runtime is bound primarily by computationally intensive graphics generation tasks when running in HTML/PNG output mode.

Notes on benchmark results

The results of the *A. thaliana* benchmark were not surprising. Perfect matches and CDS matches account for 97.5% of the comparisons, which makes sense considering that TAIR10 represents minor cumulative updates to TAIR9 (in contrast, perfect matches and CDS matches account for only 4.2% of comparisons between TAIR6 and TAIR10). There were even fewer differences between FlyBase and Ensembl annotations for the *D. melanogaster* benchmark ($\approx 0.1\%$ of loci), suggesting perhaps that these differences may be the consequence of technical artifacts in one data set or the other.

The results of the other two benchmarks, for *G. max* and *H. sapiens*, were somewhat surprising. In each case, approximately 10% of the comparisons reflected perfect matches between the two annotations (6.4% for soybean and 15.3% for human), while approximately 50% of the comparisons reflected CDS matches (45.1% for soybean and 54.9% for human). Therefore, for the remaining approximate 30% of human genes and 50% of soybean genes, the annotated coding sequence (and the associated polypeptide) is different depending on the data source. These differences are likely the result of different annotation strategies between the alternative sources of annotation. Regardless, this is an important point of consideration both for consumers and producers of gene structure annotations, and we hope that the ParsEval tool will be a useful asset to a wide variety

of scientists that rely on reliable gene annotations for their research.

Performance evaluation in comparison to Eval software

To evaluate ParsEval's performance in comparison to existing methods, we used the Eval tool [8] to repeat one of the previously described use cases. Gene annotations for *Mus musculus* were retrieved from releases 64 and 65 of the Ensembl database, and subsequently analyzed using both Eval and ParsEval. Some small differences were observed in the similarity statistics computed by the two programs, although this was not unexpected as Eval uses a different approach than ParsEval for matching reference annotations to prediction annotations. Also, the two programs provide a different breakdown of the similarity statistics, making a rigorous comparison between the Eval results and the ParsEval results impractical.

Running Eval on the complete data sets exhausted the desktop computer's memory resources after several minutes, so comparison of Eval and ParsEval was only possible after restricting the data sets to annotations for *M. musculus* chromosomes 1 through 10. To analyze these reduced data sets, Eval required an average of 12 minutes 13 seconds and consumed all available memory. On the other hand, ParsEval, running on a single processor, required an average of 1 minute 44 seconds, with memory consumption peaking at approximately 0.5 GB. When run on 4 processors, ParsEval's performance margin increased with an average runtime of 47 seconds.

To ensure that Eval's performance was not being severely affected by the desktop's limited system memory, the comparison was also performed in a high-performance computing environment in which memory could not have been a limiting factor. ParsEval continued to demonstrate superior performance in this environment as well, although by a slightly less drastic margin. The Eval program required an average of 7 minutes 18 seconds of runtime, while ParsEval required an average of 1 minute 19 seconds using a single processor, or 37 seconds using 4 processors.

These tests conclusively demonstrate two important points regarding the performance of ParsEval relative to Eval: not only is ParsEval markedly faster, but its resource efficiency also makes it much better equipped to run whole-genome comparisons on the laptop or desktop computers one might expect to see in the typical biology lab. The initial runtimes reported herein should be fairly representative of what users can expect to observe when running ParsEval on commodity hardware.

Conclusions

The accessibility of genome annotation tools to an increasingly wider variety of scientists will soon be accompanied by an increased demand for supplementary tools to manage and analyze genome annotations. We address this need with ParsEval, a tool for fulfilling a common, fundamental analytical need for which existing software is lacking. ParsEval is a portable, easy-to-install, and efficient program for comparing gene structure annotations, and facilitates a wide variety of downstream comparative analyses. We demonstrate the speed and scalability of ParsEval, even when working with large eukaryotic genomes. Furthermore, we highlight the capability of the detailed comparison statistics in ParsEval reports to highlight relevant biological trends in the data. We anticipate that ParsEval will enable a wide variety of biologists to more fully take advantage of the vast genome annotation data resources accumulating in their individual labs and in the community at large.

Availability and requirements

Source code for ParsEval is available at http://parseval.sourceforge.net under an ISC license. ParsEval is implemented in ANSI C and is designed to run on all POSIX-compliant UNIX systems (Linux, Mac OS X, Cygwin, Solaris, etc.). Aside from a C compiler with OpenMP support (such as GCC 4.2 or higher), ParsEval's only external

dependency is the GenomeTools library [13].

Authors contributions

DS designed and implemented the software and drafted the manuscript. VB supervised the project and provided design and feature suggestions. Both authors conceived the project, edited the manuscript, and approved the final version.

Acknowledgements

The authors would like to thank the developers of the GenomeTools software for helpful feedback regarding integration of AnnotationSketch, and our colleagues Carolyn Lawrence and Amy Toth as well as anonymous reviewers whose suggestions were a valuable contribution to this manuscript.

Funding: This work was supported in part by the U.S.A. National Science Foundation Plant Genome Research Program grant ISO#1126267 to V.B..

CHAPTER 4. ILOCI: SCALABLE GENOME ANNOTATION FOR PROVISIONAL GENOME ASSEMBLIES

A paper to be published in *BMC Genomics/Bioinformatics*.

Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus ac feugiat mauris. Nunc sed felis a purus finibus cursus in eu ligula. Nam cursus iaculis augue eget rutrum. Curabitur sed lorem posuere, ultricies nisl ac, dictum est. Praesent accumsan urna turpis, nec tristique nulla rutrum sollicitudin. Vivamus eu sapien id risus fringilla faucibus. Vestibulum euismod, nibh nec rutrum interdum, ex urna vehicula lorem, et fermentum tortor nunc at nisi. Curabitur urna metus, suscipit a ipsum ac, consectetur pharetra augue. Sed sit amet turpis vel risus vehicula dapibus. Duis mattis metus tellus, sit amet placerat lacus tincidunt ut. Nullam dictum lacus magna, in porttitor elit malesuada nec. Quisque quis massa luctus dui tincidunt hendrerit vel eget nisl.

Suspendisse et massa dolor. Cras cursus finibus enim in dapibus. Morbi aliquet placerat arcu, sed tristique ante pulvinar nec. Proin non metus non felis imperdiet tristique tristique vel augue. Cras posuere condimentum purus, vitae tempus tellus. Sed nibh velit, scelerisque vitae felis sit amet, dignissim sollicitudin tellus. Nam eget lacus vitae dolor fermentum fermentum id id magna. Donec auctor euismod porta. Cras in ante scelerisque, placerat enim eu, dictum nisi. Integer nunc eros, elementum tempor

arcu sed, tristique hendrerit leo.

As can be seen in Table 4.1 it is truly obvious what I am saying is true.

Table 4.1 This table shows a standard empty table

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Cras tincidunt vehicula mi in ultrices. Proin vitae mauris aliquam, rutrum mi non, maximus libero. Cras sit amet metus sit amet nisi posuere eleifend. Nam et sapien odio. In ultrices elit nibh, sit amet commodo purus lobortis vitae. Quisque ac felis interdum, ornare nulla fringilla, posuere augue. Nullam dictum et arcu non ornare. In faucibus hendrerit nibh nec mollis. Nam eu dolor sodales mauris fermentum ornare ac ac ante. Etiam non odio sed odio faucibus luctus sed sed nulla. Aliquam sit amet est bibendum, lacinia velit eget, ornare mi. Nam eros neque, scelerisque quis cursus egestas, placerat eu nisl. Vivamus scelerisque odio at ipsum faucibus faucibus. Mauris consequat eu felis nec vulputate.

Fusce finibus erat nulla, eget vestibulum diam tristique ac. Fusce nisi diam, finibus vitae fermentum nec, placerat sodales tellus. Praesent et accumsan nunc. Pellentesque quam orci, rutrum quis ultricies quis, facilisis a ante. Curabitur felis ex, efficitur ut blandit eu, luctus quis enim. Aliquam ac lacinia massa. Quisque aliquam, quam at aliquam venenatis, magna ante auctor purus, nec pharetra turpis urna et diam. Duis eu lectus eget risus ultrices lacinia. Quisque tincidunt purus ac nunc ornare, at pharetra erat rutrum. Sed massa sem, iaculis at vestibulum eget, accumsan eu nibh.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed lacus augue, euismod sed lacinia at, rutrum eget lectus. Donec egestas massa ac risus finibus mattis id eu velit. Mauris fermentum ligula vel tempor mattis. Etiam vehicula arcu a venenatis elementum. Ut cursus molestie ex eget auctor. Morbi eget risus a purus sodales semper. Quisque efficitur laoreet nunc, interdum volutpat orci. Sed quam ligula, dignissim sed dui vel, finibus ultricies elit. Praesent ipsum lectus, finibus sit amet mattis vitae, tempus non turpis. Nulla facilisi. Curabitur et dignissim nibh.

Hypothesis

Here one particular hypothesis is explained in depth and is examined in the light of current literature.

This can also be seen in Figure 4.1 that the rest is obvious.

Figure 4.1 This table shows a standard empty figure

CHAPTER 5. GENEANNOLOGY: SCALABLE AND REPRODUCIBLE GENOME ANALYSIS WITH GENE ANNOTATION VERSION CONTROL

A manuscript submitted to *Bioinformatics*.

Introduction

Here initial concepts and conditions are explained and several hypothesis are mentioned in brief.

Of course, data on this as seen in Table 5.1 is few and far between.

Table 5.1 Moon Data

Element	Control	Experimental
Moon Rings	1.23	3.38
Moon Tides	2.26	3.12
Moon Walk	3.33	9.29

Hypothesis

Here one particular hypothesis is explained in depth and is examined in the light of current literature.

Or graphically as seen in Figure 5.1 it is certain that my hypothesis is true.

Figure 5.1 Durham Centre

Parts of the hypothesis

Here one particular part of the hypothesis that is currently being explained is examined and particular elements of that part are given careful scrutiny.

Second Hypothesis

Here one particular hypothesis is explained in depth and is examined in the light of current literature.

Parts of the second hypothesis

Here one particular part of the hypothesis that is currently being explained is examined and particular elements of that part are given careful scrutiny.

Criteria Review

Here certain criteria are explained thus eventually leading to a foregone conclusion.

CHAPTER 6. SUMMARY AND DISCUSSION

Introduction

Here initial concepts and conditions are explained and several hypothesis are mentioned in brief.

Or graphically as seen in Figure 6.1 it is certain that my hypothesis is true.

Hypothesis

Here one particular hypothesis is explained in depth and is examined in the light of current literature.

As can be seen in Table 6.1 it is truly obvious what I am saying is true.

Parts of the hypothesis

Here one particular part of the hypothesis that is currently being explained is examined and particular elements of that part are given careful scrutiny.

Second Hypothesis

Here one particular hypothesis is explained in depth and is examined in the light of current literature.

Table 6.1 This table shows almost nothing but is a sideways table and takes up a whole page by itself

Element	**Control**	**Experimental**
Moon Rings	1.23	3.38
Moon Tides	2.26	3.12
Moon Walk	3.33	9.29

Parts of the second hypothesis

Here one particular part of the hypothesis that is currently being explained is examined and particular elements of that part are given careful scrutiny.

Criteria Review

Here certain criteria are explained thus eventually leading to a foregone conclusion.

Figure 6.1 Durham Centre— Another View

APPENDIX A. ADDITIONAL MATERIAL

This is now the same as any other chapter except that all sectioning levels below the chapter level must begin with the *-form of a sectioning command.

More stuff

Supplemental material.

APPENDIX B. STATISTICAL RESULTS

This is now the same as any other chapter except that all sectioning levels below the chapter level must begin with the *-form of a sectioning command.

Supplemental Statistics

More stuff.

BIBLIOGRAPHY

[1] Earl D et al. (2011) Assemblathon 1: A competitive assessment of de novo short read assembly methods. *Genome Research* 21(12):2224–2241.

[2] Bradnam KR et al. (2013) Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *GigaScience* 2(1):1–31.

[3] Salzberg SL et al. (2012) Gage: A critical evaluation of genome assemblies and assembly algorithms. *Genome Research* 22(3):557–567.

[4] Gurevich A, Saveliev V, Vyahhi N, Tesler G (2013) Quast: quality assessment tool for genome assemblies. *Bioinformatics* 29(8):1072–1075.

[5] Guig R et al. (2006) Egasp: the human encode genome annotation assessment project. *Genome biology* 7 Suppl 1:S2.131.

[6] Eilbeck K, Moore B, Holt C, Yandell M (2009) Quantitative measures for the management and comparison of annotated genomes. *BMC Bioinformatics* 10(1):67.

[7] Denton JF et al. (2014) Extensive error in the number of genes inferred from draft genome assemblies. *PLoS Comput Biol* 10(12):1–9.

[8] Keibler E, Brent M (2003) Eval: A software package for analysis of genome annotations. *BMC Bioinformatics* 4(1):50.

[9] Wang J, Kraemer E (2003) GFPE: gene-finding program evaluation. *Bioinformatics* 19(13):1712–1713.

[10] Burset M, Guigó R (1996) Evaluation of gene structure prediction programs. *Genomics* 34(3):353 – 367.

[11] Zhao XM, Wang Y, Chen L, Aihara K (2008) Gene function prediction using labeled and unlabeled data. *BMC Bioinformatics* 9(1):57.

[12] GFF3 Specification, The Sequence Ontology Project http://www.sequenceontology.org/gff3.shtml.

[13] GenomeTools library http://genometools.org.

[14] Steinbiss S, Gremme G, Schrfer C, Mader M, Kurtz S (2009) AnnotationSketch: a genome annotation drawing library. *Bioinformatics* 25(4):533–534.

[15] Bron C, Kerbosch J (1973) Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM* 16:575–577.

[16] Holt C, Yandell M (2011) MAKER2: an annotation pipeline and genome-database management tool for second-generation genome projects. *BMC Bioinformatics* 12(1):491.

[17] GBrowse: the generic genome browser http://gmod.org/wiki/GBrowse.

[18] Duvick J et al. (2008) PlantGDB: a resource for comparative plant genomics. *Nucleic Acids Research* 36(suppl 1):D959–D965.

[19] Stanke M, Waack S (2003) Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics* 19(Suppl 2):ii215–ii225.

[20] Guigó R (2000) An assessment of gene prediction accuracy in large dna sequences. *Genome Research* 10(10):1631–1642.

[21] EMBL nucleotide sequence database http://www.ebi.ac.uk/embl/.

[22] Genome Informatics Research Lab, Institut Municipal d'Investigació Mèdica http://genome.imim.es/datasets/gpeval2000/.

[23] Ensembl project http://ensembl.org.