

---

# Development of computational techniques for genomic data analysis and visualisation in model and non-model organisms

---

A thesis submitted to the School of Biological Sciences at the  
University of East Anglia in partial fulfillment of the requirements  
for the degree of PhD by Publication

Anil Shantilal Thanki

2019

© This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that use of any information derived there from must be in accordance with current UK Copyright Law. In addition, any quotation or extract must include full attribution.

## Abstract

This thesis describes the work undertaken by the author between 2011 and 2018.

With technological development, genome sequencing became affordable and accessible to the scientific communities. This led to the generation of an enormous amount of genomic data and bioinformatics tools to analyse and visualise these data. However, most of the public resources are designed for model organisms, and gold standard curated genomes. These tools are designed to run in a specifically configured environment as well as dependent on specific data formats. Chapter 1 of my thesis introduces the state of the field, the existing tools, their functionalities, and their limitations that prompted the software developments presented in the following chapters.

In chapter 2, I discuss the TGAC Browser, an open-source genome browser and wigExplorer, a BioJS plugin to visualise expression data. In chapter 3, I move towards finding gene families using GeneSeqToFamily, a Galaxy workflow based on the EnsemblCompara GeneTree pipeline. In chapter 4, I focus on a tool developed for visualisation of gene families - Aequatus, an open-source homology browser and ViCTreeView, a plugin developed as a part of the ViCTree project to visualise and explore phylogenetic trees.

In chapter 5, I discuss the availability and accessibility of these tools. All the tools and workflows I have developed are open-source, under a free licence, and are available in GitHub and/or the Galaxy ToolShed. I will also discuss the impact that these tools have made on various research projects. I also take this opportunity to discuss the possibilities of future developments of these tools.

## **Access Condition and Agreement**

Each deposit in UEA Digital Repository is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the Data Collections is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form. You must obtain permission from the copyright holder, usually the author, for any other use. Exceptions only apply where a deposit may be explicitly provided under a stated licence, such as a Creative Commons licence or Open Government licence.

Electronic or print copies may not be offered, whether for sale or otherwise to anyone, unless explicitly stated under a Creative Commons or Open Government license. Unauthorised reproduction, editing or reformatting for resale purposes is explicitly prohibited (except where approved by the copyright holder themselves) and UEA reserves the right to take immediate 'take down' action on behalf of the copyright and/or rights holder if this Access condition of the UEA Digital Repository is breached. Any material in this database has been supplied on the understanding that it is copyright material and that no quotation from the material may be published without proper acknowledgement.

# Contents

---

<b>Abstract</b>	<b>1</b>
<b>Contents</b>	<b>2</b>
<b>List of Figures</b>	<b>6</b>
<b>List of Tables</b>	<b>8</b>
<b>List of Listings</b>	<b>9</b>
<b>List of accompanying material</b>	<b>10</b>
<b>Acknowledgements</b>	<b>11</b>
<b>1 Introduction</b>	<b>12</b>
1.1 Genomics and bioinformatics . . . . .	12
1.2 Democratisation of sequencing technologies . . . . .	13
1.3 Model vs Non-model organism . . . . .	14
1.4 Growth of genomic data and bioinformatics resources . . . . .	14
1.5 Phylogenetic analysis . . . . .	16
1.6 Democratisation of bioinformatics resources . . . . .	17
1.7 Bioinformatics software design . . . . .	18
1.7.1 Standalone tools . . . . .	18
1.7.2 Web-based tools . . . . .	18
1.7.3 Web or standalone application? . . . . .	19
1.7.4 Client-server architecture . . . . .	19



<i>CONTENTS</i>	3
1.8 Data visualisation . . . . .	22
1.8.1 Standalone vs web-based visualisation . . . . .	25
1.8.2 Visualising genomic annotations with genome browsers . . . . .	27
1.9 Bioinformatics Pipeline - Workflow . . . . .	28
1.10 Galaxy: a platform for interactive genome analysis . . . . .	30
1.10.1 Workflow . . . . .	30
1.10.2 Dataset collections . . . . .	32
1.10.3 Visualisation plugins . . . . .	32
1.10.4 Galaxy ToolShed . . . . .	32
1.11 Summary . . . . .	33
<b>2 Enhancing visualisation approaches for genomic annotation</b>	<b>34</b>
2.1 TGAC Browser: an open-source genome browser for non-model organisms	35
2.1.1 Software design . . . . .	35
2.1.2 Data visualisations . . . . .	37
2.1.3 Interface features . . . . .	37
2.1.4 Data upload . . . . .	39
2.1.5 Sequence similarity search . . . . .	41
2.1.6 Summary . . . . .	42
2.2 wigExplorer, a BioJS component to visualise wig data . . . . .	43
2.2.1 Implementation . . . . .	43
2.2.2 Interface features . . . . .	44
2.2.3 Use case . . . . .	45
2.2.4 Summary . . . . .	46
<b>3 Annotation and characterisation of gene families using Galaxy</b>	<b>47</b>
3.1 The EnsemblCompara GeneTrees pipeline . . . . .	47
3.2 GeneSeqToFamily: a Galaxy workflow to find gene families based on the EnsemblCompara GeneTrees pipeline . . . . .	48
3.2.1 GeneSeqToFamily input data . . . . .	49
3.2.2 GeneSeqToFamily workflow . . . . .	50
3.2.3 Supplementary workflows . . . . .	57
3.2.4 Example . . . . .	58

CONTENTS	4
3.2.5 Benchmarking . . . . .	60
3.2.6 Current issues . . . . .	61
3.2.7 Summary . . . . .	63
<b>4 In-depth interrogation of phylogeny through new visualisation tools</b>	<b>65</b>
4.1 Multi-species genome browsers . . . . .	66
4.1.1 Ensembl genome browser . . . . .	66
4.1.2 Genomicus . . . . .	67
4.2 Aequatus: an open-source homology browser . . . . .	67
4.2.1 Software design . . . . .	68
4.2.2 Interface features . . . . .	71
4.2.3 Visualising synteny . . . . .	72
4.2.4 Visualising gene trees . . . . .	75
4.2.5 Visualising one-to-one and one-to-many relationships . . . . .	81
4.2.6 Aequatus.js, a reusable JavaScript plugin . . . . .	83
4.2.7 Ensembl REST API integration . . . . .	86
4.2.8 Exporting visualisations . . . . .	86
4.2.9 Persistent URL . . . . .	86
4.2.10 Summary . . . . .	87
4.3 ViCTree: an automated framework for taxonomic classification from protein sequences . . . . .	88
4.3.1 ViCTree pipeline . . . . .	89
4.3.2 ViCTreeView . . . . .	89
4.3.3 Use Case . . . . .	94
4.3.4 Summary . . . . .	95
<b>5 Discussion</b>	<b>96</b>
5.1 Available and accessible open-source tools . . . . .	96
5.2 Use cases and impacts . . . . .	98
5.3 Limitations and opportunities for future developments . . . . .	101
5.3.1 Analyse and visualise large datasets . . . . .	101
5.3.2 Quality check in GeneSeqToFamily using <i>Gblocks</i> . . . . .	102
5.3.3 Homology identification in GeneSeqToFamily . . . . .	102

<i>CONTENTS</i>	5
5.3.4 Ensembl REST extension in Aequatus . . . . .	103
5.3.5 Discovery and visualisation of exon duplication . . . . .	103
5.3.6 Containerisation for software . . . . .	104
5.4 Conclusion . . . . .	104
5.4.1 Simple recommendations for writing software/tools for biologists	105
<b>Acronyms</b>	<b>106</b>
<b>References</b>	<b>109</b>
<b>Appendix I: Letters of Support</b>	<b>125</b>
<b>Appendix II: Publications submitted</b>	<b>135</b>

# List of Figures

---

1.1	Introduction: EMBL-EBI disk storage . . . . .	15
1.2	Introduction: Types of client-server computing . . . . .	21
1.3	Introduction: REST API . . . . .	22
1.4	Introduction: Common visual properties . . . . .	23
1.5	Introduction: Examples of marks . . . . .	24
1.6	Introduction: Effectiveness of channels . . . . .	24
1.7	Introduction: An example of good and bad visualisation . . . . .	25
1.8	Introduction: Galaxy home page . . . . .	31
2.1	TGAC Browser: Software design . . . . .	36
2.2	TGAC Browser: Visualisations of genomic data . . . . .	38
2.3	TGAC Browser: Overview . . . . .	40
2.4	TGAC Browser: <i>BLAST</i> search from sequence . . . . .	41
2.5	TGAC Browser: <i>BLAST</i> search from annotation . . . . .	42
2.6	wigExplorer: Example . . . . .	45
2.7	wigExplorer: Integration into the TGAC Browser . . . . .	45
3.1	GeneSeqToFamily: Workflow overview . . . . .	51
3.2	GeneSeqToFamily: Workflow with intermediate steps . . . . .	52
3.3	GeneSeqToFamily: Database schema . . . . .	56
3.4	GeneSeqToFamily: Visualising BRAT1 gene family within Galaxy . . . . .	57
3.5	GeneSeqToFamily: Benchmarking . . . . .	62
4.1	Ensembl: Gene tree view example of BRAT1 . . . . .	66

4.2	Genomicus: BRAT1 example . . . . .	67
4.3	Aequatus: Software design . . . . .	68
4.4	Aequatus: Ensembl Comparison schema . . . . .	69
4.5	Aequatus: AJAX model . . . . .	71
4.6	Aequatus: Overview . . . . .	73
4.7	Aequatus: Gene order view . . . . .	74
4.8	Aequatus: Gene tree view comparison with Ensembl . . . . .	76
4.9	Aequatus: Gene tree view with pop-up . . . . .	77
4.10	Aequatus: Pairwise alignment . . . . .	80
4.11	Aequatus: Protein domains . . . . .	81
4.12	Aequatus: Sankey view . . . . .	82
4.13	Aequatus: Tabular view . . . . .	84
4.14	ViCTree: Pipeline overview . . . . .	90
4.15	ViCTree: ViCTreeView data structure . . . . .	91
4.16	ViCTree: ViCTreeView example . . . . .	93
4.17	ViCTree: ViCTreeView distance example . . . . .	94

# List of Tables

---

1.1	Introduction: Bioinformatics pipelines and workflows . . . . .	29
3.1	GeneSeqToFamily: Galaxy tools for the workflow . . . . .	49
3.2	GeneSeqToFamily: Benchmarking parameter set . . . . .	60
3.3	GeneSeqToFamily: Benchmarking results . . . . .	61
4.1	Aequatus: Comparison of phylogenetic visualisation tools . . . . .	87
4.2	ViCTree: Use case . . . . .	95
5.1	Discussion: Downloads of Ensembl tools from the Galaxy ToolShed . . .	99

# List of Listings

---

2.1	wigExplorer: variableStep wig example . . . . .	43
2.2	wigExplorer: fixedStep wig example . . . . .	44
2.3	wigExplorer: Code to configure component . . . . .	44
2.4	wigExplorer: Code to update component . . . . .	44
3.1	GeneSeqToFamily: Default <i>BLASTP</i> parameters . . . . .	53
3.2	GeneSeqToFamily: Default <i>hcluster_sg</i> parameters . . . . .	54
3.3	GeneSeqToFamily: CIGAR example . . . . .	57
3.4	GeneSeqToFamily: example Gene IDs for demo . . . . .	59
3.5	GeneSeqToFamily: example species names for demo . . . . .	59
4.1	Aequatus: Input data structure for Aequatus.js . . . . .	85
4.2	Aequatus: Code to configure Aequatus.js . . . . .	85
4.3	Aequatus: Code to update Aequatus.js . . . . .	85
4.4	ViCTree: Code to configure ViCTreeView . . . . .	92

## List of accompanying material

**Publications submitted for this PhD by Publication:** This thesis is based on the manuscripts listed in chronological order below, I have described my contribution to each manuscript in relevant chapters. My role in these publications was to define, develop and apply the analysis and visualisation techniques I present in this thesis. I was responsible for all aspects of the software development, and I wrote manuscripts in which I am the first author and provided input into writing and editing each of the manuscripts in which I am a co-author.

In Appendix I, there are letters of support from some of my co-authors and, in Appendix II, the original publications are reproduced with the kind permission of the relevant journals.

**A. S. Thanki**, R. C Jimenez, G. G. Kaithakottil, M. Corpas, and R. P. Davey “wig-Explorer, a BioJS component to visualise wig data,” F1000Research 2014

M. Spannagl, M. Alaux, M. Lange, D. M. Bolser, [and 25 others, including **A. S. Thanki**.] “transPLANT Resources for Triticeae Genomic Data,” The Plant Genome 2015

**A. S. Thanki**, N. Soranzo, W. Haerty, and R. P. Davey, “GeneSeqToFamily: a Galaxy workflow to find gene families based on the Ensembl Compara GeneTrees pipeline,” GigaScience 2018

S. Modha, **A. S. Thanki**, S. F. Cotmore, A. J. Davison, and J. Hughes, “Victree: an automated framework for taxonomic classification from protein sequences,” Bioinformatics 2018

**A. S. Thanki**, N. Soranzo, J. Herrero, W. Haerty, and R. P. Davey, “Aequatus: an open-source homology browser,” GigaScience 2018 - Selected in GigaScience Prize Track

**A. S. Thanki**, X Bian, and R. P. Davey, “TGAC Browser: An open-source genome browser for non-model organisms,” bioRxiv 2019



## Acknowledgements

In this thesis, I described my work over seven years of my research experience at the Earlham Institute. I have been very fortunate to work with some very talented individuals, some of whom get a mention below.

First of all, I would like to thank my supervisors Robert Davey and Wilfried Haerty without their support I would not have begun this PhD and thankful to them for their time, patience and guidance while assembling this thesis and support for the work included in this thesis. I am also grateful for the support of Neil Hall, Sarah Cossey, Anthony Hall and Federica Di Palma. A special thanks to Darren Heavens, who has been my guide even before the beginning of my PhD and advising me all the way through. Other special thanks to Catherine Hunter for organising periodic meetings from time to time.

I want to thank Nicola for teaching me inside out of Galaxy. I am also thankful to the rest of the team members and lunch buddies Alice, Evanthia, Felix, Martin, Simon, Toni, X-man and Gemy for making my time at work enjoyable.

The list of colleagues who have influenced during my time EI will go on so, I apologise that I cannot name you all individually.

I would also like to thank all my publication co-authors, especially those providing me with letters of support and furnishing them with such kind words.

Especially I would like to thank my wife for having faith in me, so I undertook this commitment.

# 1

## Introduction

---

### 1.1 Genomics and bioinformatics

Following the discovery of DNA structure in 1953 [1], nucleic acid sequencing became a major target of early molecular biologists [2]. Fred Sanger [3], Allan Maxam and Walter Gilbert [4] pioneered the practice of DNA sequencing, whereas Alec Jeffreys developed a method for DNA profiling [5]. These techniques have been widely applied in medicine, biotechnology, forensic science, as well as social sciences [6], generating a large amount of data requiring computers for analyses. The introduction of computers also led to the automation of techniques reducing costs, increasing output, therefore enabling scaling studies from a single gene to thousands and now the whole genome<sup>1</sup> of an organism. This progress opens up the field of genomics which encompasses the study of an organism's genome. Genomics was initially dedicated to the large scale investigation of DNA sequences has also quickly expanded toward functional levels [7, 8]. As genomics developed as a novel research field, it necessitated the accelerated development and application of bioinformatics, an interdisciplinary field that stores, retrieves and analyses large amounts of biological information [9].

Most bioinformatics tools are command-line based and generate binary or text-based data as output. For a non-computer scientist or biologist, this can be a significant challenge due to a huge learning curve to apply the required tools to perform even a small analysis.

Therefore, in this thesis, I focus on the democratisation of bioinformatics resources so biologists can perform complex analysis with minimal or no computing knowledge.

---

<sup>1</sup>Genome is a complete set of DNA sequence including both coding and non coding regions of an organism.

## 1.2 Democratisation of sequencing technologies

Over the last 15 years, technology has transformed genome sequencing completely. Sequencing technologies have remarkably progressed since the Human Genome Project, which was completed in 2003 at the cost of \$3 billion [10, 11]. This project also included the sequencing of *Drosophila melanogaster* and *Caenorhabditis elegans*, two important organisms for research.

Per base cost of sequencing with first generation sequencing platforms (e.g. Sanger sequencers) were high (approx \$2400 per million bases), sequencing runs were long relative to nucleotide output, and provided limited throughput (about tens of kilobase pairs per run) [12]. In the mid-2000s, Second-generation sequencing platforms, more widely known as Next-Generation Sequencing (NGS) Platforms (e.g. Roche 454, ABI SOLiD, and Illumina), were released reducing the run times and costs (approx \$0.13 per million bases) and increasing throughput (about hundreds of gigabase pairs per run) [11, 12]. However, the drive to reduce cost focused on the race to the ‘\$1,000 human genome’, with experimental footprints, workflows, reagent costs and run times poorly matching the needs of small laboratories studying non-standard genomes [13]. At the 2012 Advances in Genome Biology and Technology conference [14], the Oxford Nanopore Technologies (ONT) announced MinION, an advanced single-molecule sequencing technology offering multi-kilobase reads. The MinION attracted interest due to its compact size, Universal Serial Bus (USB) connection, relatively inexpensive purchase price and a streamed mode of operation that enables real time analysis of data as it generated (e.g. What’s In My Pot - WIMP workflow) [15]. The ultra-low-cost and mobile nature of the MinION device open up a considerable number of applications i.e. in-field sequencing (Ebola virus [16] and Zika virus[17]), and sequencing through the centromere [18].

In addition to sequencing the whole genome of an organism, preparation-specific sequencing technologies (i.e. RNA-seq, exome sequencing) were also developed to sequence coding and non-coding genes as well as regulatory non-coding sequences to analyse the genes’ regulation and function.

All these various sequencing technologies have delivered a step change in our ability to

sequence genomes.

### 1.3 Model vs Non-model organism

Model organisms (e.g. fruit fly, *Escherichia coli*, yeast, zebrafish, human, and mouse) are extensively studied and have led to the development of a vast array of conventional genetic techniques. Discoveries made in model organisms are expected to provide insights into the biology and physiology of other organisms.

However, there is a disadvantage of just focusing on model organisms for developing resources. They are not perfect examples of diversity compared to their close relatives. Because model organisms are specifically selected for a particular experimental rationale and not randomly spread in the tree of life, there is much we can learn from the less-well-studied organisms.

With the democratisation of genome sequencing technologies, where an increased number of researchers have access to more sequencing data, the focus is shifting from model organisms to non-model organisms, which may not be selected for extensive studies because of long life cycle, inability to grow *in-vitro* environment, or limited genetic resources. Small scale labs and research groups are now able to study non-model organisms by performing relatively cheap sequencing and address a wide array of essential and diverse questions focused on different aspects of biology.

### 1.4 Growth of genomic data and bioinformatics resources

Genome sequencing is the first step of genome analysis. Current genome sequencing technologies (see section 1.2) read nucleic acid fragments as pieces depending on the technology used to produce sequence reads. Thus genome assembly is required to reconstruct the original sequence by aligning and merging smaller fragments into longer ones (contigs and unitigs). Genome annotation is also necessary to confer biological information to the assembled sequences.

Similar to genome sequencing developments, bioinformatics has become one of the fastest growing fields and a vital element of biological research, because it allows to

analyse a huge amount of biological data quickly and cost-effectively [19]. Thus, various bioinformatics tools have been developed to perform assembly (e.g. Abyss<sup>2</sup> [20], SOAPdenovo<sup>3</sup> [21], DISCOVAR [22]), genome annotation (e.g. GENSCAN [23], ORF Finder [24], novoSNP [25]), and sequence similarity (e.g. BLAST<sup>4</sup> [26], HMMER [27]), as well as to solve many other analytical problems.

The advances in sequencing technologies and associated bioinformatics tools resulted in a rapid increase in the volume of raw and interpreted biological data (see Figure 1.1). The number of databases containing highly curated genomic information for model organisms has similarly increased (e.g. Ensembl [28], FlyBase [29], and Wormbase [30]). With the development of annotation tools which also provides functional information such as protein domains, transcription factors binding sites, and single nucleotide as well as structural variations, the main challenge is how to store, process, analyse, and visualise an ever-increasing amount of newly sequenced biological data.

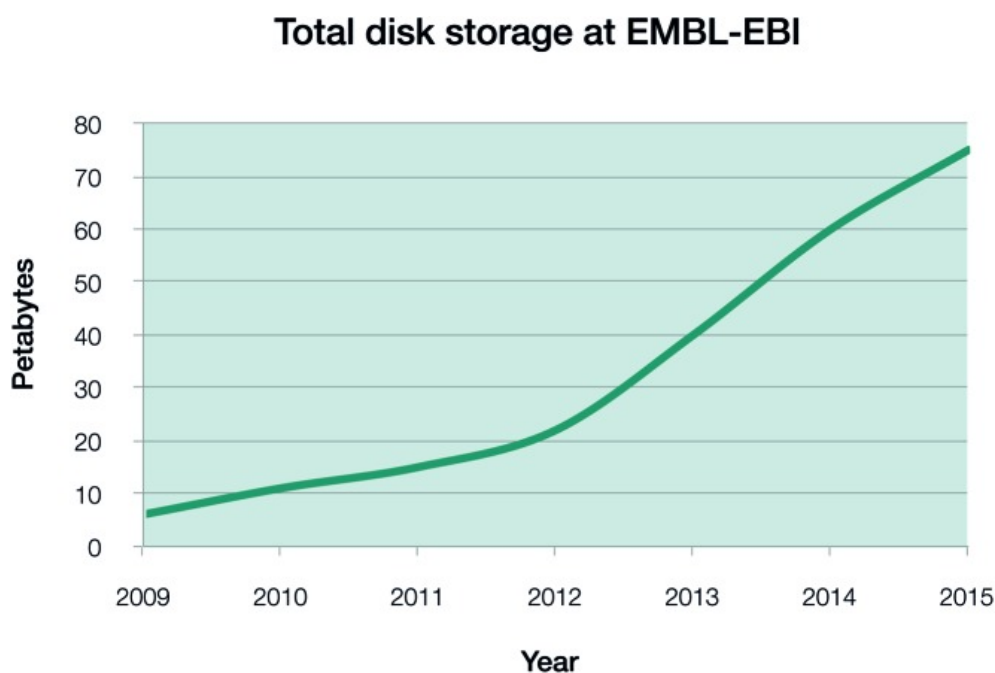


Figure 1.1: Showing disk-storage at EMBL-EBI from 2009 to 2015. These figures include all installed storage, counting multiple backups for all data resources as well as available storage. The actual total volume of a single copy of all data resources is roughly 1/3 of total installed storage capacity.

Reproduced from Cook et. al 2016 [31]

<sup>2</sup>Assembly By Short Sequencing (Abyss)

<sup>3</sup>Short Oligonucleotide Alignment Program (SOAP)

<sup>4</sup>Basic Local Alignment Search Tool (BLAST)

## 1.5 Phylogenetic analysis

Phylogenetic analysis is the study of estimating the evolutionary relationships of entities (e.g. species, genes, and sequences) [32]. Classic phylogenetic analysis dealt mainly with physical, or morphological features like size, and colour [33]. While modern phylogenetic analysis uses genetic information and it has become an important method for studying the evolutionary history of organisms [34].

There are various techniques available to perform phylogenetic analysis. One of the most accepted methods is the study of gene families expansion and contraction across species.

A gene family is a collection of several homologous genes, formed by duplication of a single original gene. Most of the genes in a single gene family are presumed to have related functions depending on sequence identity. With the possibility of sequencing non-model organisms, comparative study of newly sequenced genomic data with existing gold standard genomic resources became possible. Finding a gene family for newly annotated genes from non-model organisms can help to identify candidates for follow up study as well as experimental validation.

Various bioinformatics methods are available to infer the homology, shared ancestry between a pair of structures, or genes belonging to a gene family. These methods can be generally classified into sequence comparison based and tree-based. The sequence-based methods are much faster and provide pairwise ortholog relationships while tree-based methods are a good choice for finding fine-grained distinction among many-to-many, one-to-many and one-to-one relationships. Sequence comparison based methods include InParanoid [35], OrthoDB [36], and OrthoMCL [37]. Tree-based phylogenetic methods include PhyOP [38] and TreeFam [39]. The third category of hybrid approaches uses both sequence comparison, and phylogenetic methods, they are useful to construct clusters of homologous genes and determine trees, such as Ortholuge [40], and EnsemblCompara GeneTrees [41].

Despite, having many alternatives available, most of these tools and pipelines are command-line based, designed to run in a specifically configured environment and requires various dependencies to run successfully.

## 1.6 Democratisation of bioinformatics resources

An overwhelming amount and diversity of new analytical algorithms packaged as software tools [42] are being developed as all aspects of sequencing data analysis rely on bioinformatics tools [43]. Whilst the rapid development of resources is beneficial, but only consistently accessible software provides a foundation for the reproducibility of published research, defined as the ability to replicate published findings by running the same computational tool on the data generated by the study [44, 45].

Traditionally, most of the bioinformatics tools have been made available from institutional servers. However, not all published bioinformatics tools are accessible; for example, 24,490 omics software resources were published from 2000 to 2017 from which 26% is not currently accessible through Uniform Resource Locator (URL) published in the original paper. Whilst, this percentage is declining, 200 unstable resources are still published each year [46].

To improve stability, these resources can also be made available through open-source repositories, such as GitHub<sup>5</sup>, SourceForge<sup>6</sup>, and Bitbucket<sup>7</sup>. These repositories are Distributed Concurrent Versions System (DCVS), which stores the current version(s) of a project and its history as well as allows several developers to work on the same project concurrently, that help the community to contribute to the resources as well as identify and solve software issues.

Despite, tools are available to download and install either from the institutional server or from the open-source repositories, research conducted by Mangul et al. shows that almost half of the 99 randomly selected tools were challenging to install due to the implementation problems [46]. To mitigate this issue, tools can be made available through a package manager (e.g. Bioconda<sup>8</sup> [47]), which makes it easier to install, upgrade and configure the software.

---

<sup>5</sup>GitHub is a web-based hosting service for version control using Git, an open-source distributed version control system.

<sup>6</sup>SourceForge is a web-based service that offers a centralised online location to control and manage free and open-source software projects.

<sup>7</sup>Bitbucket is a web-based version control repository hosting service for source code and development projects that use either Mercurial or Git version control systems.

<sup>8</sup>Bioconda is a channel for the conda package manager, an open-source package and environment management system, specialising in bioinformatics software.

## 1.7 Bioinformatics software design

Similar to many other software, bioinformatics tools follow similar software design strategies. Generally, software/tools are divided into two categories: standalone tools and web-based tools.

### 1.7.1 Standalone tools

Standalone tools are designed to be installed on a single machine (a laptop or a desktop) for an individual user or on a High-Performance Computing (HPC) environment for CPU and memory-intensive jobs. Standalone tools tend to be highly optimised to make the best use of computing resources. Examples in bioinformatic are assembling sequence reads (e.g. Abyss, SOAPdenovo), analysing sequencing data (e.g. GENSCAN, BLAST) and visualising genomic annotations (e.g. IGV<sup>9</sup> [48]). However, one of the main drawbacks of standalone tools is platform dependency, i.e. applications can only run under a specific operating system, in a certain type of computer architecture, and requiring other particular tools. To take advantage of the latest version with new features, users need to update the tool manually as the tool is installed and running on a local machine.

### 1.7.2 Web-based tools

Web-based bioinformatics tools and services are designed to reach several users and can be accessed from remote computers with different operating systems solving one of the challenges of standalone tools. This type of tools are generally hosted on an institutional server and uses a website as the front-end interface, allowing users to access the application from any computer connected to the internet using a standard web browser. Web-based tools are generally implemented to make data available hosted at the repository (e.g. Ensembl and JBrowse [49]) or to perform analysis using the command-line tools installed on the host computer (e.g. BLAST, T-Coffee, Galaxy [50], and EMBOSS<sup>10</sup> [51]). Web-based tools can also be installed and used locally,

---

<sup>9</sup>Integrative Genomics Viewer (IGV)

<sup>10</sup>The European Molecular Biology Open Software Suite (EMBOSS)



in which the same computer acts as a server and a client, for example, the Galaxy platform [50] (see section 1.10) install on a server and can be accessed using a standard web browser.

### 1.7.3 Web or standalone application?

There is no straightforward answer to the question of whether to develop web or standalone applications. It is up to the developers to decide which solution suits the data, the computing environment, and any end-users. If the main objective is to develop an easy to deploy and maintainable application that can be accessed from remote computers with different operating systems, then the web-based solution is the right choice. If the main objective is to have a fast and secure application that caters for single users on a local system, then the standalone implementation is a more suitable choice.

Importantly, however, an increasing number of standalone tools are now implemented in web environments targeting a wider audience, thus increasing accessibility.

### 1.7.4 Client-server architecture

The client-server model describes how a server provides resources to client devices, such as computers, tablets, and smartphones. Most servers have a one-to-many relationship with clients, in which a single server (or a group of mirrored servers) provides resources to multiple clients at one time. Some examples of servers are web servers, mail servers, and file servers.

One of many objectives for bioinformatics applications is to make data available from central repositories. For example, genome browsers (e.g. Ensembl, JBrowse) are used to retrieve data from data repositories connected to the server. Therefore, I am describing the following client-server architectures (see Figure 1.2) regarding data access:

1. 2-tier architecture:

In 2-tier architecture, clients and servers communicate with each other without any intermediate point or node. Because of the tight coupling of both the tiers, the application tends to run faster when serving a small number of users. As the

number of users increases, performance will be affected, because the server needs to respond to multiple requests at the same time [52, 53].

2. 3-tier architecture:

In 3-tier architecture, an additional middle-tier sits in between clients and servers, with the purpose of handling application execution and data access. This type of architecture benefits from enhanced security compared to 2-tier architecture because the client has no direct access to the database. However, it comes with increased communication complexity as the number of communication points increase [52, 53].

3. n-tier architecture:

This type of architecture is also known as a multi-tier architecture or Component-based architecture. It is an expanded form of 3-tier architecture where tiers can be physically and/or topologically separated. Therefore, this architecture gives flexibility to developers to create or implement modular application components. However, due to the tiering of components, these types of applications could show disadvantages through issues such as system evolution, compatibility, and migration [52, 53].

Each type of computing architecture has pros and cons, and tend to be suitable for specific needs. However, in modern applications, 3-tier and n-tier computing architectures are more dominant than 2-tier because of their key factors like better security, re-usability and flexibility.

## **REST API**

To access the data or service programmatically, web-based applications may provide a REST API [55].

An Application Programming Interface (API) is a set of rules that allow programs to talk to each other. It does not define data formats, but is usually associated with exchanging text-based information between a client and a server.

Representational State Transfer (REST) is an architectural style for developing web

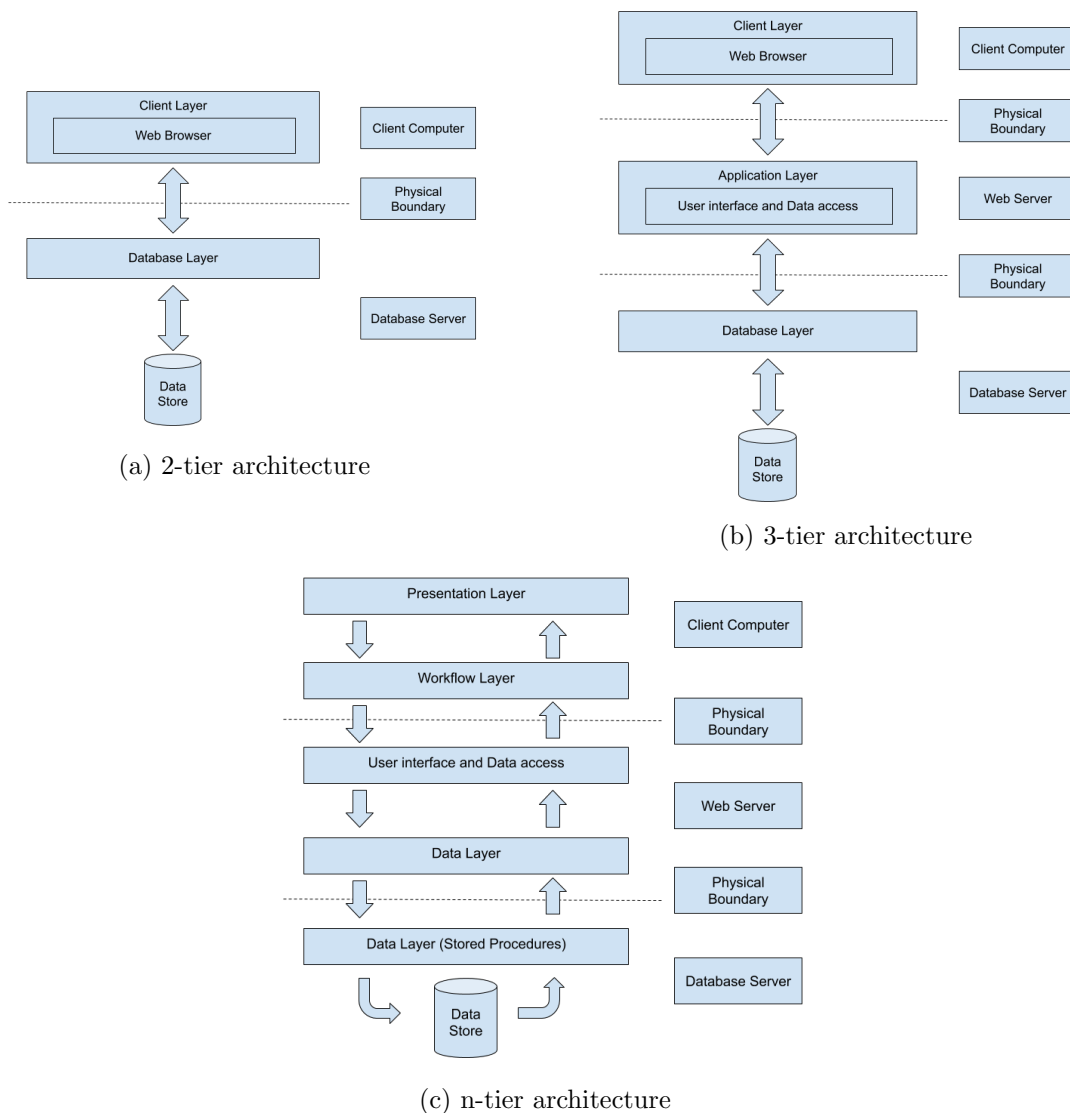


Figure 1.2: Diagram demonstrating the types of client-server computing a) 2-tier architecture, b) 3-tier architecture, and c) n-tier architecture [54]

services defining how applications communicate over the internet using the Hypertext Transfer Protocol (HTTP), which allows applications to transfer information quickly and efficiently (see Figure 1.3).

An example of a REST API for bioinformatics application is the Ensembl REST services [56] (<https://rest.ensembl.org/>), which provides multiple endpoints to access diverse data programmatically.

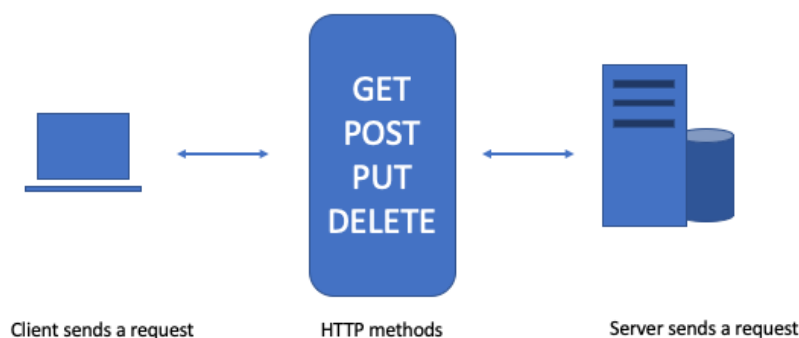


Figure 1.3: Showing REST API architecture

## 1.8 Data visualisation

According to the definition provided in 1987 at the National Science Foundation's Visualisation in Scientific Computing Workshop report: '[v]isualization offers a method for seeing the unseen' [57].

With advances in biological data acquisition, data management and data processing technologies, researchers face challenges of developing hypotheses from data that continues to increase in volume and complexity. With this, data visualisation can play a vital part in the biological sciences to record information, analyse data to support reasoning and interpretation, as well as to communicate information [58].

There has been an increase in the number and variety of tools for visualising biological data such as, genome browsers (e.g. Gbrowse [59], JBrowse [49]), sequence alignment viewers (e.g. Jalview [60]), phylogenetic tree viewers (e.g. PHYLOViZ [61], TreeView [62]). These tools offer a wide range of functionalities and different degrees of scalability, some with interactive visualisation.

A major challenge for bioinformatics data visualisation tools is to find a compelling visual presentation of size and types of datasets. Different techniques for representing data can affect the interpretation of the results and conclusions of an experiment [63]. Good data visualisation practices should place meaning into complicated (multi-layer

and interconnected) datasets so that their message is clear and concise. A number of guidelines have been established for a visualisation to be considered effective.

Julie Steele and Noah Iiinsky described in chapter 4 of their book on “Designing Data Visualizations” [64] that different visual properties which can be modified in several ways make them suitable for encoding different types of data. The factors are categorised as to whether a visual feature is naturally ordered, or has a number of distinct values.

Natural ordering is suitable for quantitative or ordinal differences such as position, length, line thickness or weight properties. Distinct differentiation features are suitable for categorical data such as shape, texture, and line style (e.g. solid, dotted, dashed). Figure 1.4 shows visual properties which can be used to select proper encoding for various data types.

Example	Encoding	Ordered	Useful values	Quantitative	Ordinal	Categorical	Relational
	position, placement	yes	infinite	Good	Good	Good	Good
1, 2, 3; A, B, C	text labels	optional alpha or num	infinite	Good	Good	Good	Good
	length	yes	many	Good	Good		
	size, area	yes	many	Good	Good		
	angle	yes	medium	Good	Good		
	pattern density	yes	few	Good	Good		
	weight, boldness	yes	few		Good		
	saturation, brightness	yes	few		Good		
	color	no	few (<20)			Good	
	shape, icon	no	medium			Good	
	pattern texture	no	medium			Good	
	enclosure, connection	no	infinite			Good	Good
	line pattern	no	few				Good
	line endings	no	few				Good
	line weight	yes	few		Good		

Figure 1.4: Showing common visual properties for the selection of an appropriate encoding based on data type.

Reproduced from Steele et. al 2011 [64]

Similarly, Tamara Munzner describes ‘Marks’ (see Figure 1.5) and ‘visual channels’ (see Figure 1.6) as basic graphical elements and controllers for their appearance in her book “Visualization Analysis Design” [65]. The effectiveness of a channel for encoding data depends on its type. For example, channels that perceptually convey magnitude infor-

mation are a good match for ordered data and those that convey identity information with categorical data.

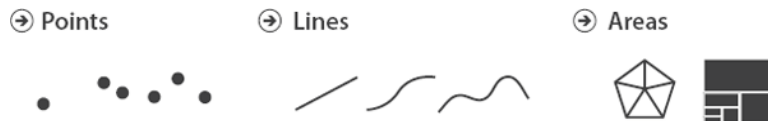


Figure 1.5: Examples of marks: point - a zero-dimensional, line - a one-dimensional mark, and area - a two-dimensional (2D) mark.

Reproduced from Munzner et. al 2015 [65]

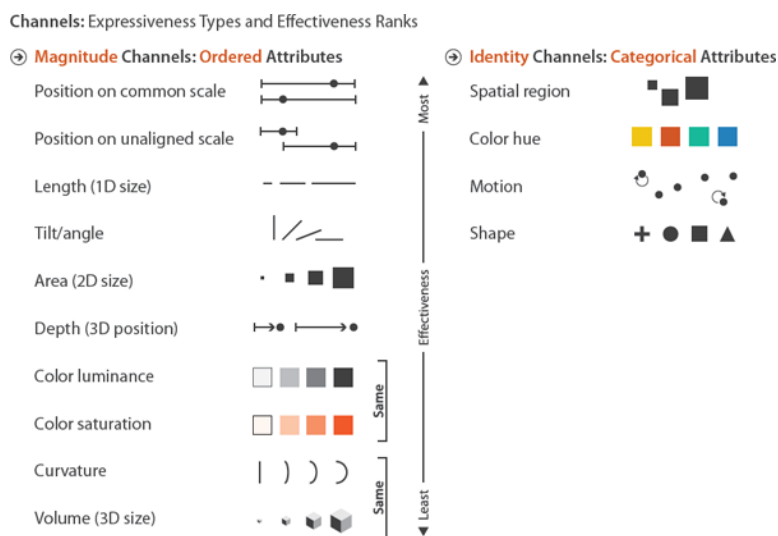


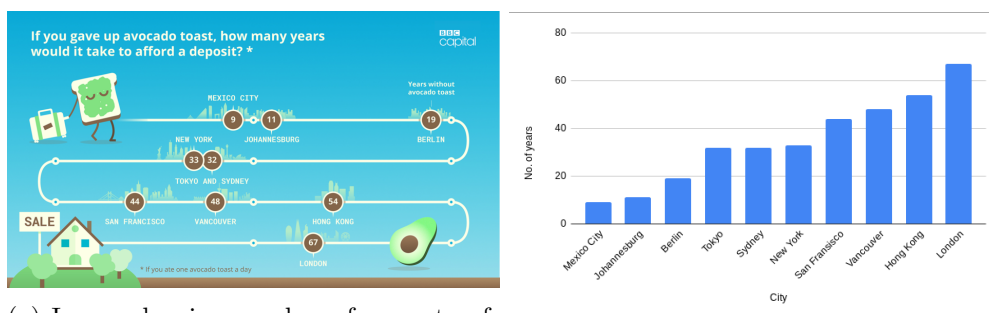
Figure 1.6: Effectiveness of channels according to data types.

Reproduced from Munzner et. al 2015 [65]

Figure 1.6 also shows the effectiveness rankings for visual channels according to the two expressiveness types of ordered and categorical data, ranging from the most effective channels to the least effective from top to bottom. Ordered attributes should be presented with the magnitude channels, while categorical attributes should be presented with the identity channels. However, it is possible to use them interchangeably, but this can lead to misinterpretation and oversimplification.

Figure 1.7a shows one of the five pictures BBC used in an article about how many years would it take to afford a deposit for a house if you give up eating one avocado toast a day. This image looks attractive but does not convey information in a comparative manner, i.e. the difference in years between London and Berlin. This can be easily shown with a straightforward bar graph (see Figure 1.7b).

Here, I have summarised some of the many principles of visual design, but there is



(a) Image showing number of years to afford a deposit. From BBC News website 2017 [66]. (b) A bar chart showing number of years to afford a deposit.

Figure 1.7: Showing an example of good and bad visualisation

more information available on choosing colours, arranging networks and trees, as well as manipulating the view [65, 64].

## Community initiatives

Most biology related visualisation research has been conducted by people who have learned about visualisation but are often not aware of the on-going research in the visualisation community. Typically, the current tools do not embrace the latest advances in design, usability, visualisation principles, and evaluation [67].

International community initiatives (e.g. VizBi<sup>11</sup>, BiVi<sup>12</sup>, and BioVis<sup>13</sup>) bring visualisation experts together with computational biologists, bioinformaticians, graphic designers, animators, and medical illustrators, and aim to raise the global standard of the next generation of tools for visualising biological data.

### 1.8.1 Standalone vs web-based visualisation

Standalone visualisation tools are installed on a local computer to visualise data such as IGV and Jalview. These applications can utilise more computing resources and perform much faster compared to web-based tools. However, they do require specific environments to run in, and they can not reach a wider audience.

In contrast, web-based data visualisation tools allow users to access data hosted on a remote server and to upload their data to visualise. It is therefore ideal to use web-based

<sup>11</sup>Visualising Biological Data (VizBi) (<http://vizbi.org/>)

<sup>12</sup>The Biological Visualisation Network (BiVi) (<https://bivi.co/>)

<sup>13</sup>Biological Data Visualisation (BioVis) (<http://biovis.net/2019/index.html>)

visualisation tools to reach to a broader audience. They can be rich in information as numerous web-based visualisation technologies are now available to create interactive web interface.

### Web-based visualisation technologies

Static websites are written in the HTML markup language, which is useful to some extent, but dynamic website is necessary for data visualisation. There are numerous technologies available to add interactivity to the web interface. Here, I will introduce some of the technologies, which I have used for the development of the tools discussed in this thesis.

- JavaScript - a programming language for the web used to add dynamic behaviour to a website, store information and handle requests and responses.
- jQuery - a JavaScript library used for simplifying the manipulation of the HTML Document Object Model (DOM), such as position, size, and style as well as event handling like binding and unbinding mouse over and click events.
- Scalable Vector Graphics (SVG) - a vector image file format used to generate two-dimensional graphics on the web.
- Data-Driven Documents JavaScript (D3.js) [68] - a JavaScript library for creating dynamic, interactive data visualisations in web browsers using SVG.
- jQuery SVG - a jQuery plugin to manipulate SVG.
- jQuery DataTable - a highly flexible plugin for the jQuery JavaScript library, can be used to create interactive HTML tables.

Using all these popular web technologies, an interactive and information-rich front end can be generated.

**D3.js:** D3.js, also known as D3 or D<sup>3</sup>, is a well-established open-source JavaScript library to create and manipulate documents based on data. D3 uses HTML, SVG and CSS to bring the data to life. D3's emphasis on web standards gives developer the full capacities of modern web browsers without binding to a restrictive framework, combin-



ing visualisation components and a data-driven approach to DOM manipulation [69]. D3 is very popular and widely used because of its vibrant open-source community and availability of a large number of examples. In addition, D3 is a declarative language, in which the developer specifies what needs to be done instead of how to do it, which gives several advantages such as faster iteration, better visualisation, re-usability, portability and better performance [70].

### 1.8.2 Visualising genomic annotations with genome browsers

A genome browser is a tool that provides a graphical interface to visualise and explore the genomic sequence and annotation. Genome browsers play a vital role in analysing data to examine the results and generate hypotheses. Typically genome browsers can be classified into two categories based on whether the image is rendered on the server-side or the client-side [71].

Traditional genome browsers such as University of California Santa Cruz (UCSC) genome browser [72], Ensembl genome browser [28], GBrowse [59] are categorised as server-side image rendering browsers. They retrieve the data from the back-end databases and render them into pictures on the server, and then send the pictures to the client web browsers. More recent genome browsers such as JBrowse [49] are categorised as client-side rendering browsers. They retrieve data from the server and send the data to the client directly in the form of text and draw the pictures dynamically in web browsers. The server-side rendering browsers could be beneficial to be used with a low bandwidth internet connection, but on the other side, client-side rendering browsers reduce the server burden by allocating rendering tasks to client sides.

Genome browsers can also be categorised into species-specific and multi-species.

#### **Species-specific genome browser**

Species-specific genome browsers mainly focus on a single organism and may include multiple layers of genomic annotations, where all annotations are mapped back to a reference sequence such as a chromosome or scaffold. There are many species-specific genome browsers available such as Ensembl genome browser, UCSC genome browser,

Mouse Genome Informatics (MGI) [73], FlyBase [29], and Rice-Map [74], which play a vital role in providing in-depth visualisation of genomic annotations from one or more data source.

### **Multi-species genome browser**

Multi-species genome browsers are built for cross-species comparative analysis by combining sequence data and annotations for more than one organism, and these comparisons are pre-computed. There are many multi-species genome browsers available such as Ensembl, Genomicus [75], and Gbrowse\_syn [76]. Applications of multi-species genome browser vary from tool to tool. Genomicus visualises the order of syntenic genes along with phylogeny (see section 4.1.2). Gbrowse\_syn is based on GBrowse platform and visualises alignments among various species.

Some browsers such as Ensembl genome browser has dual functionality of presenting detailed species-specific genomic information as well as providing an overview of gene tree along with the alignment of coding sequences (CDS) (see section 4.1.1).

Most of these species-specific and multi-species genome browsers require heavily curated data hosted on public repositories, as well as require data in a specific format. There is a need for versatile visualisation tools to be able to visualise genomic data of newly sequenced non-model organisms which are not available in public repositories.

## **1.9 Bioinformatics Pipeline - Workflow**

A workflow or a pipeline is an abstract description of steps needed for executing a particular process, and the flow of information between each step, where each step is performed either by people or by system functions (e.g. computer programs) [77].

A bioinformatics analysis typically involves using two or more tools. Each tool processes input files and produces output files and can require many dependencies such as programming libraries and reference databases. Therefore, to simplify bioinformatics analysis and increase reproducibility, automated pipeline or workflows are becoming popular [77]. Scripts written in the Unix shell or Perl, as well as the Make utility [78],

can be seen as a basic form of pipelines. However, they were not designed specifically for running complete multi-step analysis pipelines, and do not have built-in support for dependency management or restarting the pipeline in case of failure. The Make utility has no built-in support to run parallel tasks on multiple distributed hosts.

In recent years, several new pipeline frameworks have been developed to address these limitations. These systems are explicitly designed to access, manage and process scientific data sets. These systems can use remote computational resources as well as execute a series of computational or data manipulation steps as a workflow. Table 1.1 shows a brief comparison amongst some workflow management systems. The comparison includes basic features for a bioinformatics pipeline, such as whether it is accessible by Command-line interface (CLI) or Graphics User Interface (GUI), its ability to scale to be used with HPC or to utilise cloud infrastructure and support for containers [79]. Bioinformatics analysis is not limited to a single dedicated system or institute; it spans from personal computers to HPC, and cloud computing [80]. Therefore the comparison also includes the support for The Common Workflow Language (CWL), an open standard for describing analysis workflows and tools, which makes pipelines portable and scalable across a variety of software and hardware environments.

Pipeline / Workflow	CLI	GUI	Scheduling System (LSF, SLURM, HTCondor)	Cloud (AWS)	Docker Support	CWL
Snakemake [81]	Y	N	Y	N	Y	Y
Jupyter Notebook [82]	N	Y	Y	Y	Y	Y
Galaxy	N	Y	Y	Y	Y	Y
Apache Taverna [83]	N	Y	Y	N	Y	?

Table 1.1: Comparing well-known platform used for developing bioinformatics pipelines / workflows. As per Google Summer of Code (GSoC) 2016 update, Apache Taverna was working on CWL support. CLI: Command-line Interface, GUI: Graphical User Interface, LSF: Load Sharing Facility, SLURM: Simple Linux Utility for Resource Management, AWS: Amazon Web Services, CWL: Common Workflow Language.

In bioinformatics, several alternative computation tools may be available for a single analysis, and choosing a specific tool can alter the results and affect biological interpretation. Therefore, data analysis pipelines should allow researchers to adapt, understand, experiment with, and integrate new computational tools into their analyses [84].

As there is not a single way of finding out the ‘best’ pipeline framework, the choice of a framework should be informed both by the demands of developing a given pipeline and the requirements of those using it. System-specific configured pipelines are suitable to perform routine bioinformatics analyses, while configurable pipelines that can be accessed remotely and run in the cloud are suitable for collaborative studies [80].

## 1.10 Galaxy: a platform for interactive genome analysis

As sequencing techniques have become affordable, and a large amount of data is being generated, performing in-house bioinformatics analysis and visualisation of genomics data is very important. However, small-scale research groups might not have the necessary bioinformatics expertise or computing infrastructure.

To overcome this issue, the Galaxy platform [50] (<https://galaxyproject.org>) (see Figure 1.8) was developed enabling biologists to perform computational analysis on the web without extensive programming and system administration knowledge. Galaxy allows users to collect and manipulate data from existing resources in a variety of ways, including uploading and fetching data from a remote server. Multi-step analyses can be performed by running tools in succession, and every action of the user is recorded and stored in the history system, a key element of Galaxy enabling reproducibility. This allows users to conduct independent queries on genomic data to combine or refine them, perform calculations, or extract and visualise corresponding sequences or alignments. The Galaxy has various components that make it widely accepted bioinformatics analysis tool.

### 1.10.1 Workflow

The Galaxy is equipped with a workflow system, where multiple tools can be assembled to perform a comprehensive analysis. Galaxy allows users to create multi-step analyses by using a drag and drop workflow editor. Tools can be added and connected so that the output of one tool becomes the input of other tools. Workflows enable automating the repeatedly running comprehensive analyses. Once created, workflows act like tools, and they can be accessed and run from Galaxy’s main analysis interface [85].

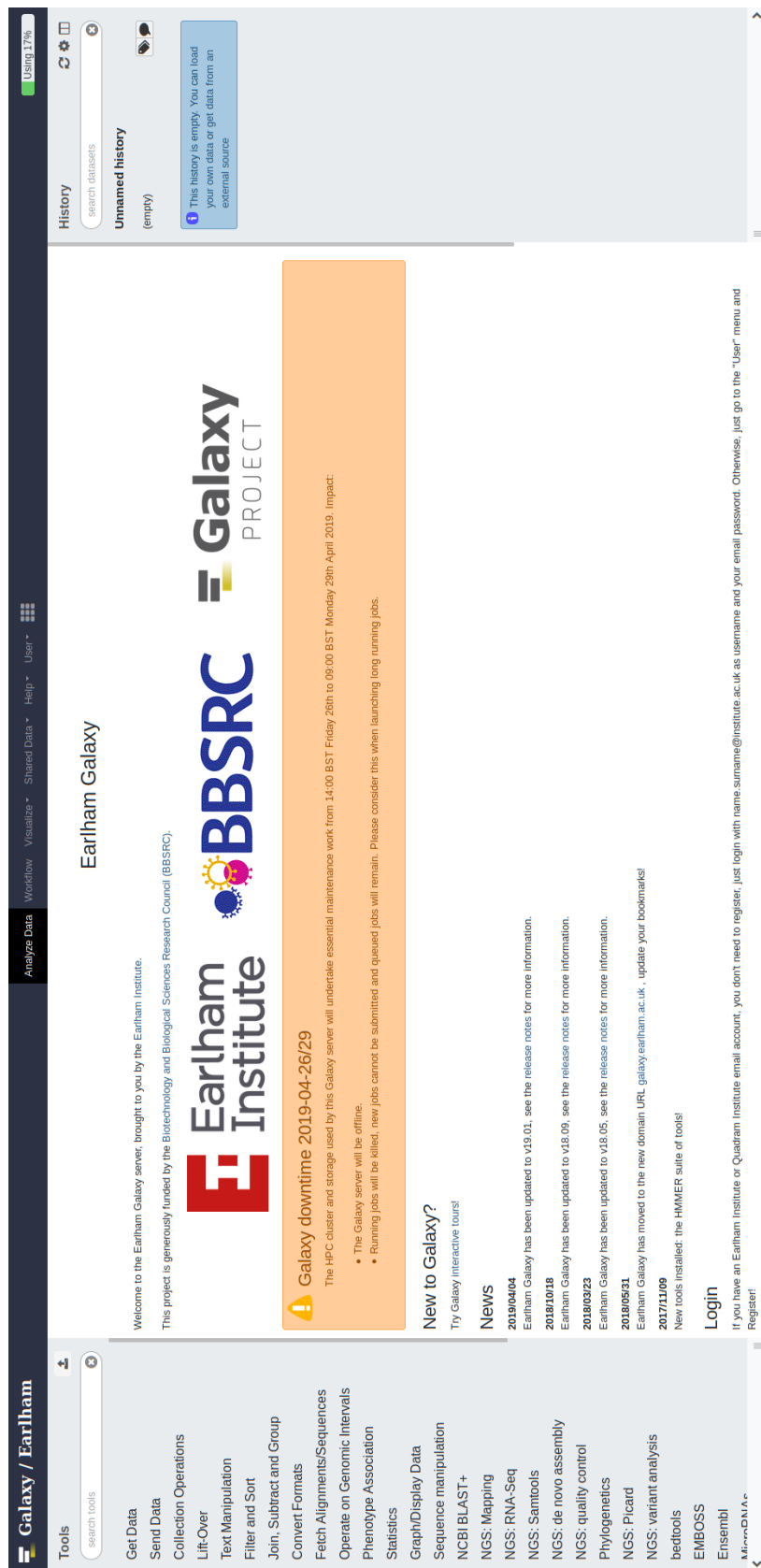


Figure 1.8: Galaxy platform: The main page of a typical Galaxy platform, with available tools listed in the left pane and analysis history in right pane

### 1.10.2 Dataset collections

To facilitate analysis of many datasets concurrently, Galaxy includes dataset collections comprising related datasets and defining their relationships. Currently, Galaxy supports two types of collections: a simple collection of multiple datasets and a paired collection containing dataset pairs, for the data generated by paired-end or mate-pair sequencing. Galaxy collection can be used as single dataset and execution of a tool on a collection will run for the number of times of the datasets in a collection [85].

### 1.10.3 Visualisation plugins

Galaxy has the functionality to visualise data in various formats using two types of integration: standalone visualisations and visual analysis applications [86]. Visualisations include various graphs, charts, tree viewer and genomic annotation viewer such as Circos [87]. In a visual analysis, visualisation and analysis tools are blended to enable seamless and often integrative use of both to understand data, one example of such application is Trackster [88].

Galaxy integrated visualisations are particular useful when using a workflow with multiple tools as it is often helpful to be able to check the data produced by each tool to ensure its validity [86]. Galaxy's visualisation framework is flexible enough to accommodate nearly any web-based visual application; the integration of Aequatus.js has demonstrated this (see section 4.2.6).

### 1.10.4 Galaxy ToolShed

Galaxy ToolShed (<https://toolshed.g2.bx.psu.edu/>) is a repository which acts as an “appstore”. It allows Galaxy administrators to install Galaxy utilities into their instances. These utilities include tools and recipes for installing and compiling tool dependencies, data, custom datatypes and Galaxy workflows. It is freely available to all Galaxy instances as well as for tool developers and Galaxy admins.

## 1.11 Summary

Here, I have described the rise of genome sequencing technologies and bioinformatics resources. However, many of current bioinformatics tools are designed to access data from heavily curated public repositories or run on command-line to perform analysis relying on various dependencies.

Several tools can be adapted to improve bioinformatics analysis of newly sequenced genomic data. Decreasing or avoiding the necessity of making the data available through public resources to be able to perform downstream analysis or visualisation as well as the need of computer experts to perform bioinformatics analysis can be achieved through the development of versatile bioinformatics applications, these strategies are discussed in Chapters 2, 3, and 4.

**Chapter 2** describes my work on TGAC Browser and wigExplorer developed to provide visualisation for newly sequenced and annotated genomes.

**Chapter 3** describes the implementation of the EnsemblCompara GeneTrees pipeline into Galaxy as a GeneSeqToFamily workflow to gain new insights into genomic data by phylogenetic analysis.

**Chapter 4** presents the Aequatus, a homology browser, developed to provide in-depth visuals of gene families along with internal gene structural changes. It also describes the development of aequatus.js and its integration into Galaxy to visualise gene families discovered using GeneSeqToFamily. This chapter also includes ViCTreeView, which is developed to visualise viral phylogeny along with pairwise distance information.

**Chapter 5** consists of discussion on use cases and their impacts as well as limitations, potential future developments and conclusion.

# Enhancing visualisation approaches for genomic annotation

---

Most of the current genome browsers rely on a single gold standard reference data source and are only able to explore genomic data. There was a need for the ability to visualise non-model, incomplete and fragmented genomic data from multiple sources as well as perform analysis such as sequence similarity searches while exploring data. In this section, I am discussing TGAC Browser [89], an open-source genome browser developed at the Earlham Institute (see section 2.1). Also, visual elements from previous visualisation tools cannot be extracted to be used independently. Thus, BioJS [90] was developed as an open-source project to create a library of interconnected JavaScript components to visualise biological data. Here, I will discuss the development of wigExplorer, a BioJS visualisation component (see section 2.2), based on the following publications, see Appendix II for the articles in their entirety.

1. A. S. Thanki, X Bian, R. P. Davey “TGAC Browser: An open-source genome browser for non-model organisms,” bioRxiv 2019

I designed and developed TGAC Browser. I wrote the pre-print and amended it according to co-authors comments.

2. A. S. Thanki, R. C Jimenez, G. G. Kaithakottil, M. Corpas, R. P. Davey “wigExplorer, a BioJS component to visualise wig data,” F1000Research 2014

I designed and developed the wigExplorer component. I wrote the paper and improved it according to co-authors comments.



## 2.1 TGAC Browser: an open-source genome browser for non-model organisms

Many of the current web-based genome browsers are configured to visualise data from a gold-standard and manually curated public repository and support specific data format. There was a need of a genome browser to visualise newly sequenced genomic data from various popular genomic data formats as well as early draft incomplete, fragmented genomes of non-model organisms. Therefore, I have developed TGAC Browser as an open-source web-based genome browser, which can visualise genomic annotation directly from the Ensembl core database [28] hosted locally as well as other Next-Generation Sequencing (NGS) datatypes such as Variant Call Format (VCF), General Feature Format (GFF) and Browser Extensible Data (BED). These NGS data can be held on the server or uploaded directly on the TGAC Browser client.

### 2.1.1 Software design

TGAC Browser is developed with a typical server-client architecture (detail in section 1.7.4) to distribute the load between host and client, which also makes TGAC Browser accessible to a wider audience through the internet. Figure 2.1 is showing a simple representation of server-client architecture implemented in TGAC Browser.

#### Server side

The server side of TGAC Browser is written in Java programming language. It retrieves the data from the Ensembl database using the Java Database Connectivity (JDBC)<sup>1</sup> and Java Data Access Object (DAO)<sup>2</sup>.

TGAC Browser uses the Ensembl database system, because it is a standard format containing various genomic annotation, and it is widely accepted in both companies as well as academic sites. It also provides a framework to load any standard NGS formatted data into Ensembl databases. TGAC Browser supports the Ensembl core

---

<sup>1</sup>JDBC is a Java API to connect and execute the query with the database.

<sup>2</sup>DAO is an object or interface used in a programming language, it provides an abstract interface to access data from a database without exposing details of the database.

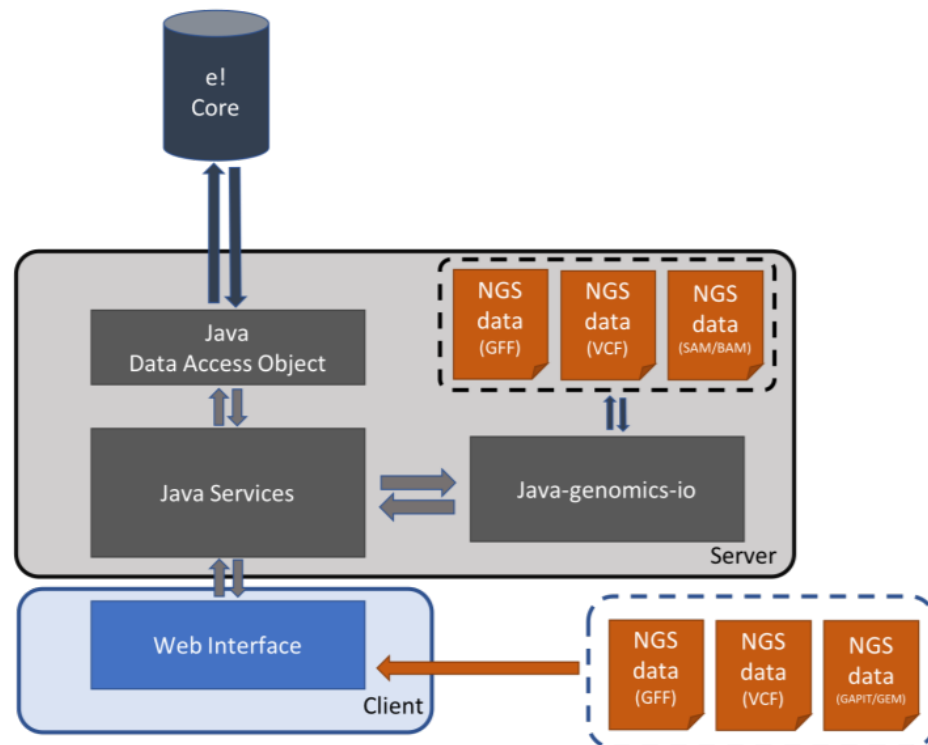


Figure 2.1: Showing the client-server architecture implemented in TGAC Browser. NGS = Next Generation Sequencing, GFF: Generic Feature Format, VCF: Variant Calling Format, SAM: Sequence Alignment/Map format

Reproduced from Thanki et al. [89]

database from out of the box. It retrieves references from `seq_region` table along with assembly information from `coord_system` and `assembly` tables. It retrieves meta information for the genomic annotations from `analysis` and `analysis_description` tables and relevant genomic features from respective tables within the database such as `gene`, `transcript`, `exon_transcript`, `exon`, and `translation` tables.

In addition, TGAC Browser is also able to retrieve data from NGS files such as, Sequence Alignment/Map format (SAM), wig, GFF, and VCF using Java-Genomics-IO library [91], a Java library developed by Timothy Palpant, a software engineer at Dropbox, to parse NGS data.

TGAC Browser retrieves and converts these data into JSON fragments and sends them to the TGAC Browser web client using AJAX technology.

## Client side

TGAC Browser's client-side is designed to utilise web technologies to visualise genomic information in a web browser. It uses JavaScript, jQuery, and d3.js (see section 1.8.1) to visualise genomic annotations received from the server.

### 2.1.2 Data visualisations

Depending on the type and volume of genomic data, TGAC Browser automatically chooses a distinctive style of visualisation for different genomic annotations (see Figure 2.2). For small datasets of less than 1000 elements, each annotation is presented independently as classic individual tracks using glyphs, while a large dataset presented either as a heat map for more than 5000 elements (see Figure 2.2 A) or as a histogram for less than 5000 elements (see Figure 2.2 B) depicting summary quantitative information. TGAC Browser uses wiggle plots (see Figure 2.2 C) for expression data using wigExplorer [92] (see section 2.2) from BioJS [90], and Manhattan style (see Figure 2.2 D) visuals for SNPs. TGAC Browser can also visualise paired-end sequencing data from SAM and BAM files (see Figure 2.2 E), in which first in the pair is coloured in blue and second in the pair is coloured in brown, reads shown with yellow colours are orphans.

This different forms of visualising data by the type and volume help the user to glance at a larger region and then focus on a particular segment for a detailed view. It is also memory efficient, using bar charts and heat maps to condense large amounts of information.

### 2.1.3 Interface features

Following the user experience model from existing genome browsers, TGAC Browser renders genomic coordinates left to right and genomic tracks top to bottom (see Figure 2.3) to provide a familiar user interface. TGAC Browser visualises assembly level genomic information on top using chromosomal karyotype (see Figure 2.3 F) if the information is available as well as horizontal selectable region (see Figure 2.3 G). A user can move selector on the selectable region, and the respective selected region will

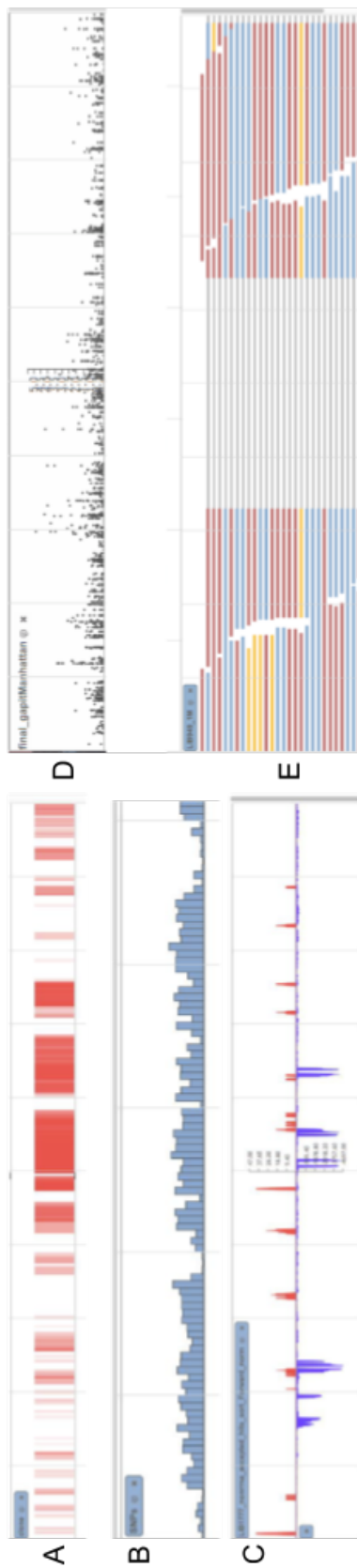


Figure 2.2: TGAC Browser visualises genomic data based on the type and amount of data: A: Heat map presentation of large data, B: Graphical presentation of large data e.g. SNPs, alignments density, C: Wiggle plot for expression data using wigExplorer, D: Manhattan style visuals for SNPs from GAPIT data, E: Visualising reads directly from SAM/BAM file.

Reproduced from Thanki et al. [89]

be shown below (Figure 2.3 H), this can also be visualised as nucleotide sequences and forward three-frame translation if zoomed enough.

TGAC Browser is equipped with various browsing functionalities, such as navigation controls in the top control bar (see Figure 2.3 J) for panning and zooming. Besides, TGAC Browser has Google Maps style navigation controls such as panning by dragging the mouse and zooming with a scroll or double click as well as panning with arrow keys on the keyboard.

TGAC Browser can visualise multiple genomic annotations together, they can be ordered by dragging them with the label of the track and can be toggled from Tracks/Settings (see Figure 2.3 C).

TGAC Browser is equipped with flexible keyword-based search functionality (see Figure 2.3 A). It searches against chromosome names, assembly information as well as all the relevant genomic feature information such as gene symbols, Ensembl stable IDs (unique identifiers in the Ensembl project for each genomic annotation), common names in the database. It visualises results along with Chromosomal view if available or in tabular form with a link to respective browser view.

### Multi-functional popup

TGAC Browser provides a contextual menu system via pop-ups (see Figure 2.3 K) containing additional information for genomic annotation (depending on the type of annotation), such as analysis type, position, and description. Pop-ups also contain options to zoom on the annotation, fetch sequence in FASTA format, perform *BLAST* analysis for a selected sequence, highlighting annotation and provides a link to Ensembl for more information, where the feature is in an Ensembl Core database.

#### 2.1.4 Data upload

TGAC Browser allows users to upload their data (e.g. GFF, GAPIT<sup>3</sup> and GEM<sup>4</sup>) and visualise. The data remains on the client side and provides a user with the possibility

---

<sup>3</sup>Genome Association and Prediction Integrated Tool (GAPIT)

<sup>4</sup>GEM is a file containing information about genes, SNPs and expression data.

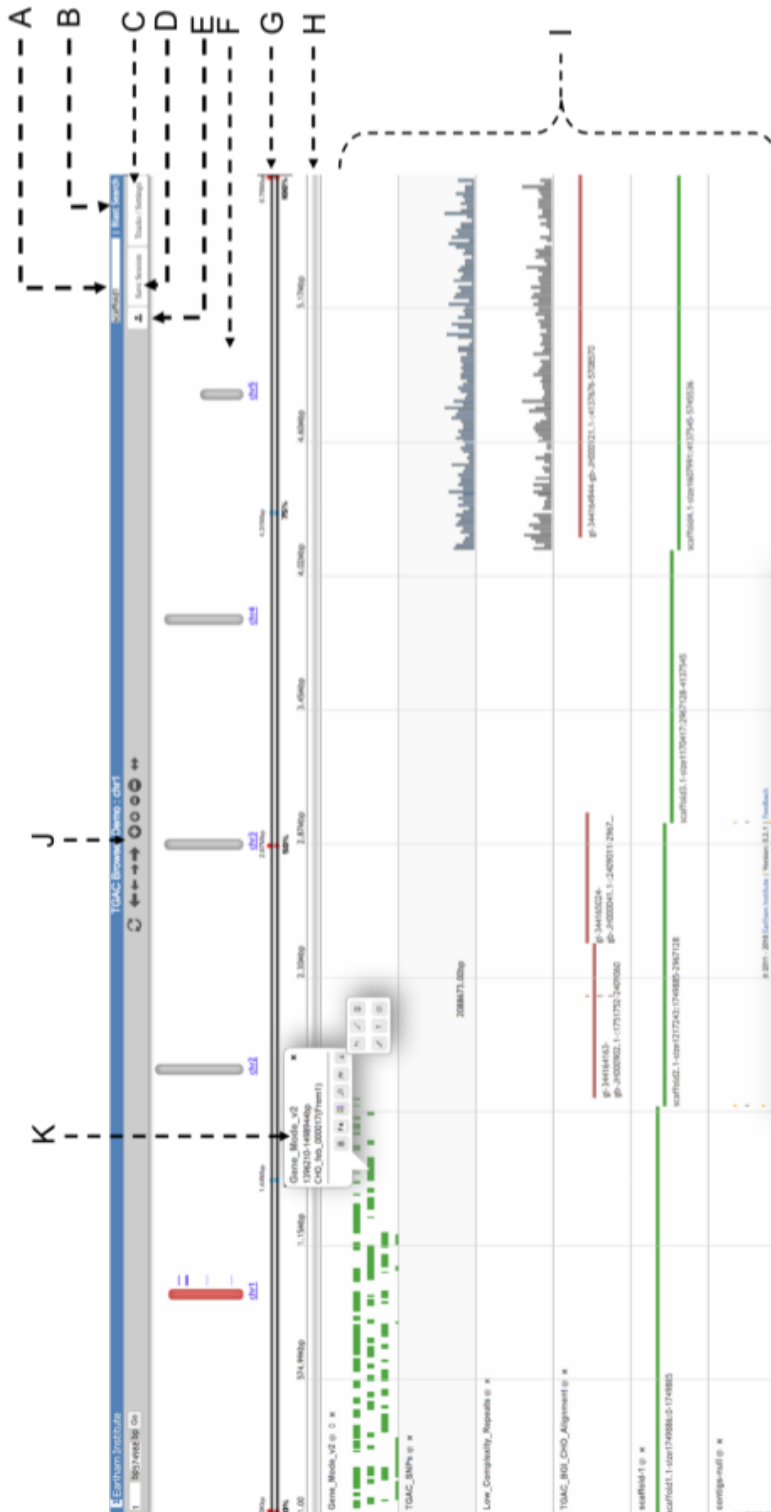


Figure 2.3: The main view of TGAC Browser: A: Search box, B: Link to *BLAST* Search, C: An option to toggle tracks, D: Save session, E: Upload tracks, F: Chromosomal reference (selected chromosome is coloured in red), G: A zoomed view of reference, H: A horizontal view of the reference, I: Genomic tracks, J: Control bar, K: Popup.

Reproduced from Thanki et al. [89]

to visualise confidential data without making it available from the server, potentially breaching data privacy or licensing.

### 2.1.5 Sequence similarity search

TGAC Browser has integrated *BLAST* analysis functionality, making it stand out from many existing genome browsers. In TGAC Browser, *BLAST* can be configured to run on the server, an HPC connected to the server or NCBI. A user can utilise *BLAST* functionality in two ways:

Firstly, a user can perform *BLAST* on a sequence of interest and results are visualised in tabular view with links to a specific result (see Figure 2.4). This feature gives TGAC Browser facility to search with a sequence of interest rather than just a keyword-based search. A user can perform multiple *BLAST* analyses, and all the results are shown as a selectable list, where the user can toggle between results. In here, the user can also choose the type of *BLAST* as well as configure other parameters for the analysis, such as scoring parameters and word size.

The screenshot displays the TGAC Browser interface for a BLAST search. The search parameters are as follows:

- Sequence: >scaffold1621.1-size467779: 231201bp - 236577bp  
tggataaatctgc
- Blast DB: TGAC\_CHO\_v1
- Type: blastn
- Advanced Parameters:
  - Repeats:  Include Repeats
  - Scoring Parameter: Match/Mismatch: 1/-2
  - Gap Costs: Existence: 5 Extension: 2
  - Word Size: 16
  - Short Queries:  Automatically adjust parameters for short input sequences.

The results table is as follows:

Query id	Subject id	% Identity	alignment length	mismatches	gap openings	q.start	q.end	s.start	s.end	e-value	bit score	Subject db	Download Sequence
scaffold1621.1-size467779:	scaffold_v1_2_076268 size9662	100.00	28	0	0	1	28	9554	9581	8e-07	54.5	TGAC_CHO_v1	<a href="#">Download Sequence</a>
scaffold1621.1-size467779:	scaffold_v1_2_316988.1 size920 unplaced	92.00	25	2	0	1	25	457	433	0.14	37.2	TGAC_CHO_v1	<a href="#">Download Sequence</a>
scaffold1621.1-size467779:	scaffold_v1_2_41562.1 size10905 unplaced	92.00	25	2	0	3	27	9050	9026	0.14	37.2	TGAC_CHO_v1	<a href="#">Download Sequence</a>
scaffold1621.1-size467779:	scaffold_v1_2_165868 size6991	100.00	19	0	0	3	21	63	45	0.14	37.2	TGAC_CHO_v1	<a href="#">Download Sequence</a>

The BLAST History panel on the right shows three entries:

- BLAST job 2Ftw7g6T (Active)
- BLAST job Ges6kbJO (No hits found)
- BLAST job 06Nkwb9R

Figure 2.4: BLAST search showing results with links out to associated TGAC Browser instance. On top right showing list of *BLAST* run, allowing previous results to be shown and removed.

Reproduced from Thanki et al. [89]

Secondly, a user can perform *BLAST* from a pop-up menu on any of the selected genomic feature and results are presented as a genomic track alongside others (see

Figure 2.5). *BLAST* Results are coloured using the standard *BLAST* schema based on the bit score, and insertions and deletions are presented as a black box on relevant *BLAST* hit. This type of *BLAST* feature can be used to find out other matching regions for the selected annotation, to find the copy number of particular genomic feature in the case for ploidy<sup>5</sup>.

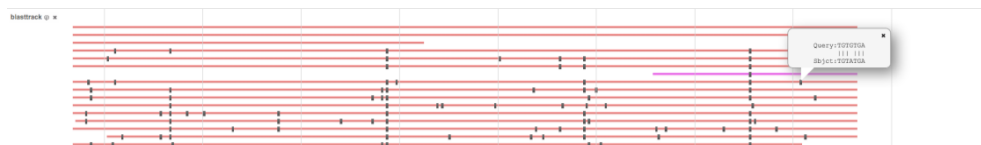


Figure 2.5: *BLAST* search showing results as genomic track. On top right pop-up showing insertions and deletions at the position, *BLAST* hits are coloured based on score.

Reproduced from Thanki et al. [89]

### 2.1.6 Summary

To summarise, TGAC Browser is an open-source web-based genome browser, designed to visualise genomic annotations of model and non-model organisms from a variety of data sources. Ability to visualise genomic data straight from the Ensembl database as well as various NGS data format makes it widely adaptable. Integration of *BLAST* analysis makes it more than just a genomic data exploration tool. TGAC Browser was ahead of its time when it was developed, and it is super-seeded by modern genome browsers but, still many of them can not work with non-model organisms and fragmented datasets.

<sup>5</sup>Ploidy is the number of complete sets of chromosomes in a cell



## 2.2 wigExplorer, a BioJS component to visualise wig data

Wiggle (wig) data is a text-based genomic data format, designed to describe the continuous density data, such as probability scores and gene expression values. The size of the wig file depends on the density of the data (e.g. single base coverage or compressed data using span) rather than the length of the genomic region. Wig formatted file is widely used by many genome browsers such as Gbrowse, JBrowse and TGAC Browser (see section 2.1). Most of the existing genome browsers use a hard-coded system to visualise wiggle data specifically developed for the tool, which can not be used independently or in conjunction with other tools.

I have developed wigExplorer [92], as a BioJS component, to visualise wig-formatted data. BioJS provides an open-source platform for 235 interconnected and reusable biological visualisation components at the time of writing. Therefore, wigExplorer can be used independently as well as in conjunction with other components.

### 2.2.1 Implementation

wigExplorer visualises data from both `variableStep` as well as `fixedStep` formatted wig files. `variableStep` format (see Listing 2.1) is more commonly used wig format, it contains data with irregular intervals between new data points. `fixedStep` format (see Listing 2.2) is the more compact wiggle format, it is used for data with regular intervals between new data values.

```

1 variableStep chrom=chr1
2 10 34
3 20 41
4 60 57
5 70 66
6 80 67
7 90 66
8 100 73
9 110 75
10 120 73
11 130 72

```

Listing 2.1: Example of variableStep wig file

```

1 fixedStep chrom=chr3 start=400601 step=100 span=5
2 34
3 41
4 46
5 49
6 52
7 57
8 66
9 67
10 66

```

Listing 2.2: Example of fixedStep wig file

wigExplorer follows the standard BioJS component specification. It requires a minimal configuration (see in Listing 2.3), where the `target` (line 2) is set to the HTML component (div ID) of the placeholder for visualisation to render, the `dataSet` (line 3) is set to the path of the input wig file, and the colour for the wiggle plot can be set using `selectionBackgroundColor` (line 4). wigExplorer also allows interaction via predefined events using an API (see in Listing 2.4) to update the visualisation using a new start and end position.

```

1 var instance = new Biojs.wigExplorer({
2   target: YourOwnDivId,
3   dataSet: <path-to-wig-file >,
4   selectionBackgroundColor: <background-colour >
5 });

```

Listing 2.3: Snapshot of code to configure wigExplorer

```

1 instance._updateDraw(start, end)

```

Listing 2.4: Snapshot of code to update wigExplorer

### 2.2.2 Interface features

To generate an interactive and clear visualisation from wiggle data, wigExplorer implements the D3.js library (see section 1.8.1). Users can pan and zoom the visible region of the selected reference using the controls on the top-left. For a wig file containing

multiple regions, the reference region can be changed by using the dropdown on the top-right (See Figure 2.7).

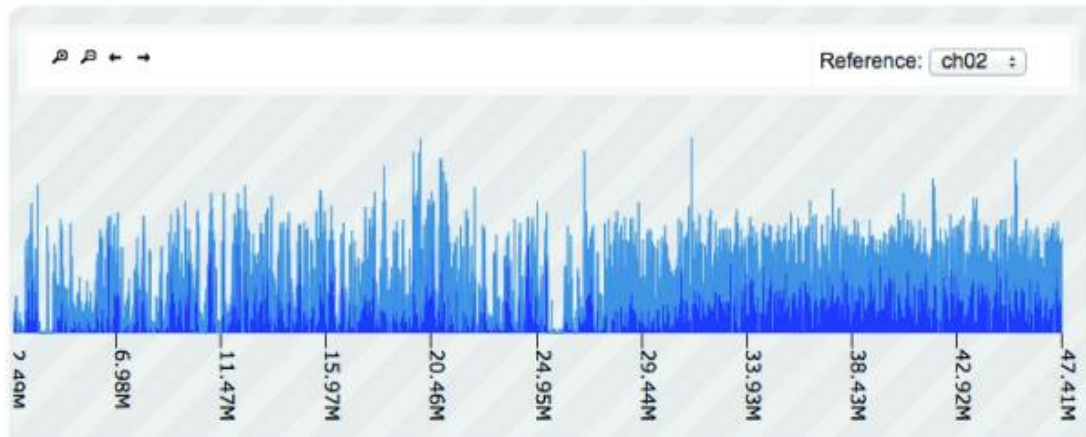


Figure 2.6: wigExplorer view of tomato variety Heinz chromosome 2. Peaks show single nucleotide polymorphism (SNP) density of 1KB size bins. A change of SNP density can be observed around the 24M mark, with a slightly greater density of SNPs on the right, indicative of a potential interrogation segment from another related species.

Reproduced from Thanki et al. [92]

### 2.2.3 Use case

Being a BioJS component, wigExplorer can easily be integrated into and controlled by any web-based application, thereby advancing its uptake. This has been demonstrated by integration of wigExplorer into the TGAC Browser (see section 2.1) to visualise wig data, either from data on the server or from user-uploaded data.

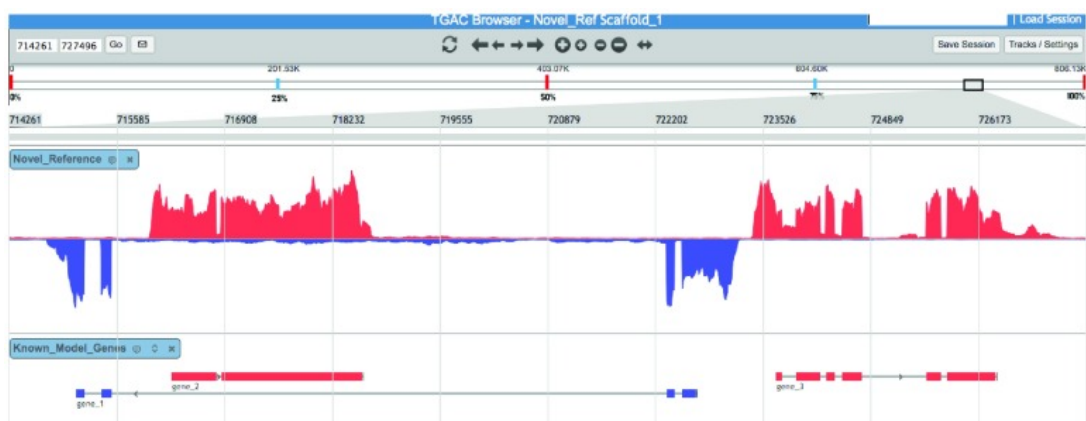


Figure 2.7: The wigExplorer track shows read coverage in *Myzus spp.* for scaffold 1. Forward and backward strands are depicted in red and blue, respectively. Evidence genes from a closely related species are displayed in the track below.

Reproduced from Thanki et al. [92]

### 2.2.4 Summary

To summarise, wigExplorer is an open-source BioJS visualisation component developed to visualise genomic expression data. It can easily be integrated into any web-based application that supports JavaScript. To date, wigExplorer is the only component to visualise wig data that complies with BioJS standards. wigExplorer was developed during the BioJS version 1 life cycle, so, unfortunately, it only works specifically with BioJS legacy code. BioJS went through a complete redesign of the library and was released in 2014 as version 2.0. wigExplorer was revised to support this new release by Sebastian Wilzbach [93]. Though, it is not fully compatible with version 2.0, and later.

# Annotation and characterisation of gene families using Galaxy

---

Most of the tools and pipelines developed for analysing comparative genomics data are command-line based and require many dependencies. In the case of annotating and characterising gene families, this is no different. To overcome this issue, I have developed GeneSeqToFamily, a Galaxy workflow based on the EnsemblCompara GeneTrees pipeline. I discuss the EnsemblCompara GeneTrees pipeline in section 3.1 and the development of GeneSeqToFamily in section 3.2 based on the following publication:

1. A. S. Thanki, N. Soranzo, W. Haerty, and R. P. Davey, “GeneSeqToFamily: a Galaxy workflow to find gene families based on the Ensembl Compara GeneTrees pipeline,” GigaScience 2018

I defined the overall problem with the current solution and developed the Galaxy platform-based software solution also wrote the paper and improved it according to co-authors comments. Nicola Soranzo advised on integrating new and existing tools into Galaxy.

## 3.1 The EnsemblCompara GeneTrees pipeline

The EnsemblCompara GeneTrees pipeline [41] was developed to predict gene trees using existing tools. It is made up of eight main steps for clustering, multiple sequence alignments, and tree generation using popular tools such as BLAST, hcluster\_sg [94, 95], T-Coffee [96], and Tree Building guided by Species Tree (TreeBeST) [97], a phylogenetic tree construction tool. The advantage of this pipeline is that it provides gene tree

topology for the discovered gene families as well as the protein alignments.

The pipeline uses TreeBeST to generate genetree using CDS. TreeBest, developed as a part of TreeFam, implements various separate phylogenetic methods and merges the results into a consensus tree to minimise duplications and deletions relative to a known species tree. In this way, TreeBeST takes advantage of the fact that DNA-based trees are usually more accurate for closely related species, and protein-based trees are better at longer evolutionary distances.

The Ensembl GeneTrees pipeline is a gold standard analysis pipeline used within Ensembl resources to find gene families among the deposited genomes. According to Google Scholar, it has been cited for 876 times. The results of the pipeline are being used by other resources for further analysis such as Genomicus uses Ensembl data to visualise homology.

### **3.2 GeneSeqToFamily: a Galaxy workflow to find gene families based on the EnsemblCompara GeneTrees pipeline**

The EnsemblCompara GeneTrees pipeline [41], useful as the pipeline is, it is command-line based and requires 44 dependencies, which makes it challenging to use by a non-computer expert and also makes it difficult to configure for a bioinformatician. Thus, I decided to use Galaxy to overcome these problems and developed GeneSeqToFamily [98], a Galaxy workflow based on the EnsemblCompara GeneTrees pipeline.

The GeneSeqToFamily allows users to run gene family analyses through a graphical user interface without any knowledge of the command-line. It also allows tool parameters, configurations, and the tools themselves to be modified to make it suitable for different genomes and need of the analysis.

For this workflow, I took advantage of already existing tools (*BLASTP*, *T-Coffee*, *EMBOSS*) to reduce redundancy but also developed new tools (*BLAST parser* [99] and *hcluster\_sg parser* [100]). Some tools required to write wrappers to be used in Galaxy (*hcluster\_sg* and *TreeBeST*). A complete list of these tools is available in Table

## 3.1.

Tool name	Tool ID	Developed at Earlham Institute	
		Tool	Wrapper
Get sequences by Ensembl ID	get_sequences	✓	✓
Get features by Ensembl ID	get_feature_info	✓	✓
Select longest coding sequence per gene	ensembl_longest_cds_per_gene	✓	✓
ETE species tree generator	ete_species_tree_generator	✓	✓
GeneSeqToFamily preparation	gstf_preparation	✓	✓
BLAST parser	blast_parser	✓	✓
hcluster_sg	hcluster_sg		✓
hcluster_sg parser	hcluster_sg_parser	✓	✓
T-Coffee	t_coffee		✓
TreeBeST best	treebest_best		✓
Gene Alignment and Family Aggregator	gafa	✓	✓

Table 3.1: Galaxy tools and wrappers developed for the workflow.

### 3.2.1 GeneSeqToFamily input data

The workflow requires three inputs: (1) coding sequences (CDS), (2) gene feature information and (3) a species tree. User can provide these inputs or can be prepared using tailored tools developed as a part of the workflow package, *the Ensembl suite* [101] and *GeneSeqToFamily preparation tool*, to retrieve and prepare data.

#### Retrieve data with the Ensembl suite

To avoid the tedious process of downloading data from the Ensembl and making them available, as well as to help the user with data preparation, a REST-based suite of Ensembl tools was developed for Galaxy. This enables retrieving data (sequence, genomic feature and gene tree) directly from the Ensembl server with the help of Ensembl REST API [56]. These tools take a list of feature IDs as the input and retrieve respective information from Ensembl resources. These tools are useful for the data preparation for GeneSeqToFamily, as using annotated Ensembl data used with newly sequenced non-model organism data can help define gene families. These tools could also be used to retrieve data for various other analysis.

### GeneSeqToFamily data preparation

GeneSeqToFamily workflow requires input in specific formats. *GeneSeqToFamily Preparation* tool was developed to convert data into GeneSeqToFamily supported formats. It takes CDS sequences in FASTA format and generates FASTA with modified headers containing species information. This FASTA headers with species information is required by *TreeBeST* to generate gene tree using species tree. To visualise gene features along side the gene tree, the *GeneSeqToFamily Preparation* tool also takes gene features in GFF3 or JSON, or both, then generates a SQLite database for gene feature information. We chose SQLite over GFF3 or JSON because, the GFF3 format has a relatively inconvenient and unstructured additional information field (9th column) and SQLite provides faster and efficient search functionality than text-based GFF3 or JSON.

*GeneSeqToFamily Preparation tool* has following options:

1. Keep only the longest CDS per gene

This will select the longest CDS per gene to avoid self-matching isoforms for the gene, which can result in the unreliable gene tree.

2. Set transcript ID to FASTA header

This will set transcript IDs from the gene feature as CDS FASTA header, If CDS FASTA header is not transcript IDs, Aequatus visualisation will not be able to link gene tree with gene features

3. Filter out CDS from chosen references.

This option can be used to discard CDS with special codons such as mitochondrial genes.

### 3.2.2 GeneSeqToFamily workflow

The GeneSeqToFamily workflow comprises seven main steps (see Figure 3.1). The complete workflow, containing several intermediate steps, can be seen in Figure 3.2.



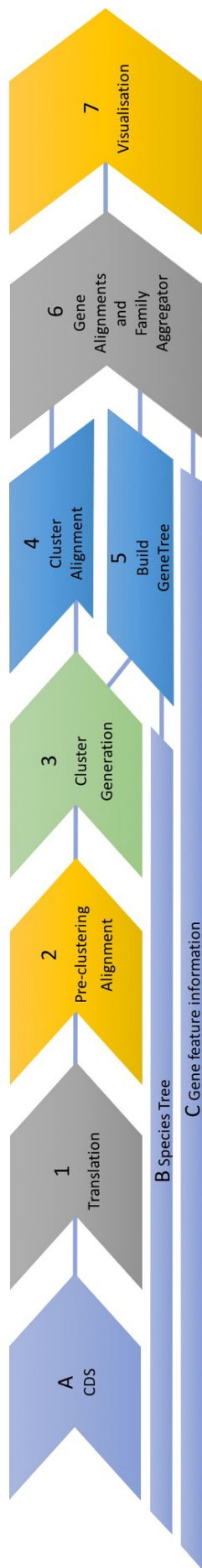


Figure 3.1: Overview of the GeneSeqToFamily workflow. *A*, *B* and *C* are showing inputs and *1-7* are showing workflow steps.

Reproduced from Thanki et al. [98]

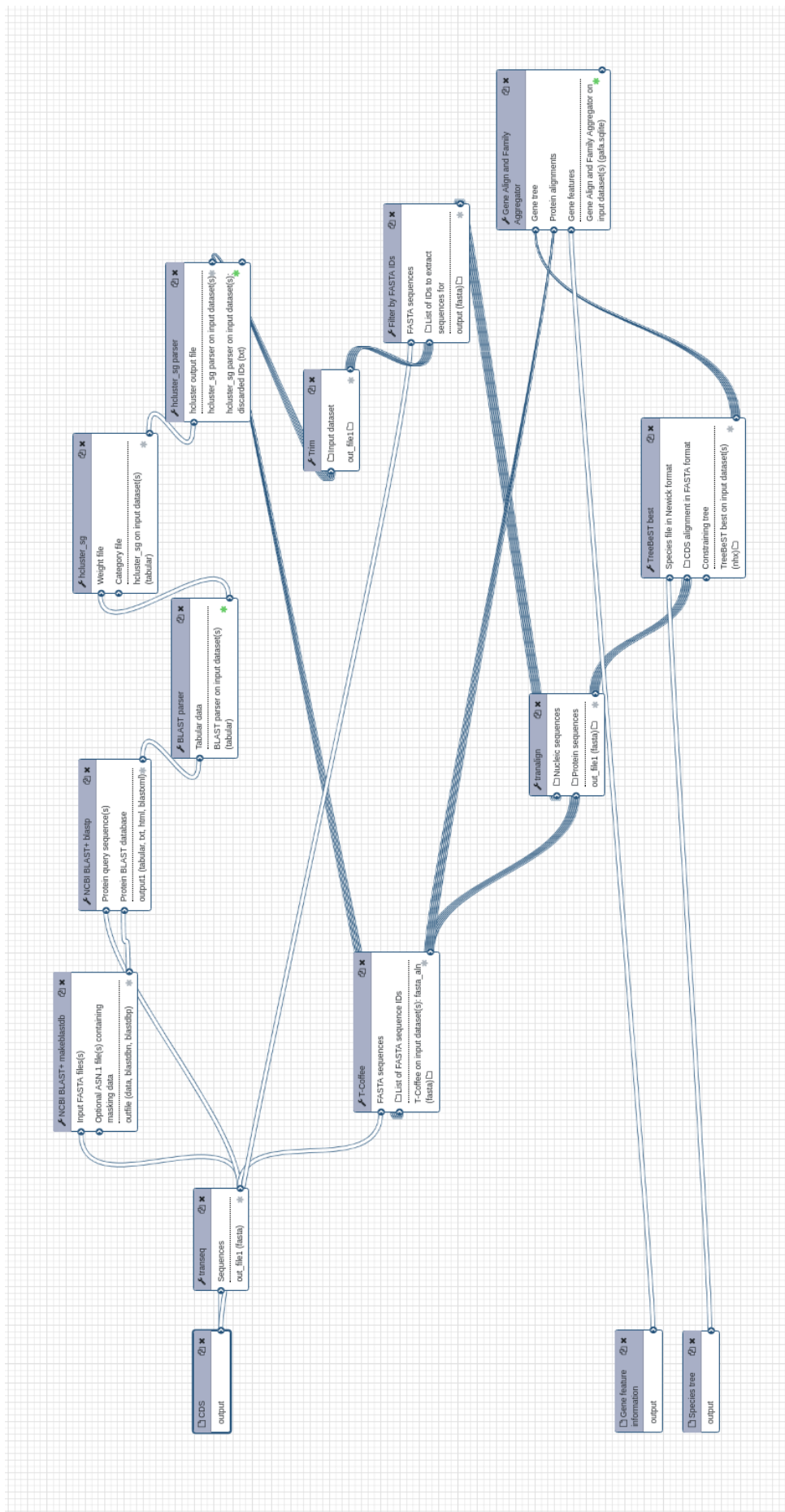


Figure 3.2: The GeneSeqToFamily workflow in Galaxy workflow editor showing all intermediate steps.

**Step 1: CDS to protein translation with *Transeq***

GeneSeqToFamily workflow takes CDS file (see Figure 3.1 A) and translates into protein sequences rather than taking protein sequences as an input, because *TreeBeST* (see step 5) requires nucleotide sequences to generate a gene tree.

To translate CDS input to protein sequences, GeneSeqToFamily uses *Transeq*, part of EMBOSS [51]. These protein sequences will be used to run *BLASTP* to find protein clusters (see step 2).

**Step 2: Pre-clustering alignment with *BLAST***

Pre-clustering alignment is generated by using the *BLASTP* of BLAST+ package [26, 102] tool within Galaxy. Here, *BLASTP* aligns protein sequences against the database created from the same sequences to find matching sequences from the set of input data. Default parameters for *BLASTP* are shown in Listing 3.1 and the output format is set to 12 or 25-column because workflow requires pairwise statistical comparison and does not require sequence alignment.

```

1  -seg no (Filter out low complexity regions (with SEG))
2  -max_hsps_per_subject 1 (Maximum number of alignments to
   keep for any single query-subject pair)
3  -evalue 1e-10 (Expectation value cutoff)

```

Listing 3.1: Default *BLASTP* parameters in the workflow.

*BLAST* output is then fed into the *BLAST Parser* [99] to convert it into the 3-column tabular input format required by *hcluster\_sg* [94, 95]. This format comprises the *BLAST* query ID, the hit result ID, and the edge weight. The edge weight is calculated as  $Weight = \min(100, \text{round}(-\log_{10}(\text{evalue})/2))$ . *BLAST Parser* removes self-matching *BLAST* results and also provides an option to filter out non-reciprocal *BLAST* hits for stringent analysis.

**Step 3: Cluster generation with *hcluster\_sg***

Clusters are generated from *BLAST parser* output using *hcluster\_sg*, a hierarchical clustering algorithm, with parameters shown in Listing 3.2. It

iteratively groups the two closest nodes (sequence Ids in this case) from the *BLAST Parser* output, then the distance between other nodes and joined node would be the mean distance for joined nodes. At the end of the process, it generates a single list of clusters from the *BLAST Parser* output.

```

1  -m 750 (Maximum cluster size)
2  -w 0 (Minimum edge weight)
3  -s 0.34 (Minimum edge density between a join)
4  -O no (Use once-fail-inactive-forever mode)

```

Listing 3.2: Default *hcluster\_sg* parameters in the workflow.

*hcluster\_sg parser* tool [100] is then used to separate each cluster because downstream analysis of the workflow is performed separately on each cluster. *hcluster\_sg parser* takes output from *hcluster\_sg* and creates an individual list of sequence IDs for each cluster. It also has options to set minimum and maximum threshold for cluster size, and sequence IDs for the clusters does not meet the minimum, and maximum threshold will be added into a separate file. Here, *hcluster\_sg parser* provide an option for minimum threshold as *TreeBeST* can not generate gene tree for a cluster with less than three sequences and the maximum threshold is set to separate large clusters for reiteration of previous steps with stringent parameters to get smaller clusters.

#### Step 4: Cluster alignment with *T-Coffee*

Multiple Sequence Alignment (MSA) for each cluster is generated using *T-Coffee* [96], a package in which multiple alignment methods (e.g. clustal, MAFFT and MUSCLE) can be combined and generate a single alignment.

Typically, *T-Coffee* takes multi-FASTA<sup>1</sup> as an input and performs MSA on them. To find the similarity among the sequences in each cluster, workflow needs to perform MSA for each cluster separately. For that, the first workflow needs to generate individual FASTA files for each cluster then perform MSA using *T-Coffee*. I modified the *T-Coffee* wrapper for Galaxy [103] to perform MSA on a subset of sequences provided by the list of FASTA IDs. This modification helps to remove an extra step for the workflow of filtering input

<sup>1</sup>A multi-FASTA file contains multiple FASTA formatted sequences.

sequences for the clusters, avoids the addition of intermediate dataset to the Galaxy.

The GeneSeqToFamily workflow uses ClustalW as default alignment method in *T-Coffee*, but it can be changed per users' preference.

### Step 5: Gene tree construction with *TreeBeST*

Gene trees are constructed for each cluster using TreeBeST [39, 97]. *TreeBeST* is designed to generate a gene tree with a known species tree, but it can also be used to display and manipulate trees.

*TreeBeST* requires nucleotide MSA as input to generate gene trees, but till now the workflow was using protein sequences and generated MSA for protein sequences (in step 4). *Tranalign*, part of EMBOSS [51], is used to generate MSA of CDS (one of the input, see Figure 3.1 A) by mapping protein alignments on nucleotide sequences. Resulting MSA of CDS is then used in *TreeBeST* to generate gene tree.

Here, the 'best' option from *TreeBeST* is used, which builds five different gene trees using different phylogenetic algorithms (maximum likelihood tree from nucleotide and protein alignment, a neighbour-joining tree using p-distance, dN distance and dS distance), then merges them into a single consensus tree using a species tree as a reference. Resulted gene tree also contains useful annotations for phylogenetic information like speciation (S), duplication (D) and duplication score (DCS).

### Step 6: Gene alignment and family aggregation (GAFA)

I developed *Gene alignment and family aggregation (GAFA)* as a Galaxy tool to merge results from *T-Coffee* (step 4), *TreeBeST* (step 5) and gene information (from input C) into a single SQLite database. It requires gene trees in Newick format, protein MSA in multi-FASTA format, and gene feature information in SQLite format (generated using *GeneSeqToFamily preparation* tool) and stores them in an SQLite database (see Figure 3.3).

The database contains four interconnected tables (`gene`, `transcript`, `gene_fa`

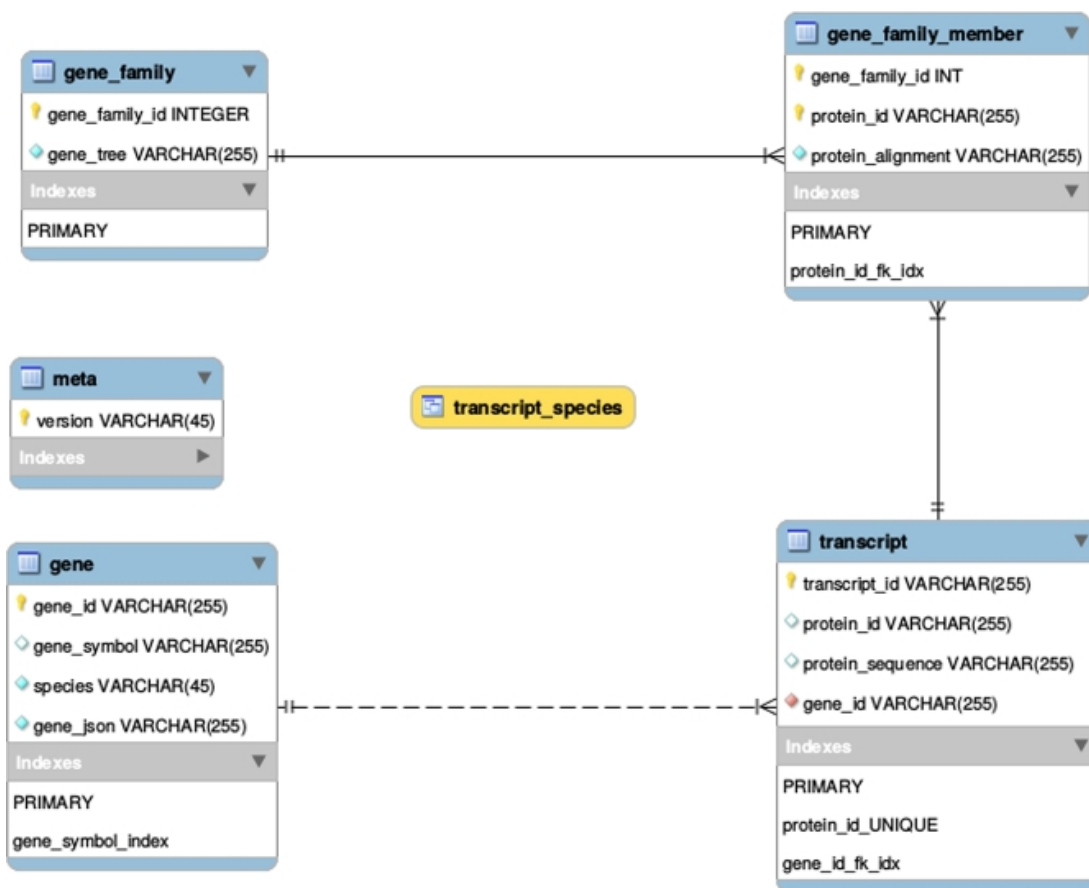


Figure 3.3: Gene alignment and family aggregation (GFAFA) SQLite database schema. Reproduced from Thanki et al. [98]

mily, and `gene_family_member`) with genomic information and a `meta` table with version information. It also contains a database view<sup>2</sup> called `transcript_species` for an efficient link between `transcript` and species information from the `gene` table.

MSA generated by *T-Coffee* (step 4) is in multi-FASTA format, which can be large in size and occupy more space in the database. To reduce the size of the database, *GFAFA* tool converts each FASTA formatted MSA into a simple CIGAR string before storing in the database. Because CIGAR is a string that describes the alignment efficiently. Generally, CIGAR is used for the pairwise alignment, but here we are using it for MSA. A small example of CIGAR for MSA is shown in Listing 3.3.

<sup>2</sup>A view contains rows and columns, much like a real table. A view does not store data, but is a searchable object in a database that is defined by a query, also known as a virtual table

```

1 seq1:  NLYIQWLKDGGPSSGRPPPS
2 seq2:  NLYIQWLKDGGPSSGRPPPS
3 seq3:  GDAYAQWLADGGPSSGRPPPSG
4
5 aln1:  -NLYIQWLKDGGPSSGRPPPS-
6 aln2:  -NLYIQWLKDGGPSSGRPPPS-
7 aln3:  GDAYAQWLADGGPSSGRPPPSG
8
9 CIGAR1: D19MDM
10 CIGAR2: D19MDM
11 CIGAR3: 22M

```

Listing 3.3: Example of MSA and CIGAR. MSA generated using online *T-Coffee* with default parameters.

### Step 7: Visualisation with Aequatus

SQLite database can be queried using standard SQLite compatible tools. However, we use Aequatus.js plugin (see section 4.2.6) to explore gene family and homologous genes relationships. The Aequatus plugin takes SQLite database (from step 6) as input and visualises gene families (see Figure 3.4) similar to the Aequatus project. The Aequatus plugin in Galaxy is now available by default in the Galaxy platform (since version 19.01).

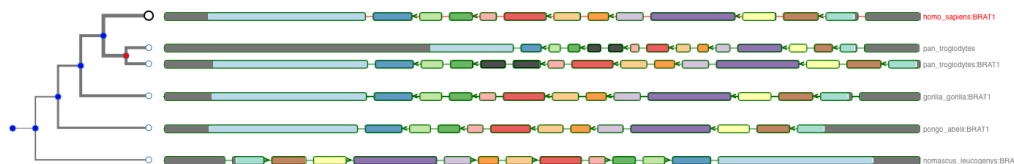


Figure 3.4: Homologous genes of BRAT1 for the *Homo sapiens* (human), *Pan troglodytes* (chimpanzee), *Nomascus leucogenys* (gibbon), *Gorilla gorilla* (gorilla), and *Pongo abelii* (orangutan), where shared exons are colour coded in same colours.

### 3.2.3 Supplementary workflows

Along with GeneSeqToFamily workflow, two supplementary workflows are developed: one to retrieve lists of orphan genes and another to find homologous genes for the orphan genes using SwissProt, a curated protein sequence database.

The first workflow is developed to find orphan/unique genes from the GeneSeqToFamily workflow. As part of the homology search, some genes might appear to have no homologous relationship to any other genes among the species used. This could be

the consequence of the choice of parameters of the tools, incomplete annotation or the input dataset may contain data from the distant evolutionary species with little, or no phylogenetical relationships.

The workflow starts with finding sequence IDs present in GeneSeqToFamily input data (input A) but missing from the results of *BLAST parser* (see section 3.2.2, step 2) then merging this list with the sequence IDs discarded by *hcluster\_sg parser* (see section 3.2.2, step 3). After that, it retrieves the respective sequence for each ID from GeneSeqToFamily input CDS using *Filter by FASTA IDs* [104], a Galaxy tool to filter FASTA by the headers and/or the sequences. The resulting list of sequence belong to the orphan/unique genes.

The second workflow is developed to facilitate the annotations and interpretation of orphan genes, CDS for these genes are fed into the SwissProt workflow to find homologous genes or protein domains in other species not present in the input dataset.

The second workflow starts with the translation of CDS (typically retrieved from the previous workflow) into protein sequences using *Transeq*. Then it performs the *BLASTP* for the protein sequences against the Swiss-Prot database<sup>3</sup> (downloaded from NCBI<sup>4</sup>) after that it extracts UniProt IDs<sup>5</sup> from *BLASTP* results and retrieves Ensembl IDs (representing genes and transcripts) for each UniProt ID using *UniProt ID mapping and retrieval tool* [105]. For these Ensembl IDs, it retrieves genomic information for gene IDs as well as CDS for transcript IDs using the *Ensembl suite*. These data can be used to re-run GeneSeqToFamily workflow to find and visualise gene families for new data.

These two workflows are designed to help users identifying and interpreting unique genes present in the input dataset.

### 3.2.4 Example

Here, I present an example with step by step process of finding homologous genes starting from data retrieval followed by data preparation and then running the workflow.

<sup>3</sup>Swiss-Prot is a curated protein sequence database developed to provide a high level of annotation.

<sup>4</sup>National Center for Biotechnology Information (NCBI)

<sup>5</sup>UniProt is the Universal Protein resource, a central repository of protein data combining the Swiss-Prot, TrEMBL and PIR-PSD databases.



1. Upload or paste Ensembl Gene IDs for the genes of interest (see Listing 3.4) as well as species names (see Listing 3.5) into Galaxy.
2. Retrieve CDS for the uploaded Gene Ids by using *Get sequences by Ensembl ID* tool, and setting the ‘type’ as CDS.
3. Retrieve gene feature information using *Get features by Ensembl ID* tool, and setting ‘expand’ to yes.
4. A species tree can be generated from a list of species by using *ETE species tree generator* (Galaxy Version 3.0.0b35), setting ‘Use in TreeBeST’ as yes.
5. Prepare data using *GeneSeqToFamily preparation* tool, selecting gene features in JSON format (from step 3) and CDS (from step 2). Set ‘Keep only the longest CDS per gene’ to yes to keep only one CDS per gene and change FASTA header to ‘TranscriptId\_species’ to yes. It will create formatted FASTA file (with the header as `transcriptid.speciesname`), an SQLite database (containing gene features), and filtered out FASTA.
6. Run the GeneSeqToFamily workflow using the datasets generated in step 5 and the species tree (from step 4).
7. Visualise results using the Aequatus plugin (see Figure 3.4) within Galaxy described in 3.2.2, step 7.

```
1 ENSPTRG00000018865
2 ENSPTRG000000052179
3 ENSG00000106009
4 ENSGGOG00000008928
5 ENSPPYG00000017318
6 ENSNLEG00000010655
```

Listing 3.4: List of Gene IDs of BRAT1 for the *Homo sapiens* (human), *Pan troglodytes* (chimpanzee), *Nomascus leucogenys* (gibbon), *Gorilla gorilla* (gorilla), and *Pongo abelii* (orangutan)

```
1 homo sapiens
2 pan troglodytes
3 nomascus leucogenys
4 gorilla gorilla
5 pongo abelii
```

Listing 3.5: List of species

### 3.2.5 Benchmarking

The GeneSeqToFamily workflow was tested with a small dataset of 39 sequences of three gene families by comparing the results of GeneSeqToFamily with existing gene tree reported in the Ensembl. The workflow led to similar gene tree inferences as of the Ensembl, further consolidating our aim to reproduce the EnsemblCompara GeneTrees pipeline within a Galaxy workflow.

The GeneSeqToFamily workflow is developed using existing tools with pre-defined parameters. To look into the effect of parameters used by the workflow, it was tested with the reference proteomes of 754,149 sequences from 66 species established by the Quest for Orthologs (QfO) consortium [106]. QfO provides a shared reference dataset to benchmark, improve and standardise orthology predictions tools. This test was performed with different sets of parameters for *BLASTP* and *hcluster\_sg* (see Table 3.2). A brief statistical report, generated from the results (see Table 3.3), showing that parameters used in *BLASTP* and *hcluster\_sg* affects the number of gene families heavily. Strict parameters for *BLASTP* and *hcluster\_sg* (parameter set F) generates a large number of smaller families, where relaxed parameters (parameter set A) results into a small number of larger families, which may include distantly related genes.

Tool	Parameter	Parameter set					
		A	B	C	D	E	F
<b>BLASTP</b>	Expectation value cutoff	1e-03	1e-03	1e-03	1e-10	1e-10	1e-10
	Query coverage per hsp	0	0	90	0	0	90
<b>hcluster_sg</b>	Minimum edge weight	0	20	0	0	20	20
	Minimum edge density between a join	0.34	0.50	0.34	0.34	0.50	0.50

Table 3.2: Set of parameters used in *BLASTP* and *hcluster\_sg* to compare results. *BLASTP* was configured with maximum number of HSPs set to 1, and *hcluster\_sg* with single link clusters set to ‘no’ and maximum size set to 500, and parameter set D is showing the default parameters used in the Ensembl Compara pipeline.

Reproduced from Thanki et al. [98]

To compare the results of GeneSeqToFamily with other homology tools, I have performed benchmarking using the QfO, which assesses the accuracy of a tool or pipeline to predict 1-to-1 orthology by comparing results with existing resources. GeneSeqToFamily performs comparably to other tools benchmarked in QfO, even surpassing them for

Summary						
Analysis	A	B	C	D	E	F
No. of genes	754,149					
No. of families	58,272	74,252	83,900	63,289	74,309	79,879
No. of larger families (>200)	435	168	56	350	167	46
No. of smaller families (<200)	30,563	40,530	44,295	33,308	40,579	41,794
Considered families (>3 and <200)	27,274	33,556	39,548	29,628	33,562	38,039
Largest family size	615	567	556	652	561	527

Table 3.3: Results of the GeneSeqToFamily workflow run with 6 different set of parameters, the complete list of which are shown in Table 3.2.

Reproduced from Thanki et al. [98]

true positive ortholog discovery in some parameter spaces (see Figure 3.5). As the QfO service is designed for 1-to-1 orthologs, it records paralogs and 1-to-many orthologs as false positives, hence reducing our overall specificity because the GeneSeqToFamily focuses on whole gene families, regardless of the type of homology among the members of a gene family. More details about both of these tests are available in GeneSeqToFamily workflow manuscript [98].

### 3.2.6 Current issues

#### Forever running *BLAST* jobs

The first version of the workflow is developed based on the EnsemblCompara GeneTrees pipeline, using the same tools and parameters wherever possible. After using it for big datasets such as QfO proteome data, it became necessary to modify the workflow to cope with larger datasets.

Specifically, *BLAST* seemed to be the bottleneck taking the substantial amount of time compared to other steps of the workflow. To solve this issue, input data for *BLAST* was split into multiple smaller chunks using *fasplit* [107] and perform *BLAST* alignment then concatenate results for each chunk before performing *BLAST parser*. In this approach, *BLAST* database is generated using the single input file, thus results would be the same as running single *BLAST*. Also, this process runs multiple *BLAST* in parallel for smaller input query; it is much faster than running a single *BLAST*.

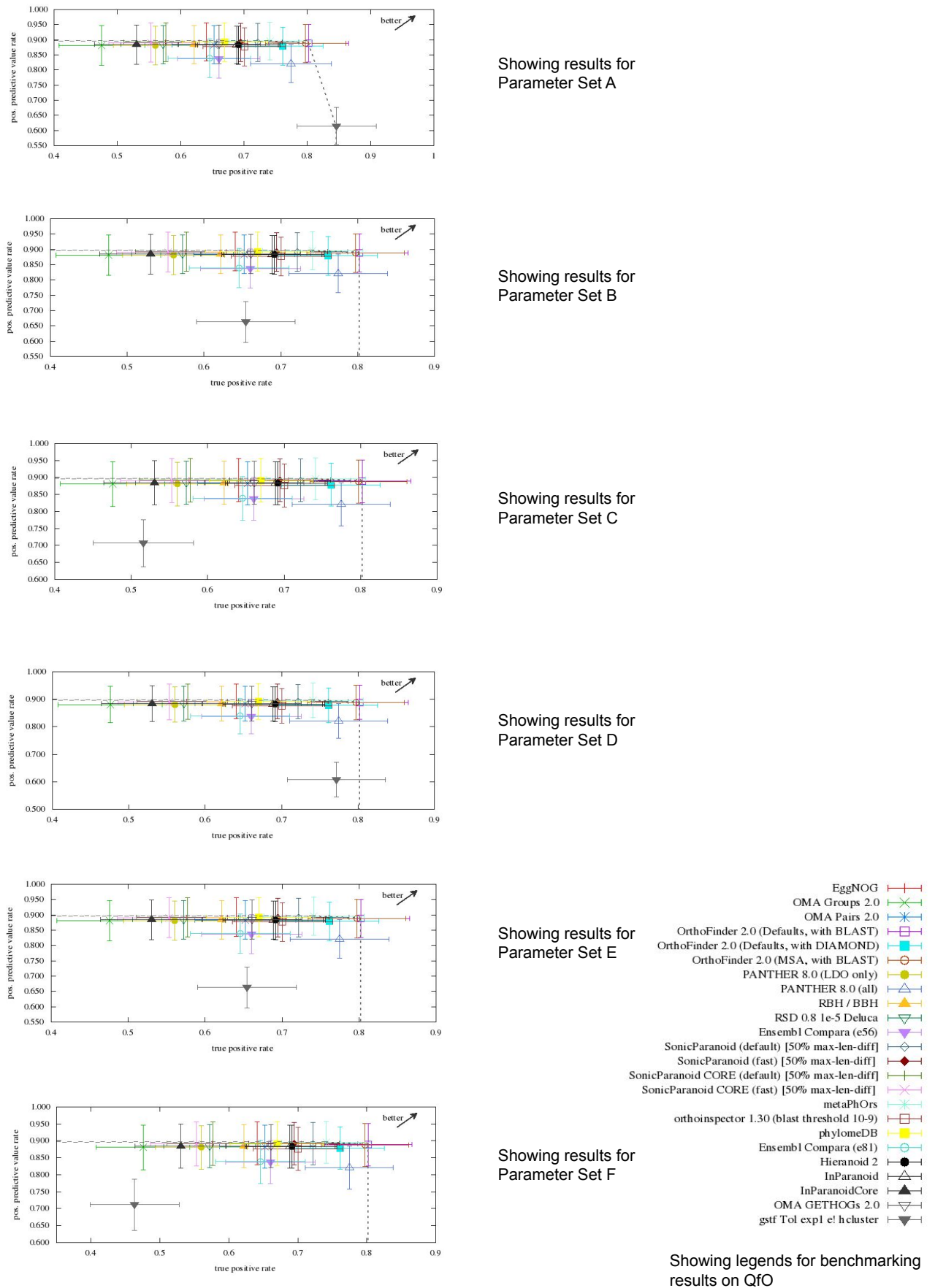


Figure 3.5: Showing results for benchmarking on Quest for Orthologs

Reproduced from Thanki et al. [98]

### Large Galaxy collections

GeneSeqToFamily workflow performs well to find homologous genes and gene families, but sometimes for large datasets, the number of gene clusters can be high (tens of thousands). This results in Galaxy tools running in parallel for a large collection. Galaxy can fail to handle larger collections, especially in the case of job failure due to a technical issue with Galaxy instance or HPC, and failed jobs need to be restarted manually.

### Memory hungry *hcluster\_sg*

*hcluster\_sg* is a hierarchical clustering algorithm, it iterates through all the nodes in input data finding the closest nodes until it generates a single list of clusters. A large input dataset can affect the performance of *hcluster\_sg*. It requires a large memory, sometimes which is not possible. Here, I am investigating into finding an alternate of *hcluster\_sg* for clustering purpose.

### 3.2.7 Summary

The GeneSeqToFamily is a Galaxy workflow to analyse and discover homologous genes and their corresponding gene families based on the EnsemblCompara GeneTrees pipeline. It lets users interrogate genes of interest without using the command-line while still providing the flexibility to tailor analysis by changing parameters and tools if necessary.

GeneSeqToFamily workflow and required tools are freely available from the Galaxy ToolShed. The workflow can be installed and made available on any Galaxy instance, and a Galaxy can be configured to run locally or on a HPC. The run time of the workflow solely relies on the specification and configuration of the Galaxy host, as well as configuration and usage of the Galaxy instance.

As GeneSeqToFamily workflow developed incorporating existing tools, results of the workflow entirely rely on the chosen tools and parameters especially in the earlier stages of the workflow such as *BLAST* and *hcluster\_sg*, which has been demonstrated in the benchmarking section (see section 3.2.5). Depending on the quality of input dataset

and the application, some parameters and tools might need to be replaced with a more suitable alternative available.

The Galaxy community can undertake their analyses and provide feedback to improve various tools and share combinations of parameters used in the GeneSeqToFamily workflow to achieve better gene families for their datasets. This collaborative approach also offers an opportunity to add additional analytical functionalities to the workflow.

# In-depth interrogation of phylogeny through new visualisation tools

---

In recent years, there has been significant progress in comparative genomics research and development of new software aimed at improving visualisation approaches for phylogenetic information. There have been surprisingly few providing interactive visualisation for detailed changes in gene structure as well as visualising virus phylogeny along with options to distinguish sub-clusters by pairwise distance matrix. Here, I will discuss the development of Aequatus and VicTreeView, two visualisation tools, in section 4.2 and 4.3 based on the following publications:

1. A. S. Thanki, N. Soranzo, J. Herrero, W. Haerty, and R. P. Davey, “Aequatus: an open-source homology browser,” *GigaScience* 2018

For Aequatus project, I defined the overall problem and developed the software solution. I wrote the entire draft version of the paper and revised it according to co-authors comments.

2. S. Modha, A. S. Thanki, S. F. Cotmore, A. J. Davison, and J. Hughes, “Victree: an automated framework for taxonomic classification from protein sequences,” *Bioinformatics* 2018

For the ViCTree project, I designed and developed ViCTreeView to resolve the visualisation problem defined by the first author. I also assisted in drafting and reviewing the paper, and proposed various refinements to the draft proposal made by the first author.

## 4.1 Multi-species genome browsers

### 4.1.1 Ensembl genome browser

The Ensembl genome browser [108] visualises gene trees (see Figure 4.1) through the main Ensembl server. These gene trees are generated using the EnsemblCompara GeneTrees pipeline [41]. The display shows the phylogenetic tree representing the evolutionary history of genes alongside sequence alignment. It represents evolutionary events by using different colours for internal nodes of a phylogenetic tree. The amino acid alignment is shown next to the corresponding node, and the consensus amino acid alignment is shown for the collapsed node.

In the gene tree, the gene of interest is highlighted with red label, and homologues are shown in black and within-species paralogs are shown in blue, if the option to view paralogs is selected (below the tree diagram). The gene tree can be collapsed and expanded by clicking on an internal node. Ensembl also provides predefined options to expand and collapse a node: such as View current gene only, View paralogs of the current gene, View all duplication nodes, View fully expanded tree, Collapse all the nodes at the taxonomic rank and the custom tree.

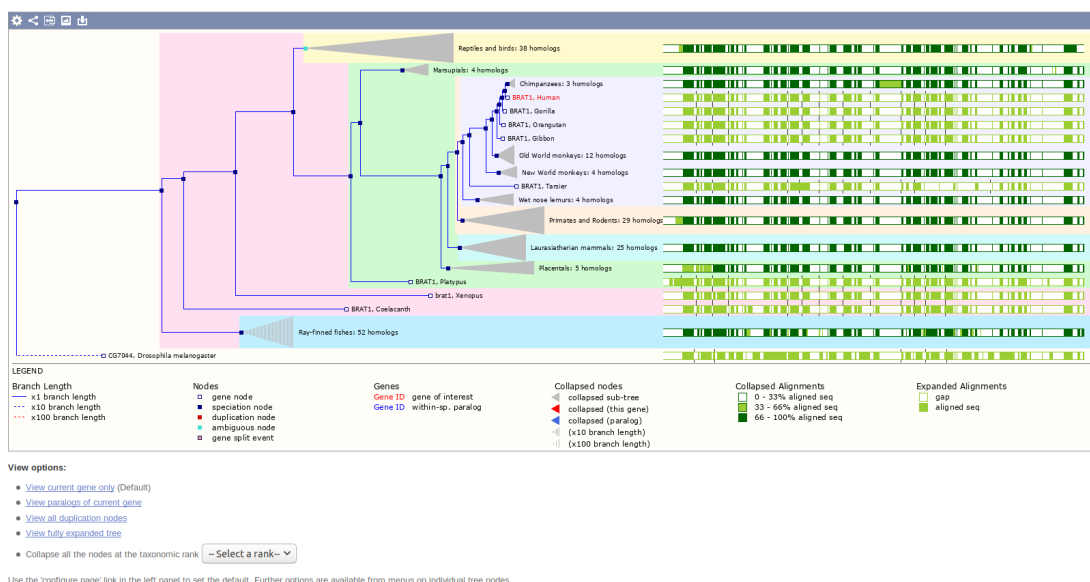


Figure 4.1: Showing an example of BRAT1 gene tree in Ensembl.



### 4.1.2 Genomicus

Genomicus [75] is a multi-species genome browser developed to visualise syntenic genes extracted from the Ensembl databases. It presents syntenic genes and their order along with an interactive collapsible phylogenetic tree on the side. It provides a very informative and interactive way of visualising phylogenetic data; however, it is only able to present an overview of syntenic regions reaching down to the gene order and orientation. Besides, Genomicus also provides two other visualisations for pairwise genome comparison: KaryoView and MatrixView.

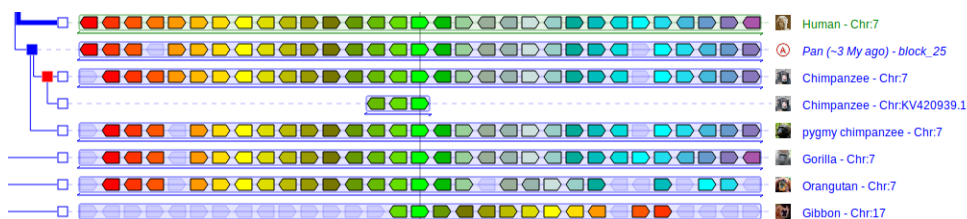


Figure 4.2: Showing the example of homologues for BRAT1 and neighbouring genes in Genomicus.

## 4.2 Aequatus: an open-source homology browser

As useful as these tools are, they only provide an overview of a given phylogeny but do not provide genomic structural differences confirming insertion and deletions in the gene, which is a result of mutation and responsible for evolution. Thus, I have developed Aequatus [109] to overcome these limitations of the currently available resources and bridge the gap between a representation of phylogeny and gene structure comparison.

Aequatus visualises phylogenetic information such as gene tree and homology from the Ensembl Compara and Ensembl Core databases but is not limited to them. It can also visualise gene tree within Galaxy using the results of the GeneSeqToFamily workflow (see section 3.2). To speed up development and to adhere to well-defined and well-used standardised structured data storage formats, I chose not to develop a novel database schema or data format but rather to use the already existing Ensembl database schema. The Ensembl database presents a standardised genomic features schema that has been widely accepted and used by many projects, such as Genoverse [110], easyGWAS [111], and Genomicus [75].

### 4.2.1 Software design

Aequatus is designed with a typical server-client architecture (see section 1.7.4) to avoid any additional effort of setting up a local environment for application installation for the user. Figure 4.3 is showing a simple representation of server-client architecture implemented in Aequatus. Server module of Aequatus is installed on an institutional server and provides a standardised transparent interface to clients. Client computers provide an interface so users can access it remotely with a standard web browser.

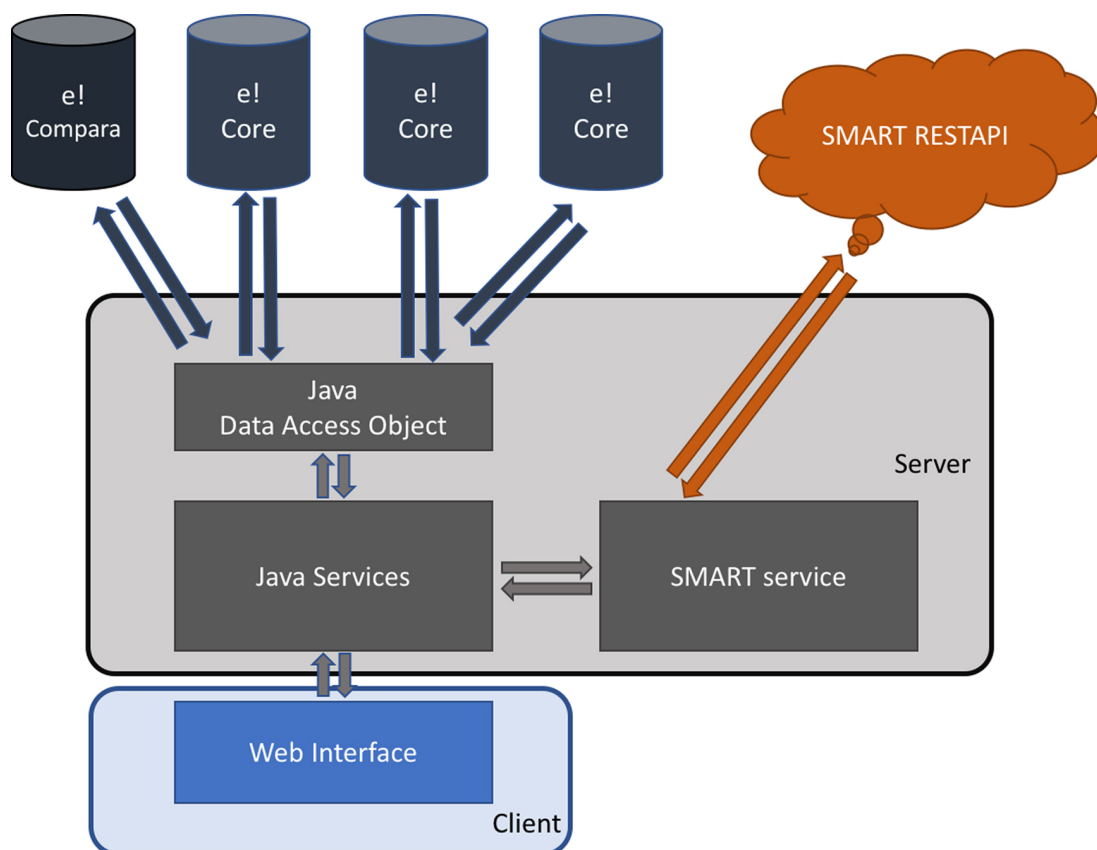
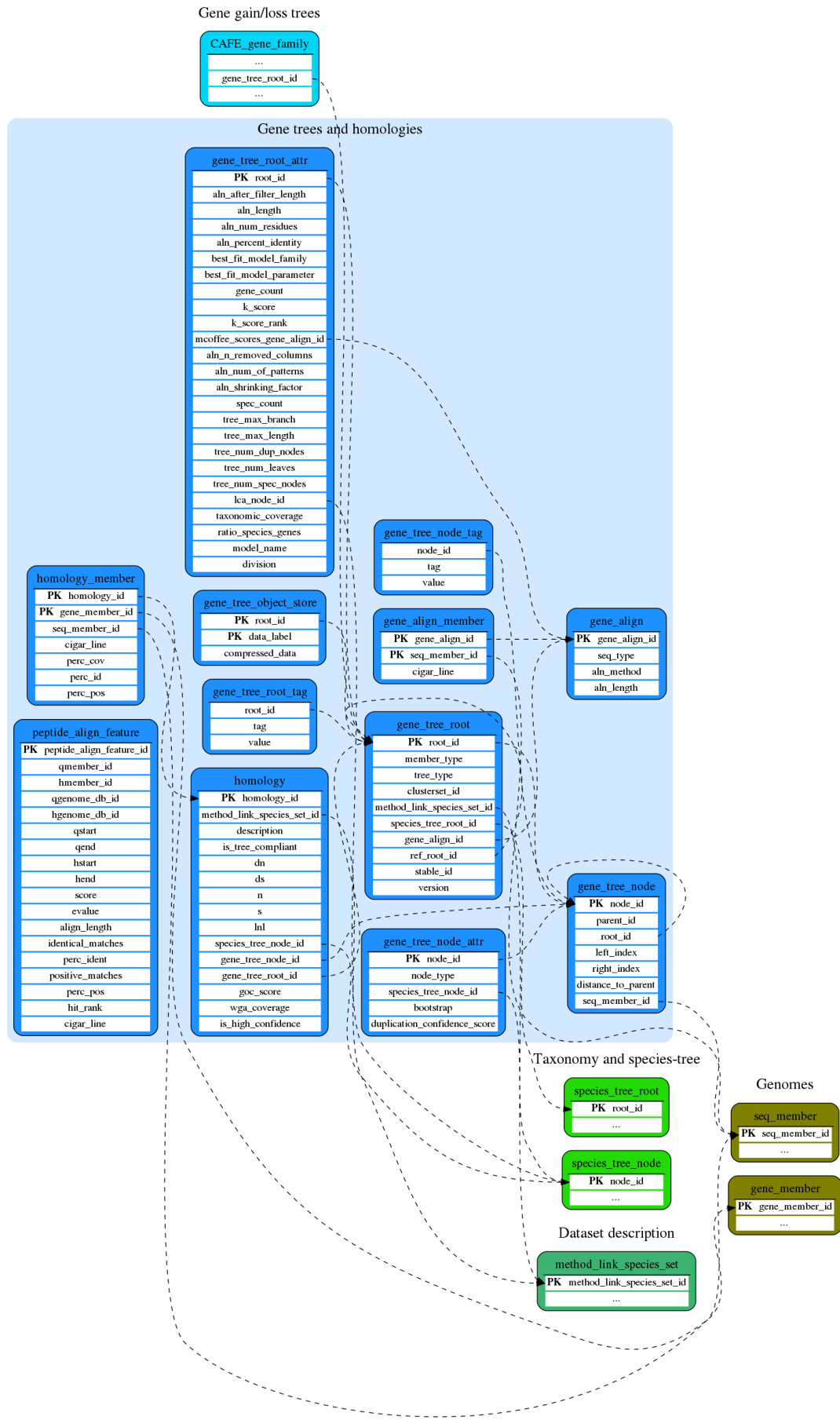


Figure 4.3: Showing the Aequatus infrastructure.

Reproduced from Thanki et al. [109]

#### Server side

The server side of Aequatus is implemented in the Java programming language. It fetches data from the Ensembl databases using Java Data Access Object (DAO). It retrieves phylogenetic information from the Ensembl Compara database and gene structure information from the Ensembl Core databases.



Compara schema diagram: Gene trees and homologies tables

Figure 4.4: Showing the Ensembl Compara database schema for gene trees and homologies.

Reproduced from Ensembl website.

Aequatus retrieves meta information such as genome ID, genome name and chromosome name from `genome_db` and `dnafrag` tables. Gene and protein information are retrieved from `gene_member` and `seq_member` tables respectively.

Relevant information regarding gene tree and alignment for gene members is retrieved by combining phylogenetic information from `gene_align`, `gene_tree_root` and `gene_tree_node` tables (see Figure 4.4). For one-to-one and one-to-many relationships, homologues for the guide gene are retrieved from `homology` and `homology_member` tables of the Ensembl Compara database.

For gene order view (see section 4.2.3), neighbouring genes are retrieved from Ensembl Core databases for guide gene and its homologues. To associate one-to-one relationship for all these genes, `homolog_ids` for each neighbouring gene is retrieved from `homology_member` table.

Gene structure information for each gene is retrieved from `gene`, `transcript`, `transcript_exon`, `exon` and `translation` tables of the Ensembl Core database.

Aequatus converts information from relevant tables into a JSON fragment and sends it to the Aequatus web client using Asynchronous JavaScript And XML (AJAX) technology, a simple representation of the AJAX model implemented in Aequatus is shown in Figure 4.5.

Aequatus is also capable of visualising protein domains for selected gene. It retrieves protein domain information from the Simple Modular Architecture Research Tool (SMART) [112] service using SMART REST API (see section 1.7.4).

### **Client side**

The client side of Aequatus receives data from the server in JSON format and generates visualisations using popular web technologies (e.g. JavaScript, jQuery, D3.js and SVG) to create information-rich and interactive representations of the gene families and comparative information. These web technologies are discussed in section 1.8.1

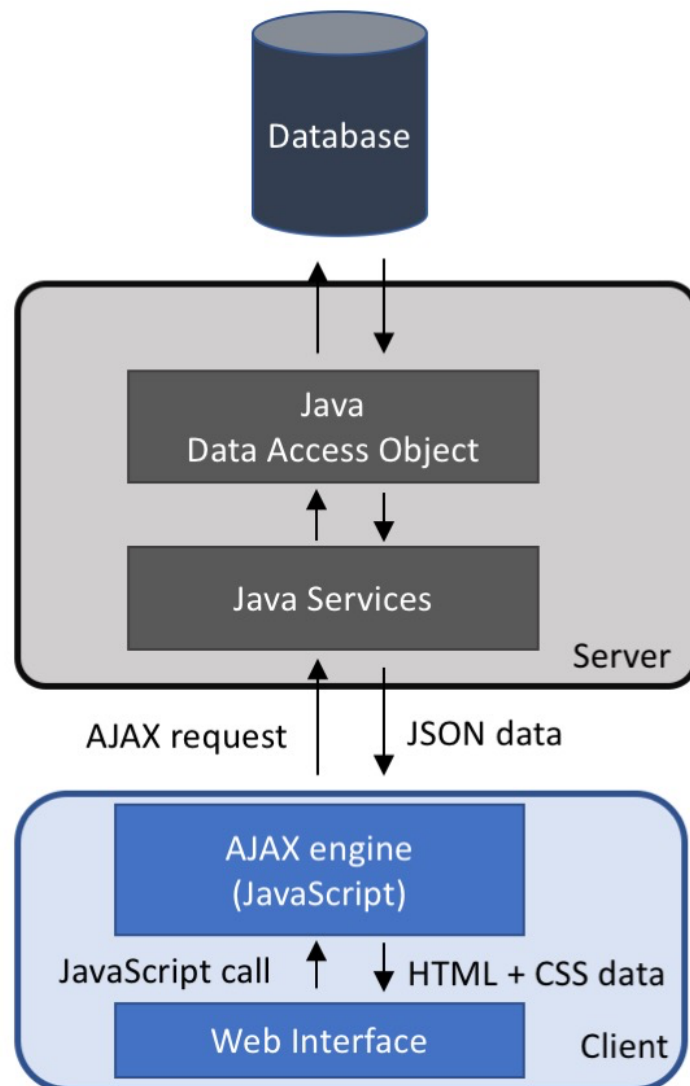


Figure 4.5: Showing the AJAX model implemented in Aequatus.

#### 4.2.2 Interface features

The landing page of Aequatus has a modular design (see Figure 4.6), with a header at the top of the page which provides a search box (see Figure 4.6 A) and a genome list (see Figure 4.6 B). Aequatus includes a chromosomal view providing a karyotype-like overview (see Figure 4.6 C) of a selected reference genome if available and provides a mechanism for users to start to navigate across a genome by selecting a chromosome. Below, there is an overview of genes (see Figure 4.6 D) in selected chromosome with a draggable selector for navigation. Upon selection of a given area, Aequatus produces a gene order view (see Figure 4.6 E, detailed in section 4.2.3), showing genes from the selected region. This is followed by the main Aequatus views detailed in section 4.2.4

and 4.2.5.

Aequatus also has a draggable control panel on the left-hand side (see Figure 4.6 F), which contains options to toggle the visibility of the chromosome view, to access the search box, to access settings for filtering, and exporting options for the respective views, as well as a link to the help pages.

To find a gene of interest, Aequatus employs simple keyword-based search functionality, and it searches against gene symbol, gene ID, or common names for genes or proteins. For example, if a user is interested in BRAT1 genes. The user can search using the keyword 'brat1', and Aequatus returns all the genes that contain the keyword 'brat1' in their name, stable\_id or description. Results are displayed as a list with brief meta information about genes (see Figure 4.6 G) such as gene ID, origin species, gene name, description, and the number of homologues for the search result. Each element in the result has options allowing the user to visualise the corresponding gene tree view or homologous genes in the tabular or Sankey views.

### 4.2.3 Visualising synteny

To support the homology, Aequatus provides gene order representation of the guide gene and its homologous genes. In this view, the selected gene is shown with a red border in the centre between neighbouring genes. Homologues for these genes are shown below with coloured in similar colours as reference genes, respectively (see Figure 4.7). Each homologue is matched to the gene in reference species using unique homology\_id. Mouse over to any gene will highlight its homologous genes.

Conserved gene order adds an extra level of confirmation for gene families because positively selected and real orthologs genes are more likely to retain their position on a genome. Genes that are descended from the same gene are likely to be part of a block of genes and conserve their gene order. Though, due to evolution, there might be some rearrangements of genes that occurs over time.

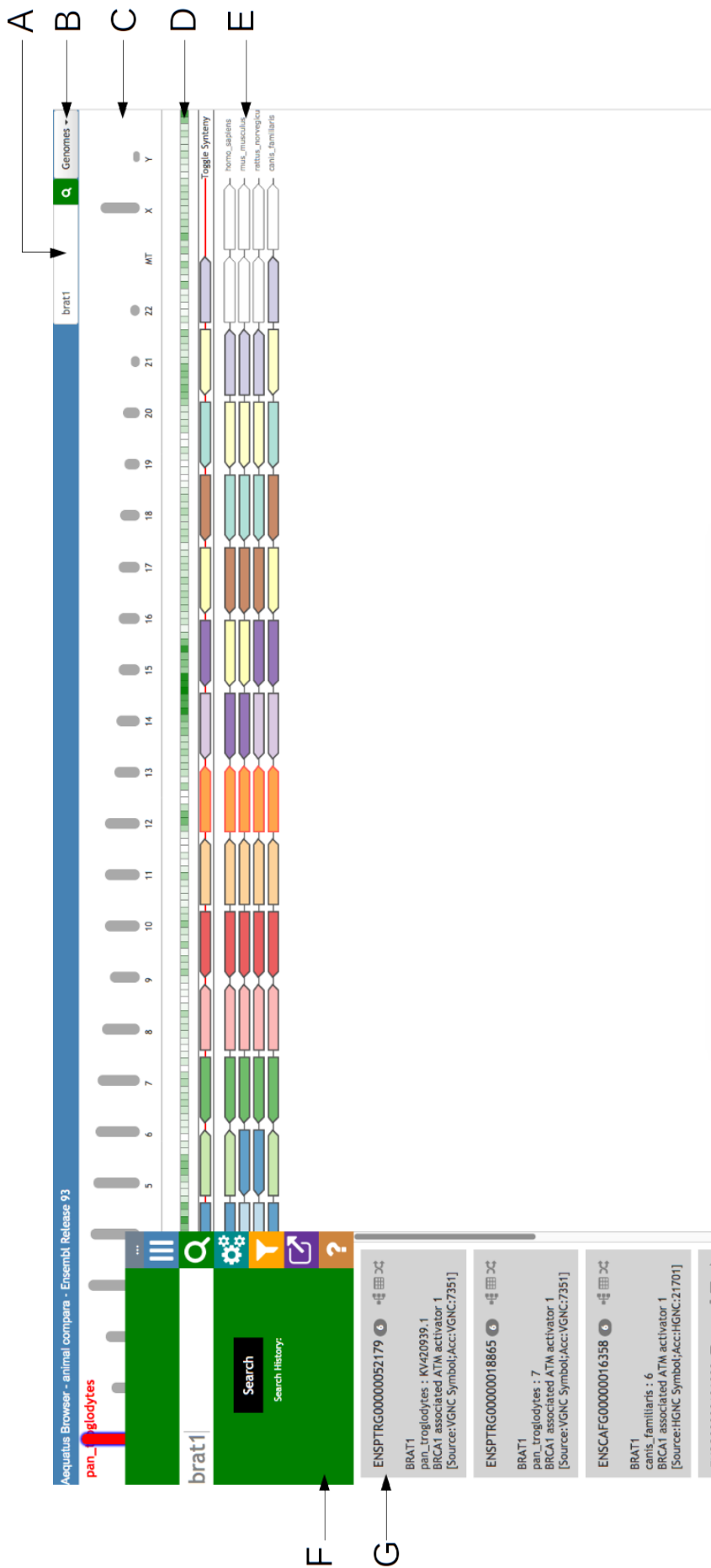


Figure 4-6: The main view of Aequatus. The header bar at the top provides a search box (A) and a genome list (B). It is followed by the chromosomal view (C), where the selected chromosome is coloured in red. Below there is an overview of genes for the selected chromosome (D), followed by a zoomed area of the chromosome with genes shown in the gene order view (E). The Aequatus control panel (F) is visible on the far left showing search result (G) for keyword 'brat1'.

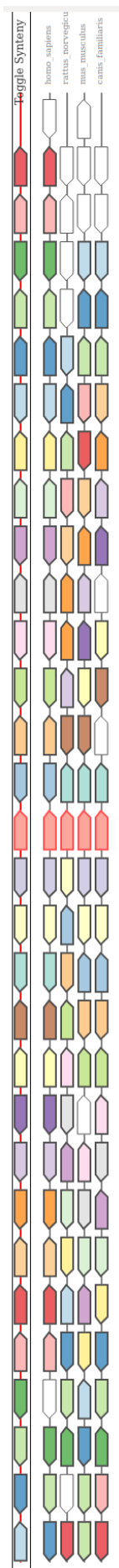


Figure 4.7: Showing Gene order view for the BRAT1 and neighbouring gene. In which BRAT1 gene is in the centre highlighted with red outline. Homologues from other species are drawn below with species name and they are colour coded with genes from reference species.



#### 4.2.4 Visualising gene trees

In the gene tree view (see Figure 4.8a and 4.8b), Aequatus visualises homologous genes by rendering an interactive phylogenetic tree on the left alongside the gene features, represented as colour coding exons, on the right. Internal nodes in the phylogenetic tree are coloured based on potential evolutionary events, such as duplication, speciation, and gene splits.

Due to the pairwise nature of the comparative analysis (see GeneSeqToFamily section 3.2), the visualisation requires that one of the homologous genes returned from a search is set as a guide, or reference, and exons of the guide gene are coloured one by one using qualitative colour scheme chosen from ColorBrewer [113] for readability. Corresponding exons present within the homologous genes are coloured according to the exons of the guide gene using the pre-calculated CIGAR alignment. Matching exons are therefore rendered in the same colour, which helps the investigation into effects of evolutionary events such as gene split, skipping exon, and exon fusions which might have resulted in different functions of the gene. Insertions are shown as black blocks within an exon, and deletions with red lines above an exon. Examples of this are shown in Figure 4.8a and 4.8b, where homologous genes for the *Homo sapiens* (human) gene BRAT1 are shown for the species *Pan troglodytes* (chimpanzee), *Pan paniscus* (bonobo), *Gorilla gorilla* (gorilla), *Pongo abelii* (orangutan), and *Nomascus leucogenys* (gibbon). As shown in Figure 4.8 the Aequatus gene tree view has similar phylogeny as the Ensembl gene tree visualisation and genes from *Pan troglodytes* are showing insertion of two exons in both the genes coloured in black, they are also visible in the Ensembl view (see Figure 4.8c).

The gene tree view in Aequatus has two interchangeable views, whereby the default is the exon focused view (shown in Figure 4.8b), in which introns are presented in fixed width irrespective of their actual length. This allows the user to focus on coding regions as evenly spaced concepts, and larger introns do not dwarf the shorter exons in the visualisation. To also have a full display of all genomic features sizes and distances, another view visualises exons and introns with real length (see Figure 4.8a). Using this mode, the user can compare the length of exons, as the previous visualisation mode provides the exon comparison but does not give a sense for exon length.

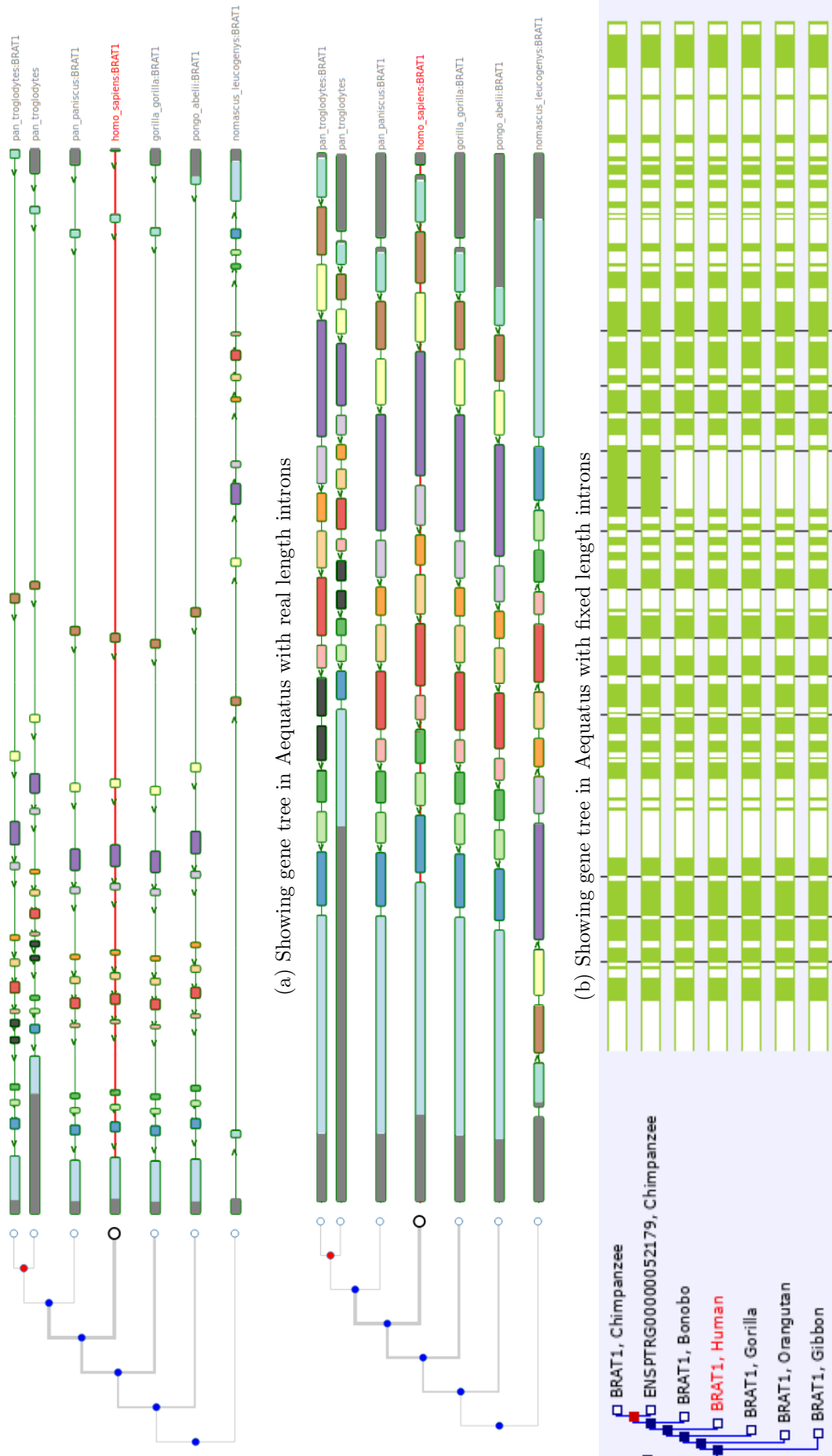


Figure 4.8: Showing gene tree for gene BRAT1 for the *Homo sapiens* (human), *Pan troglodytes* (chimpanzee), *Pan paniscus* (bonobo), *Gorilla gorilla* (gorilla), *Pongo abelii* (orangutan), and *Nomascus leucogenys* (gibbon).

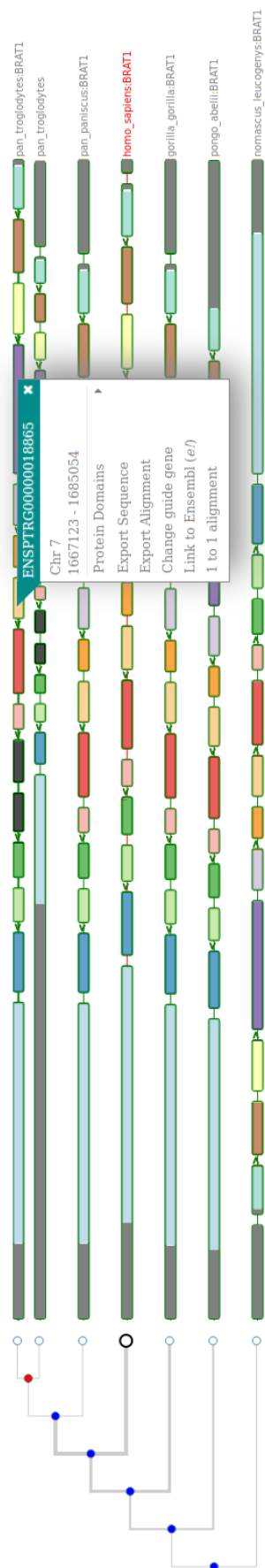


Figure 4.9: Showing Gene tree view for the Homologues of a gene BRAT1. The guide gene is rendered with red coloured label and larger black node. The popup on the right shows additional information for the homologue, along with available operations for the selected gene.

The user can also alter the gene tree view. Nodes of the gene tree can be toggled by the species as well as by the distance from the guide gene. Visuals such as match, insertion and deletion can be toggled, and gene labels can be set to gene name, gene ID or protein ID.

### **Multifunctional pop-up**

Aequatus gene tree view is equipped with a contextual pop-up (see Figure 4.9) containing additional information for the selected gene such as chromosome/scaffold name and locus. It also contains various options:

1. Find protein domains

Protein domains for the selected gene can be retrieved.

2. Visualise pairwise alignment

Pairwise alignment represents a one-to-one comparison of the gene structure of selected gene and guide gene, similar to gene tree view but for two genes, alongside pairwise sequence alignment (see Figure 4.10).

3. Set the selected gene as the guide gene

With the option of **change guide gene** current gene set as guide gene and alignment gets re-mapped with new guide gene.

4. A link to the Ensembl server

This provides a link out using Ensembl IDs to Ensembl gene summary page providing more information.

5. Export sequence, and alignment

With the export option FASTA sequence and CIGAR alignment can be exported to carry out further analysis.

For example, once the user is looking at the gene tree view of BRAT1 gene of *Homo sapiens* as a guide gene and want to change BRAT1 gene of *Pan troglodytes* as a guide gene. User can click on BRAT1 gene of *Pan troglodytes* and click on **change guide gene**. It

will change the guide tree and redraw the gene tree with alignment re-mapped to the new guide gene.

### **Finding and visualising protein domains using SMART**

Protein domains are structural units of proteins that are inferred to result in a potential function or interaction, contributing to the overall character of that protein. Using gene phylogenies along with domain information can help a user to study gain and loss of protein domains [114]. To assess the effect of evolutionary changes on protein domains for the homologous genes, Aequatus has integrated protein domain information using the SMART [112] service.

Simple Modular Architecture Research Tool (SMART) is a curated protein domain resource hosted at European Molecular Biology Laboratory (EMBL) which allows users to search for protein domains with a Sequence ID or Accession, as well as protein sequences. Here, Aequatus uses SMART REST API to query protein sequences and fetch domain information direct from the SMART server programmatically.

Aequatus visualises domains using D3.js together with jQuery DataTable (see Figure 4.11). Domains are mapped on to the coding part of gene separated by red vertical lines and coloured by the types such as Pfam [115], SMART, low complexity region, repeats, and signal peptides. Domains can be ordered and filtered by position, E-value<sup>1</sup> and type of domain. It also allows exporting of visible domains in comma-separated values (CSV) format for further downstream analysis.

Once the user has changed the gene tree view with new guide gene, the user can find protein domains related to the gene by clicking on **Protein Domains** option in the pop-up and then select SMART parameters such as Pfam, Signal peptide, Internal repeat, Internal protein disorder and Homologues. Then click on **Find Domains** to retrieve protein domain information from SMART service and visualise it.

---

<sup>1</sup>E-value is a parameter that describes the number of hits one can “expect” to see by chance when searching a database of a particular size.

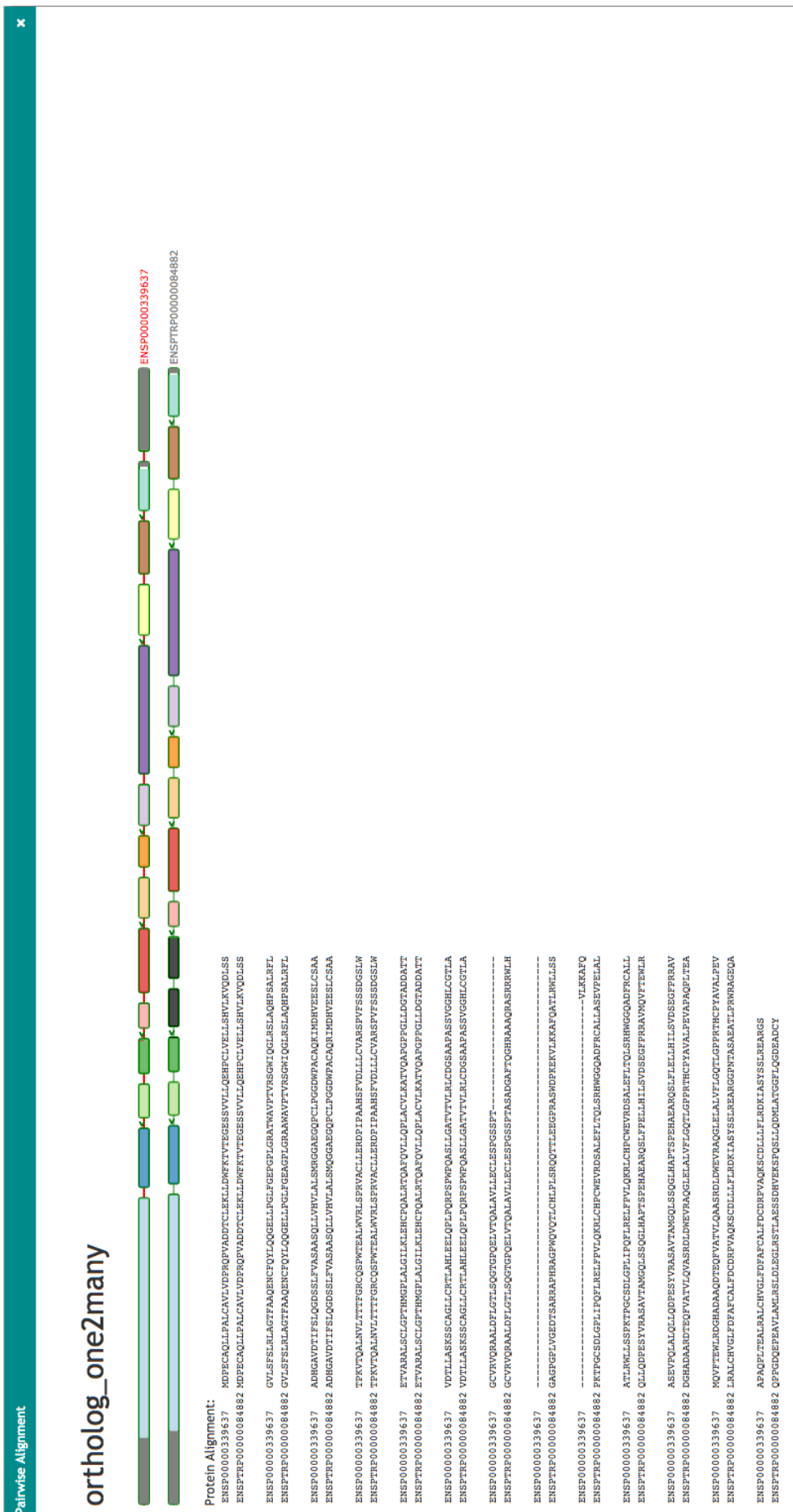


Figure 4.10: Showing Pairwise alignments between homologous genes in *Pan troglodytes* and *Homo sapiens*. Visualising alignment on gene structure on top and visualising pairwise sequence alignments below.

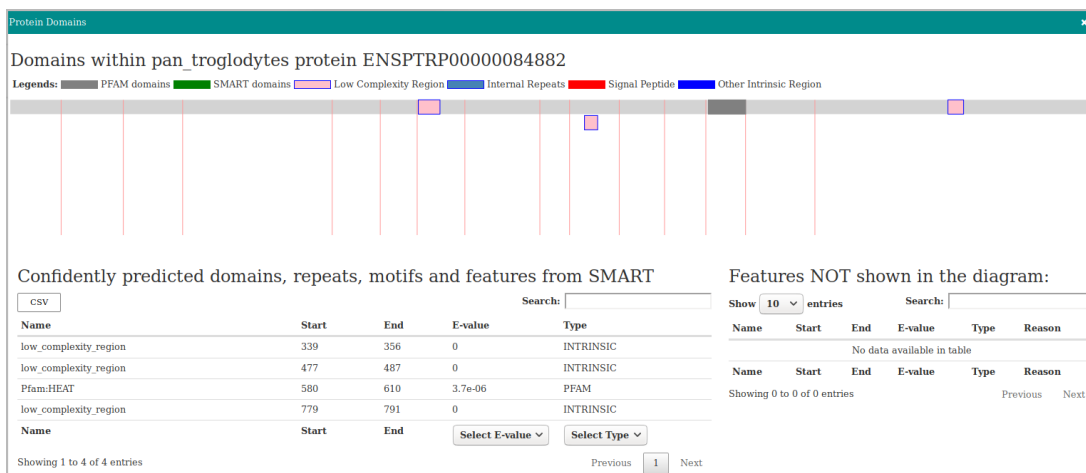


Figure 4.11: Visualisation of the protein domain information for the protein ENSPTRP00000084882 (BRAT1) retrieved from the SMART server. On the top, domains mapped on a CDS. CDS boundaries are shown with red lines. Domains are coloured based on source and type as shown in legends on top. The tables below list the features shown in the diagram as well as hidden features.

#### 4.2.5 Visualising one-to-one and one-to-many relationships

In addition to the gene tree view, Aequatus also visualises one-to-one and one-to-many relationships for the selected gene using interactive Sankey plots and interactive tabular format. Here, Aequatus provides a reference gene based visualisation for homology; thus, it presents only one-to-one and one-to-many relationships for the reference gene.

#### Visualising homologous genes with the Sankey view

Generally, a Sankey diagram is used for illustration of flows. It named after Irish Captain Matthew Henry Phineas Riall Sankey, who used this diagram design for showing the energy efficiency of a steam engine in 1898 [116].

Here, Aequatus uses Sankey plots to represent homology (see Figure 4.12), where flow runs from the selected gene to the homologues (left to right). In Sankey view, homologues are first grouped by the homology type and later by species. Each homologue node is coloured by species to distinguish homologous genes from the same species. The width of the links is proportional to the number of homologues for the type of homologies and the number of homologues per species. Homologues can be filtered by the type of homologies such as one-to-one orthology, one-to-many orthology, or paralogy. Clicking on any homologue in the Sankey view will bring up additional information

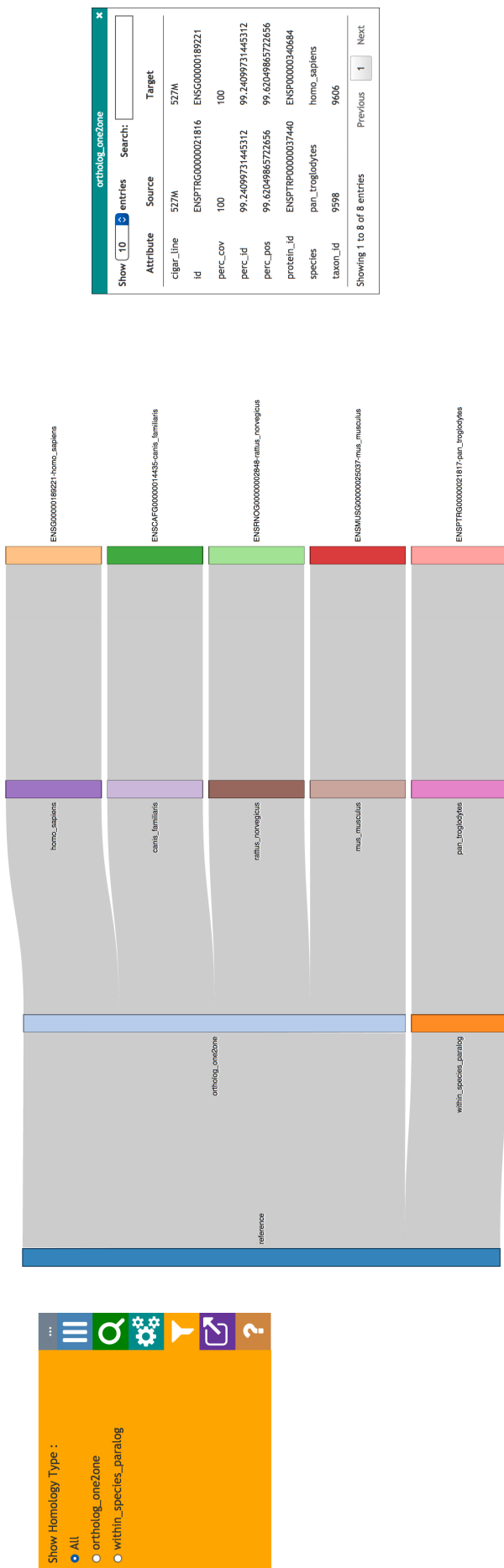


Figure 4.12: Showing sankey view for the homologues of a gene BRAT1, grouped together by type of homology then species. The control panel on the left showing available filters for the Sankey view. Additional information for the homologue is shown in a box on the right.



about the homology pair in a box on the right.

Here, the Sankey view provides a unique way of presenting meta-information about homology at a glance and also allows to look at each homologous pair in detail for pairwise statistical comparison.

### **Visualising homologous genes with tabular view**

Aequatus uses tabular view (see Figure 4.13) to represent homologues for the gene and their statistical information such as coverage, similarity, and identity. This view is generated using jQuery DataTables. The tabular view provides options to order homologues by statistical data as well as filter data based on species and type of homology. These data can be exported in Excel, CSV, and Portable Document Format (PDF) formats.

The tabular view represents additional information using the first and second columns. Pairwise comparative information can be shown by clicking the first column of each homologue. Pairwise alignment comparison on gene structure (similar to gene structure comparison in Figure 4.10) can be visualised by clicking on the second column for each homologue.

This view provides an interactive way of summarising overall comparative information for each homolog and also includes an in-depth pairwise gene structure and statistical comparison.

#### **4.2.6 Aequatus.js, a reusable JavaScript plugin**

Aequatus is designed to work with server and client synchronised architecture to fetch data either from the Ensembl databases or Ensembl server (detailed in section 4.2.7) and generate visualisations. This architecture is not flexible enough, so it can be used by other tools and services to take advantage of Aequatus's innovative way of visualising gene families and homology.

Aequatus.js is developed as an independent JavaScript plugin so that it can be incorporated into any third party web-based services to visualise phylogenetic information.

### Confidently predicted Homology for BRAT1 (Gene: ENSG00000106009)

Show 10 entries
Search:

Detail	Pairwise	Gene ID	Protein ID	Species	dn/ds	Description	Similarity (%)	Identity (%)
		ENSPPAG00000040486	ENSPPAP00000035332	pan_paniscus	0.19231	ortholog_oneZone	99.2692	98.782

Description		Source	Target
Gene ID	ENSG00000106009	ENSG00000040486	ENSPPAG00000040486
Protein ID	ENSP00000339637	ENSP00000035332	ENSPPAP00000035332
Species	homo_sapiens	pan_paniscus	
CIGAR	821M	821M	
Percentage Positivity	99.2692	99.2692	
Percentage Identity	98.782	98.782	

Detail	Pairwise	Gene ID	Protein ID	Species	dn/ds	Description	Similarity (%)	Identity (%)
		ENSPTRG00000052179	ENSPTRP00000085292	pan_troglodytes	0.18576	ortholog_one2many	90.4444	90
		ENSPTRG00000018865	ENSPTRP00000084882	pan_troglodytes	0.15123	ortholog_one2many	90.6667	90.2222
		ENSGGOG00000008928	ENSGGOP00000008732	gorilla_gorilla	0.25962	ortholog_oneZone	98.4166	98.173
		ENSPPYG00000017318	ENSPPYP00000019406	pongo_abelii	0.24715	ortholog_oneZone	97.3203	95.8587
		ENSNLEG00000010655	ENSNLEP00000013000	nomascus_leucogenys	0.17181	ortholog_oneZone	97.4421	96.2241

Detail	Pairwise	Gene ID	Protein ID	Species	dn/ds	Select Type	Similarity (%)	Identity (%)
				All Species gorilla_gorilla nomascus_leucogenys pan_paniscus				

Showing 1 to 6 of 6 entries
Previous  Next

Figure 4.13: Showing Tabular view for the homologues of a gene BRAT1.

Aequatus.js plugin does not require the Aequatus server-side implementation or Ensembl databases. I have demonstrated this by integrating Aequatus.js into the Galaxy platform (available since Galaxy version 19.01) to visualise gene families discovered by GeneSeqToFamily workflow (see section 3.2.2, step 7).

Aequatus.js plugin requires minimal configuration and a JSON input file. A snapshot of input data is shown in Listing 4.1, in which *ref* (line 2) and *protein\_id* (line 3) keywords defines the guide gene. Gene tree is defined with keyword *tree* (line 4) in JSON format and homologous genes are defined with keyword *member* (line 5) in JSON array, both are expected to be in the same JSON structure as available from the Ensembl REST service.

```
1 {
2   ref:<ref gene id>,
3   protein_id:<ref protein id>,
4   tree:<genetree in JSON>,
5   member:<JSON formatted genes array>
6 }
```

Listing 4.1: Input data structure for Aequatus.js

Aequatus.js requires a simple configuration where input JSON data stored in a local variable (line 1) and visualisation initiate by calling the `init` function of `aequatus.js` (line 2) with input data along with div place holders for Settings, Filters and Slider options to render. Finally, gene tree renders using the `drawTree` function (line 3) of `aequatus.js` where inputs are phylogenetic tree, place holder for the tree and a call back function for the click event on homologous genes.

```
1 var syntenic_data = <input JSON>;
2 init(syntenic_data , SettingsDivId , FilterDivId , SliderDivId);
3 drawTree(syntenic_data.tree , GeneTreeDivId , <call back>);
```

Listing 4.2: Snapshot of code to configure Aequatus.js

Implementer can change the guide gene using the provided API (see in Listing 4.3) and renders the gene tree with the new guide gene.

```
1 changeReference(new_gene_id , new_protein_id)
```

Listing 4.3: Snapshot of code to update Aequatus.js

Source code for Aequatus.js is available on GitHub [117] alongside example. At this stage, Aequatus.js is configured to visualise the gene tree view of Aequatus. SMART protein domains, Sankey plot and tabular views will be incorporated in future versions.

#### 4.2.7 Ensembl REST API integration

The initial version of Aequatus requires locally installed Ensembl databases. This is not always preferable because of the need of having and updating local dataset to keep it up to date with the latest Ensembl releases. Therefore, I have integrated Ensembl REST API [56] to retrieve phylogenetic data directly from the Ensembl server to visualise gene tree along with orthologs and paralogs. With the addition of this functionality, Aequatus is available to anyone to be installed on their resources to retrieve and visualise data directly from the Ensembl server.

Aequatus can retrieve data for vertebrates as well as non-vertebrates genome from <http://rest.ensembl.org> and <http://rest.ensemblgenomes.org> respectively. At this stage, Aequatus contains all of the available vertebrates species, and non-vertebrates species have been limited to plants.

I am extending Ensembl REST API further to incorporate the gene order view as well as the chromosomal view. This will allow the Aequatus based on Ensembl REST API to achieve all of the functionalities similar to the Aequatus based on local databases.

#### 4.2.8 Exporting visualisations

Aequatus is also equipped with functionality to export gene tree view and Sankey visuals in the form of SVG and Portable Network Graphic (PNG). This feature can be used to export Aequatus visualisations so they can be used in presentations and publications.

#### 4.2.9 Persistent URL

To enable consistent access to genes of interest, Aequatus provides persistent unique Uniform Resource Locator (URL). This makes it easier to go back in the browser

history, and also to share information with collaborators as well as for use in publications. For example, [http://aequatus.earlham.ac.uk/ensembl\\_rest/index.jsp?query=ENSPTRG00000018865&&view=tree](http://aequatus.earlham.ac.uk/ensembl_rest/index.jsp?query=ENSPTRG00000018865&&view=tree) leads to the gene tree view for BRAT1 gene from chimpanzee (*Pan troglodytes*) and homologous genes. Here, `view` can be set to `tree`, `tabular` or `Sankey`, and `query` can be set to gene ID.

#### 4.2.10 Summary

Aequatus is an open-source web-based homology browser developed to visualise gene families and homologous genes, at a level that was not previously possible within a single tool. It incorporates phylogenetic information with gene feature information to analyse the effect of evolutionary events on gene structure. It also provides two complementary views for one-to-one and one-to-many relationships in the form of interactive Sankey view and tabular view. Aequatus provides the functionality to predict and visualise protein domains using the SMART service. Aequatus also provides persistent unique Uniform Resource Locator (URL) to enable consistent access to genes of interest.

A brief comparison of Aequatus with other phylogenetic visualisation tools (see Table 4.1) shows that Aequatus includes the most combination set of features when compared to available tools by providing several ways of visualising phylogenetic data in a single platform. At the moment, Aequatus does not have a functionality to visualise conserved genomic regions across species, but this will be integrated into Aequatus in the future.

Feature	Aequatus	Ensembl	Genomicus	SyMap	MizBee
<b>Open source tool</b>	✓	✓	Available on request	✓	✓
<b>Gene structural comparison</b>	✓				
<b>Synteny / Gene order</b>	✓		✓	✓	✓
<b>Conserved genomic regions</b>		✓	✓	✓	✓
<b>Sequences alignment</b>	✓	✓		✓	
<b>Export sequence and alignment</b>	✓	✓		✓	
<b>Web based tool</b>	✓	✓	✓	✓	
<b>Local Installation</b>	✓				✓

Table 4.1: Comparison of Aequatus with various phylogenetic visualisation tools.

Reproduced from Thanki et al. [109]

### 4.3 ViCTree: an automated framework for taxonomic classification from protein sequences

A virus is an infectious agent, found wherever there is life. They can infect all types of life forms, replicating inside their host organism to propagate. Typically, they are small in genome size relative to their hosts, and unlike most organisms, virus genomes can be composed of DNA or RNA, be double or single-stranded, be linear or circular, and they can generate either one or multiple proteins [118]. Viruses are classified, by the International Committee on Taxonomy of Viruses (ICTV) (<http://www.ictvonline.org/>), into three ranks: family; genus; and species, and in some cases two further ranks: order; and subfamily. Classification provides a catalogue of the vast diversity of viruses infecting eukaryotes, bacteria and archaea. Due to substantial diversity in virus genomes, various phylogenetic classification tools and workflows, such as the EnsemblCompara GeneTree pipeline [41], PhyOP [38], and OrthoMCL [37], often have limited application in virus classification as they simply are not designed for this purpose.

Tools developed specifically for viral classification are becoming increasingly more and more because of the rapid increase in the amount of sequence data for viruses [119]. DivErsity pARtitioning by hieRarchical Clustering (DEmARC) [120] and Pairwise Sequence Comparison (PASC) [121] uses pairwise distance criteria to classify viral sequences. ViPTree [122] uses genome-wide similarity method to classify viral sequences and build phylogenetic relationships. These tools either provides pairwise distance or the phylogenetic information, but they do not provide the results of pairwise distances along with the phylogeny. This dual feature is essential for viral classification as viral taxonomists are moving from using distance-based to phylogeny-based classification methods; thus distance information allows for validation of the phylogeny-based classification results. Thus, ViCTree was developed to generate phylogenetic trees as well as distance measures.

I developed ViCTreeView [123] to provide interactive visualisations for phylogenetic trees as a part of ViCTree project in collaboration with Sejal Modha and Joseph Hughes from the University of Glasgow, UK.

### 4.3.1 ViCTree pipeline

ViCTree [124] is an open-source pipeline for viral taxonomic classification, incorporating various existing tools into a single pipeline of Bash shell scripts. It can be run on a UNIX based operating system such as Linux and Mac OSX.

The pipeline is divided into eight main steps (an overview flowchart is shown in figure 4.14). Firstly, the pipeline takes a curated set of representative protein sequences and taxonomic IDs as inputs and retrieves the relevant protein sequences from GenBank<sup>2</sup>. These sequences are aligned using *BLAST* [26] and the results filtered using parameters specified by the user, such as the minimum hit and query coverage thresholds. ViCTree then generates clusters with *CD-HIT* [125] on significant BLAST results using a clustering threshold of 0.9 by default. For each cluster, *CD-HIT* assigns the longest sequence in a cluster as representative by default or from an optional list of representatives supplied by the user. An MSA and pairwise distance between sequences are calculated using *Clustal Omega (ClustalO)* [126], which are subsequently used in ViCTreeView. An evolutionary tree is inferred from MSA using *Randomized Accelerated (RAx) Maximum Likelihood (ML) (RAxML)* [127] under a user-defined evolutionary model, or PROTGAMEJTT by default. The evolutionary tree is then inputted to *multi-rate Poisson Tree Processes (mPTP)* [128], a species estimation tool, for automated species delimitation. Finally, ViCTree generates a tree in Newick format, an alignment in FASTA format, a distance matrix (generated by *ClustalO*), a list of clustered sequences, and a metadata file with the GenBank accession numbers and taxonomic names to visualise alternative labels for nodes in the tree. The results generated from ViCTree framework can then be visualised using ViCTreeView [123].

### 4.3.2 ViCTreeView

ViCTreeView is an open-source web-based tool to visualise and explore viral phylogenetic trees. In a similar fashion to the Aequatus, it has a web-based client-side implementation (Section 4.2.1), and it is developed using web technologies such as

---

<sup>2</sup>GenBank is an open access sequence database produced and maintained by the National Center for Biotechnology Information (NCBI) as part of the International Nucleotide Sequence Database Collaboration.

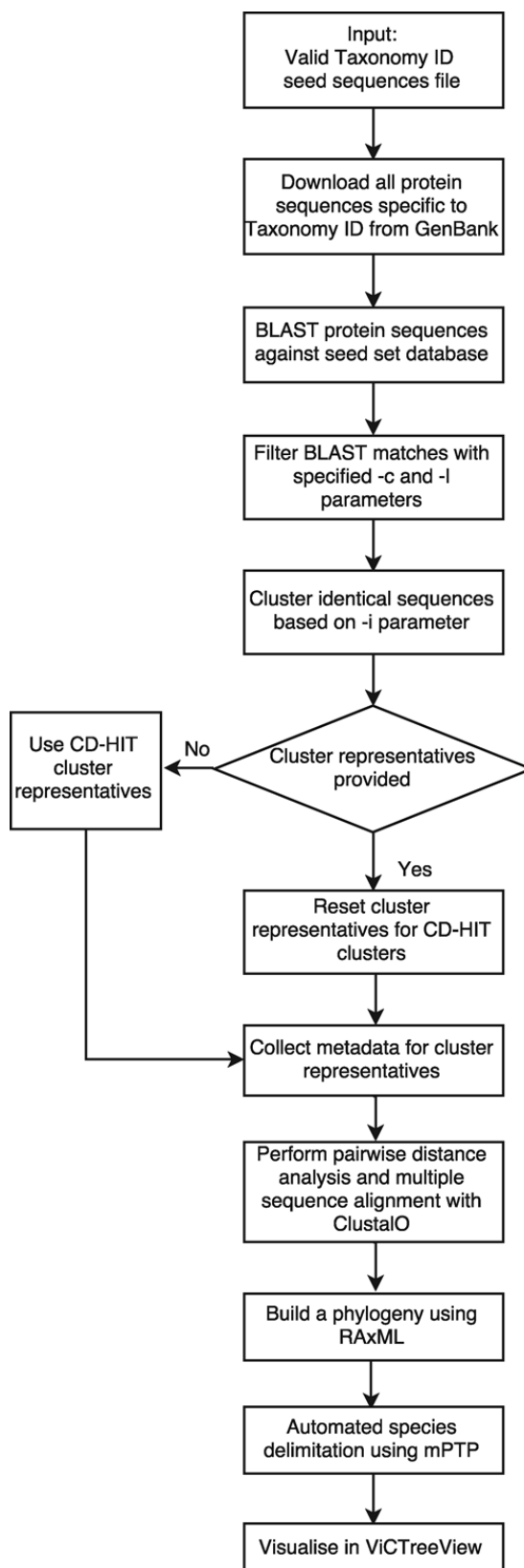


Figure 4.14: Showing overview of ViCTree framework.

Reproduced with permission from Modha et al. [124]



JavaScript, jQuery, Data-Driven Documents JavaScript (D3.js).

## Data source

ViCTreeView retrieves data directly from a configured GitHub repository using the `github.js` library [129]. ViCTreeView uses GitHub, because it can keep track of changes to the data within a configured repository over time and makes it possible to revert to any previous versions if needed. The collaborative development principles of git and GitHub means that it also facilitates sharing data and collaborations with other researchers through versioning of code and data. ViCTreeView requires a phylogenetic tree in Newick format, pairwise distance matrix in CSV format and labels in TSV format. Figure 4.15 showing an example for the data of *Densovirinae* subfamily, *Parvovirinae* subfamily, and *Herpesviridae* family in a defined data structure. ViCTreeView provides a single framework to visualise multiple datasets from the GitHub repository and phylogenetic tree visualisation for these datasets can be toggled by using the example menu in the top-right.

File Name	Last Update	Time Ago
Densovirinae.csv	Jan 2019 update	7 days ago
Densovirinae.nhx	Jan 2019 update	7 days ago
Densovirinae_label.tsv	Jan 2019 update	7 days ago
Herpesviridae.csv	Jan 2019 update	7 days ago
Herpesviridae.nhx	Jan 2019 update	7 days ago
Herpesviridae_label.tsv	Jan 2019 update	7 days ago
Parvovirinae.csv	Jan 2019 update	7 days ago
Parvovirinae.nhx	Jan 2019 update	7 days ago
Parvovirinae_label.tsv	Jan 2019 update	7 days ago

Figure 4.15: Showing input data structure in GitHub repository for ViCTreeView of *Densovirinae* subfamily, *Parvovirinae* subfamily, and *Herpesviridae* family.

## Configuration

A snapshot of the configuration code for ViCTreeView is shown in Listing 4.4. For example, a user wants to load phylogenetic data from the GitHub repository shown in Figure 4.15. Here, the user can configure GitHub user and GitHub repository by setting placeholders for `GitHub user ID` (line 2) as `josephhughes`, `repository name` (line 3)

as ViCTree. The user can also define branch and directory by setting placeholders for `branch name` (line 4) as `master` and `directory name` (line 5) as `ViCTreeView/data`. A URL will be generated using this configuration (line 6), which will be used to fetch files from the repository then files will be processed (line 8 - 10) for visualisation.

```

1 var github = new Github();
2 var user_id = <GitHub user ID>
3 var repo_name = <repository name>
4 var branch = <branch name>
5 var dir = <directory name>
6 var URL = "https://raw.githubusercontent.com/" + user_id + "/" +
    repo_name + "/" + branch + "/" + dir
7 var repo = github.getRepo(user_id, repo_name);
8 var file_list = [];
9 var files = repo.contents(branch, dir, function (err, contents) {
10     contents.forEach(function (file) {
11         ... //process each file
12     })
13 }

```

Listing 4.4: Snapshot of code to configure ViCTreeView

### Interface features

ViCTreeView fetches required data from configured GitHub repository and generates interactive visualisation using popular web-technologies. ViCTreeView visualises phylogenetic tree in the form of interactive phylogram (see Figure 4.17) as well as ultra-metric. In the phylogram view, the length of each branch is relative to the distance from the parent. In the ultra-metric view all branches from the root to a leaf have the same length. The latter is used for a clearer visualisation of all leaves in the tree.

To explore large phylogenetic trees, ViCTreeView provides a zooming function using the mouse scroll wheel. Specific branches can be expanded and collapsed, and vertical distance between branches can be expanded by using the controls in the user interface allowing users to explore selected branches of the tree.

ViCTree provides pairwise distance measures along with phylogeny to validate the phylogeny-based classification. To put this functionality in use, a distance filter has



Figure 4.16: Showing ViCTreeView of the phylogenetic tree for subfamily *Densovirinae* visualised in ViCTreeView. Sequences that fall within the 15% pairwise distance criterion are indicated as distinct clusters in different colours. Black arrows indicate new species identified using ViCTree.

Reproduced with permission from Modha et al. [124]

been integrated into ViCTreeView, which helps to find the sub-cluster(s) of the sequences that fall under a given pairwise distance threshold. A slider in the top of ViCTreeView can be set from 0 to 100, and all the clusters of nodes which fall under this threshold will be highlighted in arbitrary colour to distinguish them from one another. Here, the distance threshold for an internal node of the tree is calculated by finding the maximum distance between two pairs of all the leaves for the node (See Figure 4.17).

ViCTreeView allows users to set any node of the tree to act as the root (by default, the tree is midpoint rooted) and the phylogeny will be redrawn from the perspective of the selected node. This feature enables users to investigate the phylogeny in the context of a particular strain of a virus, represented by their selected root node.

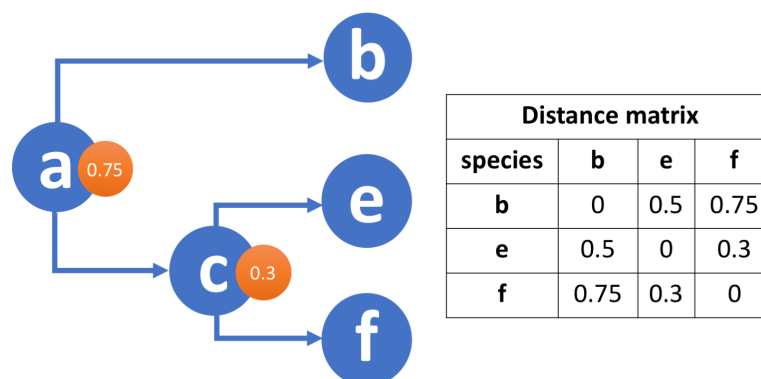


Figure 4.17: Showing example of threshold distance calculation for internal node in ViCTreeView. Threshold distance for node *a* is calculated by finding the maximum distance between all three leaf nodes (*b*, *e* and *f*).

As seen in Figure 4.16, phylogenetic trees for viruses can expand quickly containing complicated branch structure, which can make it difficult to follow a path from a leaf to the root. To overcome this problem, ViCTreeView has implemented an option to highlight the path from the selected node to the root of the tree.

Besides, ViCTreeView can visualise various labels for the leaves of the tree provided by a TSV file, such as GenBank accession numbers, taxonomic IDs, and species or genus names. These can be changed from the menu at the top. Hyperlinks to GenBank protein and nucleic acid sequences can also be visualised as node labels.

Finally, highlighted and customised versions of a tree can be downloaded in SVG and PNG formats for inclusion in publications.

### 4.3.3 Use Case

To demonstrate the application of ViCTree, we present the results for subfamily *Densovirinae* published in the ViCTree manuscript. For this use case, 916 protein sequences were downloaded from GenBank, and a subset of 21 NS1 protein [130] sequences was used as the seed set in a *BLAST* similarity search. *BLAST* output was filtered with hit length set to 100% and query coverage set to 50%, and this resulted in a subset of 187 sequences. These sequences were grouped into 103 distinct clusters using *CD-HIT*. A Representative for each cluster used to perform MSA as well as distance analysis

and phylogeny was built for the aligned sequences.

The in-built automated species delimitation using *mPTP* had identified a total of 25 species from the phylogeny. It identified all 18 previously classified species and genera in subfamily *Densovirinae*. The analysis identified six new species (see Table 4.2, shown in Figure 4.16 with black arrows), which were then submitted to and recognised by the ICTV based on proposals approved by the ICTV Parvoviridae Study Group [131].

Name of new species	Representative isolate	Genus
<i>Asteroid ambidensovirus 1</i>	Sea star-associated densovirus	<i>Ambidensovirus</i>
<i>Decapod ambidensovirus 1</i>	<i>Cherax quadricarinatus densovirus</i>	<i>Ambidensovirus</i>
<i>Hemipteran ambidensovirus 2</i>	<i>Dysaphis plantaginea densovirus 1</i>	<i>Ambidensovirus</i>
<i>Hemipteran ambidensovirus 3</i>	<i>Myzus persicae densovirus 1</i>	<i>Ambidensovirus</i>
<i>Hymenopteran ambidensovirus 1</i>	<i>Solenopsis invicta densovirus</i>	<i>Ambidensovirus</i>
<i>Orthopteran densovirus 1</i>	<i>Acheta domestica mini ambidensovirus</i>	<i>Unassigned</i>

Table 4.2: New species identified in subfamily *Densovirinae* by using ViCTree.

Reproduced with permission from Modha et al. [124]

#### 4.3.4 Summary

ViCTreeView is a unique phylogenetic visualisation tool, which can visualise phylogeny with an option to filter phylogeny by pairwise distance. It provides an interactive platform to integrate and visualise multiple datasets to explore viral phylogenies using a web browser.

The ViCTree pipeline uses existing tools with preset parameters for the test datasets. These parameters might differ for other viral data, and they need to be tested and continuously improved as viral taxonomy expands. Also, the results generated using the framework will be affected by the tools' limitations, which could be improved by other methods. Visualisation functionality of ViCTreeView is not affected by the limitations of the ViCTree pipeline, because ViCTreeView relies on the final result generated by the ViCTree pipeline, not the selection of tools or parameter.

ViCTreeView is an independent JavaScript library, so it can be configured to visualise data from other GitHub repositories and can be integrated into any web-based third-party tool, such as Galaxy.

# 5

## Discussion

---

### 5.1 Available and accessible open-source tools

In recent years, developments in sequencing technologies have fundamentally advanced genomics research. Genomics data generated by sequencing and analytical tools are text-based or in binary format and quantitatively large in size as well as often multi-dimensional. This makes it difficult for an individual to interpret and understand genomics data with the naked human eye. Thus, visualisation tools are needed to understand these interconnected data types. Most existing solutions to visualise genomic data are only available for heavily curated and gold standard genomes held in public resources. Since the Human Genome Project, sequencing technologies became more accessible and affordable, thus lots of smaller scale research institutes and research groups can perform in-house sequencing, including non-model organisms. This created the need for solutions which can visualise genomic data hosted locally, especially for non-model organisms with incomplete genomes and draft gene models. Therefore, I have developed TGAC Browser (see section 2.1) to visualise preliminary assemblies and genomic data from multiple sources such as local Ensembl database, and common NGS file formats (e.g. GFF, VCF, and wig). TGAC Browser demo instances are available from its homepage (<http://browser.earlham.ac.uk/>), and the source code is available from GitHub [132] for anyone to download and install. These features make TGAC Browser accessible to many who are interested in visualising existing model organisms or novel genomes. Similarly, wigExplorer (see section 2.2) implements a modular approach to visualise wiggle data in any JavaScript-enabled web-application. Examples of wigExplorer with BioJS legacy code are available from GitHub page (<http://anilthanki.github.io/old/tools/biojs/wigExplorer/>), and source code

is available from Zenodo<sup>1</sup> (<https://doi.org/10.5281/zenodo.8516>).

Available solutions to perform large scale gene family analysis were designed to function in a dedicated and specifically configured environment requiring various dependencies and computing expertise. Therefore, I have developed GeneSeqToFamily workflow (see section 3.2) to help researchers to find gene families using the Galaxy platform with little or no computing expertise or in-house computing facilities. GeneSeqToFamily also allows researchers to classify, and validate the newly annotated genomics data from non-model organisms by comparing them to gold standard model organisms. Thus, GeneSeqToFamily helps to deliver high-quality annotations and develop the first step of evolutionary analyses.

The GeneSeqToFamily workflow is publicly available from Galaxy's European server (<https://usegalaxy.eu>). The workflow is also available for anyone to download from the Galaxy ToolShed [133] to install it on a local Galaxy instance. Tools and wrappers developed as a part of this workflow are available from Galaxy ToolShed [134] and GitHub repository [135] under the MIT License (<https://opensource.org/licenses/MIT>).

Available solutions to visualise gene families were limited to visualise phylogeny at a higher level as a gene family and gene order but not able to provide a comparison of internal gene structure. Therefore, I developed Aequatus (see section 4.2) to visualise gene families, including a gene structure comparison from the Ensembl Compara and Core databases. A demo for the Aequatus is available from its home page (<http://aequatus.earlham.ac.uk/>) for anyone to explore the latest release of Ensembl data. The source code for the Aequatus and Aequatus.js, a JavaScript visualisation library in Aequatus, is available from GitHub repository [109, 117] under GNU General Public License (GPL) v3 and MIT License respectively.

However, Aequatus is only able to visualise phylogenetic data held in the Ensembl Compara and Core databases, which are highly curated gold standard data for model organisms. To make Aequatus more applicable to a wider range of non-model organism data such as that generated using the GeneSeqToFamily workflow, I have configured

---

<sup>1</sup>Zenodo is a general-purpose open-access repository developed to store data sets, research software, reports, and any other research-related digital artefacts.

the Aequatus.js plugin (see section 4.2.6) to work within Galaxy. The Aequatus plugin has been accepted into main Galaxy source-code since release 19.01 in January 2019 [136], so it is now available from the main Galaxy server (<https://usegalaxy.org>), and EU galaxy server (<https://usegalaxy.eu>) as well as any Galaxy public instance running version 19.01 or later. To date, the main Galaxy server has more than 124,000 registered users [85], and the EU Galaxy server has more than 5000 registered users [137]. This provides Aequatus with a whole new platform to reach a wider audience.

Finally, my contribution to the ViCTree project through the development of ViCTreeView (see section 4.3.2) aligns with my research to develop tools to investigate non-model organisms. The ViCTree pipeline can be used to taxonomically classify newly sequenced viral genomes utilising existing data from GenBank. ViCTree (including ViCTreeView) is an open-source tool available from GitHub repository [138] under GNU GPL v3.0 license. ViCTreeView provides a platform to visualise the findings of the ViCTree pipeline through a web service. The demo is available from the MRC University of Glasgow Centre for Virus Research website (<http://bioinformatics.cvr.ac.uk/victree>), in which it visualises data directly from the ViCTree project's GitHub repository [138] using ViCTreeView.

## 5.2 Use cases and impacts

I have developed several tools (discussed in this thesis) to overcome some of the known issues relating to the analysis and visualisation of non-model organisms. Here, I present some examples of use cases, in which these tools are being used to solve specific biological questions.

The GeneSeqToFamily workflow is being used externally by Dr Ksenia Krasileva's group at UC Davis for total of 18 monocots and dicot plant genomes to identify genes present in terrestrial species but lost in aquatic lineages. This led to the identification genes lost in aquatic species, which are potential candidate components of the plant immune signalling pathway [139]. Similarly, the workflow is also being applied by Dr David Thybert's group at the Earlham Institute on 19 rodent species to find one-to-one orthologs for further downstream analysis of the positive selection of these genes.



I am also involved in discussion with Dr Graham Etherington at the Earlham Institute about applying GeneSeqToFamily workflow to mustelids and ferret datasets to analyse expansion and contraction of immune gene families to quantify gene gain and loss. These various examples provided a real-world opportunity for the GeneSeqToFamily workflow to be used and tested by researchers to investigate evolutionary analysis for the species of their interest.

The GeneSeqToFamily workflow has led to improvements in the development of the Galaxy platform itself. Galaxy supports bundling multiple datasets in “collections”, but was not originally designed to handle collections with a large number of elements (in the order of tens of thousands). As a part of GeneSeqToFamily benchmarking, we used Galaxy collections with a large number of elements, and we (with the help of Dr Nicola Soranzo, EI) found and reported numerous issues (e.g. 3795<sup>2</sup>, and 3883<sup>3</sup>). As a result, the Galaxy development team has optimised the way collections are handled in Galaxy, and any Galaxy installation above the version 18.01 now includes these fixes.

The Ensembl resources hold gold-standard curated datasets, which can be used for phylogenetic analysis. This data needed to import manually in Galaxy from Ensembl. Therefore, I developed the Ensembl suite of tools to retrieve data directly from Ensembl, in order to simplify the data preparation steps necessary for GeneSeqToFamily. These tools can also be used for other analyses. These tools have been cloned/installed for more than 100 times each from Galaxy ToolShed (see Table 5.1), and this does not include the number of clones or downloads from the Earlham Institute Galaxy Tools GitHub repository [135]. This suite of the helper tools provides the groundwork for expansion to integrate other Ensembl REST endpoints.

<b>Tool</b>	<b>Times cloned / installed</b>
<i>Ensembl get feature info</i>	139
<i>Ensembl get genetree</i>	118
<i>Ensembl get sequences</i>	142

Table 5.1: Number of clones / install of Ensembl tools from the Galaxy ToolShed up to 19-June-2019.

Various research projects currently use TGAC Browser within the Earlham Institute

<sup>2</sup>3795: Opening a huge collection in history fails. <https://github.com/galaxyproject/galaxy/issues/3795>

<sup>3</sup>3883: Workflow scheduling doesn't keep track of progress within a step, <https://github.com/galaxyproject/galaxy/issues/3883>

and externally, such as Primula Research Group at the Earlham Institute and the University of Hull, SZN (Stazione Zoologica Anton Dohrn) Napoli, the Brassica RIPR community, EU transPLANT [140], and the Vietnamese Rice community [141].

The Primula Research Group have been using TGAC Browser since 2012 to explore genomic assemblies generated locally. Dr Jinhong Li from this group said (in personal communication) that “TGAC Browser is very useful to search for sequence fragments against denovo genome sequence, which makes it easier to identify gene structure for the genes with large introns, which are impossible to amplify by PCR”.

The transPlant project [140] is funded by the EU to build hardware and software for genomics research aiming to produce an integrated, coherent data infrastructure. It had made progress in sequencing, annotating, and analysing the complex genomes of *Triticaceae* species including bread wheat and barley. The data generated from this project has great potential for applications in breeding, experimental biology, and comparative genomics. Therefore, transPlant project employed the TGAC Browser to visualise the International Barley Sequencing Consortium (IBSC)’s barley genome and the wheat Chromosome Survey Sequence (CSS) genomes along with the mapped SNP markers from the 90K iSelect and Axiom arrays against the International Wheat Genome Sequencing Consortium (IWGSC) CSS contigs, through a collaboration with CerealsDB [142]. transPlant project also employed the Aequatus Browser to visualise gene families from *Brachypodium distachyon*, *Triticum aestivum*, *Aegilops tauschii*, *Oryza sativa*, and *Triticum uratu*. Thus, TGAC Browser and Aequatus played an important role in making these data available from the central transPLANT web hub allowing integrated data access.

The BBSRC Renewable Industrial Products from Rapeseed (RIPR) Programme studies the genes associated with the control of a range of bio-refining targets and fertiliser use traits. One of the aims of the project is to establish data models involving the development of databases and systems enabling the UK Brassica research community to access, analysis and exploit large-scale gene sequence and expression datasets, trait data and marker-trait associations. To support these objectives a TGAC Browser instance was configured to visualise the newly sequenced transcriptome and associated annotation of *Brassica napus*. Aequatus was configured to visualise homology among

*Arabidopsis thaliana*, *Brassica rapa* and *Brassica oleracea*. TGAC Browser and Aequatus played an important role in making these Brassica data available; also, this project inspired TGAC Browser to implement an upload functionality which allowed the breeders to visualise their classified data without sharing it publicly.

### 5.3 Limitations and opportunities for future developments

The tools presented in this thesis have been published and are being used in various domains making their impacts. I am working continuously to improve these tools by identifying limitations and describing possible solutions to them. Here, I also take an opportunity to describe future development to enhance these tools:

#### 5.3.1 Analyse and visualise large datasets

Looking into the near future, these tools need to be prepared for an enormous amount of data is being generated by various international initiatives such as Vertebrate Genome Project [143], Earth Biogenome Project [144], and Darwin Tree of Life project [145].

With the beginning of these and other extensive genome sequencing projects, greater amounts of data will need to be analysed and visualised. This data will provide GeneSeqToFamily and Aequatus with new challenges and opportunities to develop further to enable the analysis of not just tens but tens of thousands of species.

GeneSeqToFamily uses *BLAST* for pre-clustering alignment, and it is acting as a bottleneck for the workflow when analysing large datasets. I have been testing parallel *BLAST* (see section 3.2.6) to speed up the process. Even with this module, it is still taking a significant amount of time to align a large dataset. I am also investigating *DIAMOND* [146], a sequence aligner for protein and translated DNA, to replace *BLAST* for faster performance without sacrificing the results' sensitivity and specificity. *DIAMOND* claims to be up to 20,000 times the speed of *BLAST* retaining high sensitivity, which will allow GeneSeqToFamily to be more efficient for large datasets.

The benefit of GeneSeqToFamily being a modular workflow is that tools used in any of the intermediate steps can be replaced with a potentially better alternative. Similarly,

this is possible for any of the other tools used in the workflow for tailored preparing and/or outputting of for the data.

Aequatus can visualise large gene trees; however, Aequatus could struggle to remain responsive due to a large number of elements to be rendered. Also, Aequatus cannot render genes with a larger number of exons, due to limited horizontal screen size to represent genes. Therefore, representing sub-families of a large gene tree with abstract information and zooming controls for gene structure, along with horizontal scrolling, can be a possible future solution to overcome these limitations.

### 5.3.2 Quality check in GeneSeqToFamily using *Gblocks*

GeneSeqToFamily employs *T-Coffee* to generate MSA of each cluster (see section 3.2.2 step 4). *T-Coffee* can utilise more than one method to perform MSA and produce a single alignment, but the alignment quality can suffer from shortcomings in the handling of insertions and deletions. MSA plays a vital role in defining phylogeny using *TreeBeST* as well as visualisation of gene families using Aequatus to compare homologous genes, in which case checking the quality of the alignment is necessary. Here, I am testing *Gblocks* [147] to incorporate into the GeneSeqToFamily workflow for checking the quality of MSA generated by *T-Coffee*. A preliminary quality check will help to filter out clusters with poor alignment, possibly re-aligning them with suitable parameters in *T-Coffee*, or discarding them entirely based on a pass/fail threshold. Therefore, this solution will improve the quality of the workflow results and eliminate the chances for the failure of the *TreeBeST* and the workflow entirely.

### 5.3.3 Homology identification in GeneSeqToFamily

The GeneSeqToFamily workflow is designed to find homologous genes and infer a phylogeny amongst them, but not to provide one-to-one relationships among them. Therefore, I am developing a Galaxy workflow to investigate this specific scenario. The proposed workflow starts with taking the gene tree (generated by *TreeBeST*, see section 3.2.2 step 5) as input and splitting the gene tree into sub-tree using *ETE GeneTree splitter* [148], then classifying the homologous genes in paralogs and orthologs (one-to-

one, one-to-many and many-to-many) using *Homology Classifier and Filter* tool [148].

I am testing this proposed workflow, as *ETE GeneTree splitter* tool takes a collection of gene tree as input and generate a collection of collections as output. Galaxy has recently provided a fix<sup>4</sup> for the known issue<sup>5</sup> of not being able to handle a nested collection of collections as a part of the workflow. Once tested, the *GAFa* tool will be extended to accommodate results generated from the classification.

### 5.3.4 Ensembl REST extension in Aequatus

The Ensembl REST API integration into Aequatus (see section 4.2.7) helps users to retrieve genomic features and sequences from the Ensembl resources to visualise gene tree and homology. This avoids the need for downloading and creating local Ensembl Compara and Core database instances, which can be hundreds of gigabytes in size, but obviously requiring an active internet connection. Integration of more endpoints from the Ensembl REST API to visualise other comparative genomics features, such as syntenic genomic region alignments, would improve the ability of Aequatus to provide an overview for the comparison of larger genomic regions complementing the gene family, because genes which retain their positions across the species are more likely to be true homologues.

### 5.3.5 Discovery and visualisation of exon duplication

Approximately 10% of human, fly, and worm genes contain tandemly duplicated exons, which contribute to alternative splicing, diversity, and are likely to play an important role in the rapid evolution of eukaryotic genes [149]. The GeneSeqToFamily workflow can identify gene families and homologous genes, but it is not designed to identify duplicated exons within a gene. The analysis of exon duplication can be achieved using sequence similarity alongside exon boundary information, in which sequence for each exon can be retrieved using exon boundaries and perform similarity search to find duplicated exons. The results of exon duplications can be visualised along the

---

<sup>4</sup>Pull request 7633: Delay workflow step execution for discovered mapped-over input. <https://github.com/galaxyproject/galaxy/pull/7633>

<sup>5</sup>Issue 5867: Flatten Collection runs before job that discovers collection elements. <https://github.com/galaxyproject/galaxy/issues/5867>

gene structure. A long term plan for GeneSeqToFamily is to integrate or include a supplementary workflow to identify exon duplication, including tandem duplications within a gene.

### 5.3.6 Containerisation for software

Tools discussed in this thesis are complementing the application of each other, but they operate independently. Using many independent tools in combination can prove to be difficult for a user to conduct analyses and explore datasets effectively. To overcome this problem, I am investigating the use of a virtual system containing all the necessary tools and dependencies using Docker, a computer program designed to create, deploy, and run applications by packaging up an application with all required dependencies. This packaged up application could be made available using CyVerse or other docker compliant clouds. CyVerse provides computational infrastructure enabling data-driven discoveries in life sciences. The virtual system will allow users to automatically download gold-standard curated data from the Ensembl server and upload their data to perform phylogenetic analysis using the GeneSeqToFamily workflow without worrying about the software complexity of joining up the underlying tools. The gene families identified as part of the GeneSeqToFamily workflow can be visualised within Galaxy using the Aequatus plugin and the Ensembl Compara and Ensembl Core database instances for the data would be automatically created to share the findings with collaborators.

By adapting the modern approach to virtualisation and containerisation, a complete collection of tools can be made available to a user to carry out a phylogenetic analysis of newly generated genomic data with existing resources and make it accessible for public use and publication purposes.

## 5.4 Conclusion

This thesis represents my work on the development of several tools to improve the analysis and visualisation of genomic data. These new tools are being used by leading active research groups and making a significant impact on these communities. I am

optimistic that these tools will continue to contribute toward the analysis of newly sequenced genomes over the next few years.

Further development of these tools will allow non-computer experts to perform cutting edge bioinformatics analysis and explore complex results to achieve new findings. The democratisation of sequencing technologies opened up new doors for genomic resources. Thus big sequencing projects to sequence large amount of species are underway. The amount of data generated from these initiatives will be unthinkable and complicated. Here, I am attempting to democratise bioinformatics resources to explore and analyse genomic data and also proposing a possible solution to deal with a tremendous amount of genomic data.

#### **5.4.1 Simple recommendations for writing software/tools for biologists**

There have been many articles and blogs published discussing simple rules or recommendations for writing bioinformatics software and tools [150, 151, 152]. Some may be obvious and considered common sense, but they are often easier said than done in practice. The following is a list of recommendations for researchers in a biological domain based on the experience I have gained during this doctoral study:

1. Do not re-invent the wheel and utilise or adapt existing gold-standard resources.
2. Interact with the user for their expectation and requirements.
3. Benchmark the tool with real use cases.
4. Prepare an easy to follow guide or tutorial for end-user to understand the use and impact of available options.
5. Make your software or tool readily accessible and available for broader exposure.
6. Keep things simple for end-user to access and install a particular tool.
7. Adapt stable technology keeping the future extension of the tool in mind.
8. Use a version control system.

## Acronyms

**Abyss** Assembly By Short Sequencing. 15

**AJAX** Asynchronous JavaScript And XML. 36, 70

**API** Application Programming Interface. 20, 21, 44, 70, 79, 85, 86, 103

**BED** Browser Extensible Data. 35

**BioVis** Biological Data Visualisation. 25

**BiVi** The Biological Visualisation Network. 25

**BLAST** Basic Local Alignment Search Tool. 15

**BRAT1** BRCA1 Associated ATM Activator 1. 57, 59, 72, 87

**CDS** coding sequences. 28, 48–50, 53, 55, 58, 59

**ClustalO** Clustal Omega. 89

**CSS** Chromosome Survey Sequence. 26, 100

**CSV** comma-separated values. 79, 83, 91

**D3.js** Data-Driven Documents JavaScript. 26, 44, 70, 79, 91

**DAO** Data Access Object. 35, 68

**DEmARC** DivErsity pArtitioning by hieRarchical Clustering. 88

**DOM** Document Object Model. 26

**EMBL** European Molecular Biology Laboratory. 79

**EMBOSS** The European Molecular Biology Open Software Suite. 18, 48, 53, 55

**GAFA** Gene alignment and family aggregation. 55, 56, 103

**GAPIT** Genome Association and Prediction Integrated Tool. 39

**GFF** General Feature Format. 35, 36, 50, 96

**GPL** General Public License. 97, 98

**HPC** High-Performance Computing. 18, 29, 41, 63



- HSP** High-scoring Segment Pair. 60
- HTML** Hypertext Markup Language. 26
- HTTP** Hypertext Transfer Protocol. 21
- IBSC** International Barley Sequencing Consortium. 100
- ICTV** International Committee on Taxonomy of Viruses. 88
- IWGSC** International Wheat Genome Sequencing Consortium. 100
- JDBC** Java Database Connectivity. 35
- JSON** JavaScript Object Notation. 36, 50, 59, 70, 85
- MGI** Mouse Genome Informatics. 28
- mPTP** multi-rate Poisson Tree Processes. 89
- MSA** Multiple Sequence Alignment. 54-57, 89, 94, 102
- NCBI** National Center for Biotechnology Information. 41, 58
- NGS** Next-Generation Sequencing. 35, 36, 42, 96
- ONT** Oxford Nanopore Technologies. 13
- PASC** Pairwise Sequence Comparison. 88
- PDF** Portable Document Format. 83
- PNG** Portable Network Graphic. 86, 94
- QfO** Quest for Orthologs. 60, 61
- RAxML** Randomized Accelerated (RAx) Maximum Likelihood (ML). 89
- REST** Representational State Transfer. 20, 21, 70, 79, 86, 103
- RIPR** Renewable Industrial Products from Rapeseed. 100
- SAM** Sequence Alignment/Map format. 36
- SMART** Simple Modular Architecture Research Tool. 70, 79, 81, 87
- SNP** single nucleotide polymorphism. 45

**SOAP** Short Oligonucleotide Alignment Program. 15

**SVG** Scalable Vector Graphics. 26, 70, 86, 94

**TreeBeST** Tree Building guided by Species Tree. 47, 50, 54, 55

**TSV** tab-separated values. 91, 94

**UCSC** University of California Santa Cruz. 27

**URL** Uniform Resource Locator. 17, 86, 87, 92

**USB** Universal Serial Bus. 13

**VCF** Variant Call Format. 35, 36, 96

**VizBi** Visualising Biological Data. 25

## References

---

- [1] J. D. WATSON and F. H. CRICK, “Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid,” *Nature*, vol. 171, pp. 737–738, Apr 1953.
- [2] R. A. Ankeny, “Sequencing the genome from nematode to human: changing methods, changing science,” *Endeavour*, vol. 27, pp. 87–92, Jun 2003.
- [3] F. Sanger, S. Nicklen, and A. R. Coulson, “DNA sequencing with chain-terminating inhibitors,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 74, pp. 5463–5467, Dec 1977.
- [4] A. M. Maxam and W. Gilbert, “A new method for sequencing DNA,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 74, pp. 560–564, Feb 1977.
- [5] A. J. Jeffreys, V. Wilson, and S. L. Thein, “Individual-specific ‘fingerprints’ of human DNA,” *Nature*, vol. 316, no. 6023, pp. 76–79, 1985.
- [6] Y. Smith, “Applications of genomics,” Feb 2019.
- [7] “Genetics vs. genomics fact sheet.” <https://www.genome.gov/about-genomics/fact-sheets/Genetics-vs-Genomics>.
- [8] L. Del Giacco and C. Cattaneo, *Introduction to Genomics*, pp. 79–88. Totowa, NJ: Humana Press, 2012.
- [9] “What is genomics?.” <https://www.ebi.ac.uk/training/online/course/genomics-introduction-ebi-resources/what-genomics>, Jun 2016.

- [10] N. Naidoo, Y. Pawitan, R. Soong, D. N. Cooper, and C. S. Ku, "Human genetics and genomics a decade after the release of the draft sequence of the human genome," *Hum. Genomics*, vol. 5, pp. 577–622, Oct 2011.
- [11] S. Goodwin, J. D. McPherson, and W. R. McCombie, "Coming of age: ten years of next-generation sequencing technologies," *Nat. Rev. Genet.*, vol. 17, pp. 333–351, 05 2016.
- [12] E. E. Schadt, S. Turner, and A. Kasarskis, "A window into third-generation sequencing," *Hum. Mol. Genet.*, vol. 19, pp. R227–240, Oct 2010.
- [13] N. J. Loman, R. V. Misra, T. J. Dallman, C. Constantinidou, S. E. Gharbia, J. Wain, and M. J. Pallen, "Performance comparison of benchtop high-throughput sequencing platforms," *Nat. Biotechnol.*, vol. 30, pp. 434–439, May 2012.
- [14] G. Brown, Clive, "Single molecule 'strand' sequencing using protein nanopores and scalable electronic devices," in *Presentation at AGBT*, 2012. <https://doi.org/10.7490/f1000research.1110935.1>.
- [15] R. M. Leggett, D. Heavens, M. Caccamo, M. D. Clark, and R. P. Davey, "NanoOK: multi-reference alignment analysis of nanopore sequencing data, quality and error profiles," *Bioinformatics*, vol. 32, pp. 142–144, Jan 2016.
- [16] J. Quick, N. J. Loman, S. Duraffour, J. T. Simpson, E. Severi, L. Cowley, J. A. Bore, R. Koundouno, G. Dudas, A. Mikhail, N. Ouedraogo, B. Afrough, A. Bah, J. H. Baum, B. Becker-Ziaja, J. P. Boettcher, M. Cabeza-Cabrerizo, A. Camino-Sanchez, L. L. Carter, J. Doerrbecker, T. Enkirch, I. G. G. Dorival, N. Hetzelt, J. Hinzmann, T. Holm, L. E. Kafetzopoulou, M. Koropogui, A. Kosgey, E. Kuisma, C. H. Logue, A. Mazzarelli, S. Meisel, M. Mertens, J. Michel, D. Ngabo, K. Nitzsche, E. Pallash, L. V. Patrono, J. Portmann, J. G. Repits, N. Y. Rickett, A. Sachse, K. Singethan, I. Vitoriano, R. L. Yemanaberhan, E. G. Zekeng, R. Trina, A. Bello, A. A. Sall, O. Faye, O. Faye, N. Magassouba, C. V. Williams, V. Amburgey, L. Winona, E. Davis, J. Gerlach, F. Washington, V. Monteil, M. Jourdain, M. Bererd, A. Camara, H. Somlare, A. Camara, M. Gerard, G. Bado, B. Baillet, D. Delaune, K. Y. Nebie, A. Diarra, Y. Savane,

- R. B. Pallawo, G. J. Gutierrez, N. Milhano, I. Roger, C. J. Williams, F. Yattara, K. Lewandowski, J. Taylor, P. Rachwal, D. Turner, G. Pollakis, J. A. Hiscox, D. A. Matthews, M. K. O’Shea, A. M. Johnston, D. Wilson, E. Hutley, E. Smit, A. Di Caro, R. Woelfel, K. Stoecker, E. Fleischmann, M. Gabriel, S. A. Weller, L. Koivogui, B. Diallo, S. Keita, A. Rambaut, P. Formenty, S. Gunther, and M. W. Carroll, “Real-time, portable genome sequencing for Ebola surveillance,” *Nature*, vol. 530, pp. 228–232, Feb 2016.
- [17] N. R. Faria, E. C. Sabino, M. R. Nunes, L. C. Alcantara, N. J. Loman, and O. G. Pybus, “Mobile real-time surveillance of Zika virus in Brazil,” *Genome Med*, vol. 8, p. 97, Sep 2016.
- [18] M. Jain, H. E. Olsen, D. J. Turner, D. Stoddart, K. V. Bulazel, B. Paten, D. Hausler, H. F. Willard, M. Akeson, and K. H. Miga, “Linear assembly of a human centromere on the Y chromosome,” *Nat. Biotechnol.*, vol. 36, pp. 321–323, 04 2018.
- [19] M. Mehmood, Aamer, U. Sehar, and N. Ahmad, “Use of Bioinformatics Tools in Different Spheres of Life Sciences,” *J Data Mining Genomics Proteomics*, vol. 5, no. 2, 2014.
- [20] J. T. Simpson, K. Wong, S. D. Jackman, J. E. Schein, S. J. Jones, and I. Birol, “ABySS: a parallel assembler for short read sequence data,” *Genome Res.*, vol. 19, pp. 1117–1123, Jun 2009.
- [21] R. Li, H. Zhu, J. Ruan, W. Qian, X. Fang, Z. Shi, Y. Li, S. Li, G. Shan, K. Kristiansen, S. Li, H. Yang, J. Wang, and J. Wang, “De novo assembly of human genomes with massively parallel short read sequencing,” *Genome Res.*, vol. 20, pp. 265–272, Feb 2010.
- [22] “DISCOVAR: Assemble genomes, find variants.”  
<https://software.broadinstitute.org/software/discovar/blog/>.
- [23] C. Burge and S. Karlin, “Prediction of complete gene structures in human genomic DNA,” *J. Mol. Biol.*, vol. 268, pp. 78–94, Apr 1997.

- [24] D. L. Wheeler, D. M. Church, S. Federhen, A. E. Lash, T. L. Madden, J. U. Pontius, G. D. Schuler, L. M. Schriml, E. Sequeira, T. A. Tatusova, and L. Wagner, “Database resources of the National Center for Biotechnology,” *Nucleic Acids Res.*, vol. 31, pp. 28–33, Jan 2003.
- [25] S. Weckx, J. Del-Favero, R. Rademakers, L. Claes, M. Cruts, P. De Jonghe, C. Van Broeckhoven, and P. De Rijk, “novoSNP, a novel computational tool for sequence variation discovery,” *Genome Res.*, vol. 15, pp. 436–442, Mar 2005.
- [26] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic local alignment search tool,” *J. Mol. Biol.*, vol. 215, no. 3, pp. 403 – 410, 1990.
- [27] S. R. Eddy, “Profile hidden Markov models,” *Bioinformatics*, vol. 14, no. 9, pp. 755–763, 1998.
- [28] J. Stalker, B. Gibbins, P. Meidl, J. Smith, W. Spooner, H.-R. Hotz, and A. V. Cox, “The ensembl web site: mechanics of a genome browser,” *Genome Res.*, vol. 14, pp. 951–955, May 2004.
- [29] J. Thurmond, J. L. Goodman, V. B. Strelets, H. Attrill, L. S. Gramates, S. J. Marygold, B. B. Matthews, G. Millburn, G. Antonazzo, V. Trovisco, T. C. Kaufman, B. R. Calvi, N. Perrimon, S. R. Gelbart, J. Agapite, K. Broll, L. Crosby, G. D. Santos, D. Emmert, L. S. Gramates, K. Falls, V. Jenkins, B. Matthews, C. Sutherland, C. Tabone, P. Zhou, M. Zytkevich, N. Brown, G. Antonazzo, H. Attrill, P. Garapati, A. Holmes, A. Larkin, S. Marygold, G. Millburn, C. Pilgrim, V. Trovisco, P. Urbano, T. Kaufman, B. Calvi, B. Czoch, J. Goodman, V. Strelets, J. Thurmond, R. Cripps, and P. Baker, “FlyBase 2.0: the next generation,” *Nucleic Acids Res.*, vol. 47, pp. D759–D765, Jan 2019.
- [30] R. Y. N. Lee, K. L. Howe, T. W. Harris, V. Arnaboldi, S. Cain, J. Chan, W. J. Chen, P. Davis, S. Gao, C. Grove, R. Kishore, H. M. Muller, C. Nakamura, P. Nuin, M. Paulini, D. Raciti, F. Rodgers, M. Russell, G. Schindelman, M. A. Tuli, K. Van Auken, Q. Wang, G. Williams, A. Wright, K. Yook, M. Berriman, P. Kersey, T. Schedl, L. Stein, and P. W. Sternberg, “WormBase 2017: molting into a new stage,” *Nucleic Acids Res.*, vol. 46, pp. D869–D874, Jan 2018.

- [31] C. E. Cook, M. T. Bergman, R. D. Finn, G. Cochrane, E. Birney, and R. Apweiler, “The European Bioinformatics Institute in 2016: Data growth and integration,” *Nucleic Acids Res.*, vol. 44, pp. D20–26, Jan 2016.
- [32] S. Choudhuri, *Bioinformatics for Beginners*. Academic Press, 2014.
- [33] R. Shamir, “Phylogenetics and phylogenetic trees,” December 2000.
- [34] M. Nei, “Phylogenetic analysis in molecular evolutionary genetics,” *Annu. Rev. Genet.*, vol. 30, pp. 371–403, 1996.
- [35] K. P. O’Brien, M. Remm, and E. L. Sonnhammer, “Inparanoid: a comprehensive database of eukaryotic orthologs,” *Nucleic Acids Res.*, vol. 33, pp. D476–480, Jan 2005.
- [36] E. V. Kriventseva, D. Kuznetsov, F. Tegenfeldt, M. Manni, R. Dias, F. A. Simao, and E. M. Zdobnov, “OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs,” *Nucleic Acids Res.*, vol. 47, pp. D807–D811, Jan 2019.
- [37] L. Li, C. J. Stoeckert, and D. S. Roos, “OrthoMCL: identification of ortholog groups for eukaryotic genomes,” *Genome Res.*, vol. 13, pp. 2178–2189, Sep 2003.
- [38] L. Goodstadt and C. P. Ponting, “Phylogenetic reconstruction of orthology, paralogy, and conserved synteny for dog and human,” *PLoS Comput. Biol.*, vol. 2, p. e133, Sep 2006.
- [39] J. Ruan, H. Li, Z. Chen, A. Coghlan, L. J. Coin, Y. Guo, J. K. Heriche, Y. Hu, K. Kristiansen, R. Li, T. Liu, A. Moses, J. Qin, S. Vang, A. J. Vilella, A. Ureta-Vidal, L. Bolund, J. Wang, and R. Durbin, “TreeFam: 2008 Update,” *Nucleic Acids Res.*, vol. 36, pp. D735–740, Jan 2008.
- [40] M. D. Whiteside, G. L. Winsor, M. R. Laird, and F. S. Brinkman, “OrtholugeDB: a bacterial and archaeal orthology resource for improved comparative genomic analysis,” *Nucleic Acids Res.*, vol. 41, pp. D366–376, Jan 2013.
- [41] A. J. Vilella, J. Severin, A. Ureta-Vidal, L. Heng, R. Durbin, and E. Birney, “EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates,” *Genome Res.*, vol. 19, pp. 327–335, Feb 2009.

- [42] J. D. Wren, “Bioinformatics programs are 31-fold over-represented among the highest impact scientific papers of the past two decades,” *Bioinformatics*, vol. 32, pp. 2686–2691, 09 2016.
- [43] F. Markowetz, “All biology is computational biology,” *PLoS Biol.*, vol. 15, pp. 1–4, 03 2017.
- [44] V. Stodden, J. Seiler, and Z. Ma, “An empirical analysis of journal policy effectiveness for computational reproducibility,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 115, pp. 2584–2589, 03 2018.
- [45] B. K. Beaulieu-Jones and C. S. Greene, “Reproducibility of computational workflows is automated using continuous analysis,” *Nat. Biotechnol.*, vol. 35, pp. 342–346, 04 2017.
- [46] S. Mangul, L. S. Martin, E. Eskin, and R. Blekhman, “Improving the usability and archival stability of bioinformatics software,” *Genome Biol.*, vol. 20, p. 47, 02 2019.
- [47] B. Gruning, R. Dale, A. Sjodin, B. A. Chapman, J. Rowe, C. H. Tomkins-Tinch, R. Valieris, and J. Koster, “Bioconda: sustainable and comprehensive software distribution for the life sciences,” *Nat. Methods*, vol. 15, pp. 475–476, Jul 2018.
- [48] J. T. Robinson, H. Thorvaldsdottir, W. Winckler, M. Guttman, E. S. Lander, G. Getz, and J. P. Mesirov, “Integrative genomics viewer,” *Nat. Biotechnol.*, vol. 29, pp. 24–26, Jan 2011.
- [49] R. Buels, E. Yao, C. M. Diesh, R. D. Hayes, M. Munoz-Torres, G. Helt, D. M. Goodstein, C. G. Elsik, S. E. Lewis, L. Stein, and I. H. Holmes, “JBrowse: a dynamic web platform for genome visualization and analysis,” *Genome Biol.*, vol. 17, p. 66, Apr. 2016.
- [50] B. Giardine, C. Riemer, R. C. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, D. Blankenberg, I. Albert, J. Taylor, W. Miller, W. J. Kent, and A. Nekrutenko, “Galaxy: a platform for interactive large-scale genome analysis,” *Genome Res.*, vol. 15, pp. 1451–1455, Oct 2005.



- [51] P. Rice, I. Longden, and A. Bleasby, “EMBOSS: the European Molecular Biology Open Software Suite,” *Trends Genet.*, vol. 16, pp. 276–277, Jun 2000.
- [52] A. Sharma, M. Kumar, and S. Agarwal, “A complete survey on software architectural styles and patterns,” *Procedia Computer Science*, vol. 70, pp. 16 – 28, 2015. Proceedings of the 4th International Conference on Eco-friendly Computing and Communication Systems.
- [53] “Client-server architecture.” <https://sites.google.com/site/clientserverarchitecture/>.
- [54] M. Kumar, “N-tier application architecture,” Jun 2012.
- [55] R. T. Fielding, *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.
- [56] A. Yates, K. Beal, S. Keenan, W. McLaren, M. Pignatelli, G. R. Ritchie, M. Ruffier, K. Taylor, A. Vullo, and P. Flicek, “The Ensembl REST API: Ensembl Data for Any Language,” *Bioinformatics*, vol. 31, pp. 143–145, Jan 2015.
- [57] M. A. C. Gatto, “Making research useful: Current challenges and good practices in data visualisation,” May 2015.
- [58] J. Aerts, “Data visualisation - an introduction,” 2015.
- [59] L. D. Stein, C. Mungall, S. Shu, M. Caudy, M. Mangone, A. Day, E. Nickerson, J. E. Stajich, T. W. Harris, A. Arva, and S. Lewis, “The generic genome browser: a building block for a model organism system database,” *Genome Res.*, vol. 12, pp. 1599–1610, Oct. 2002.
- [60] A. M. Waterhouse, J. B. Procter, D. M. Martin, M. Clamp, and G. J. Barton, “Jalview Version 2—a multiple sequence alignment editor and analysis workbench,” *Bioinformatics*, vol. 25, pp. 1189–1191, May 2009.
- [61] A. P. Francisco, C. Vaz, P. T. Monteiro, J. Melo-Cristino, M. Ramirez, and J. A. Carrico, “PHYLOViZ: phylogenetic inference and data visualization for sequence based typing methods,” *BMC Bioinformatics*, vol. 13, p. 87, May 2012.
- [62] R. D. Page, “Tree View: An application to display phylogenetic trees on personal computers,” *Bioinformatics*, vol. 12, pp. 357–358, 08 1996.

- [63] R. Jiansi, L. Jiantao, W. Lizhe, and C. Dan, “Data Visualization in Bioinformatics,” *Advances in Information Sciences and Service Sciences*, vol. 4, pp. 157–165, Dec 2012.
- [64] N. Iliinsky and J. Steele, *Designing data visualizations*, ch. 4. Choose Appropriate Visual Encodings. OReilly, 2011.
- [65] T. Munzner, *Visualization analysis and design*. CRC Press, Taylor Francis Group, A K Peters, Ltd., 2015.
- [66] “The avocado toast index: How many breakfasts to buy a house?,” May 2017.
- [67] C. Görg, L. Hunter, J. Kennedy, S. O’Donoghue, and J. J. V. Wijk, “Biological Data Visualization (Dagstuhl Seminar 12372),” *Dagstuhl Rep*, vol. 2, no. 9, pp. 131–164, 2013.
- [68] M. Bostock, V. Ogievetsky, and J. Heer, “D<sup>3</sup>: Data-Driven Documents,” *IEEE Trans Vis Comput Graph*, vol. 17, pp. 2301–2309, Dec 2011.
- [69] S. Murray, *Interactive Data Visualization for the Web: An Introduction to Designing with*. “O’Reilly Media, Inc.”, Aug. 2017.
- [70] J. Heer, “Raising the bar (chart),” in *Keynote at OpenVis Conference*, 2015. <https://homes.cs.washington.edu/~jheer/talks/RaisingTheBar-OpenVisConf.pdf>.
- [71] J. Wang, L. Kong, G. Gao, and J. Luo, “A brief introduction to web-based genome browsers,” *Brief. Bioinformatics*, vol. 14, pp. 131–143, jul 2012.
- [72] W. J. Kent, “The human genome browser at UCSC,” *Genome Res.*, vol. 12, no. 6, pp. 996–1006, 2002.
- [73] “Mouse Genome Database (MGD) at the Mouse Genome Informatics website.” <http://www.informatics.jax.org>, 2019.
- [74] J. Wang, L. Kong, S. Zhao, H. Zhang, L. Tang, Z. Li, X. Gu, J. Luo, and G. Gao, “Rice-Map: a new-generation rice genome browser,” *BMC Genomics*, vol. 12, p. 165, Mar 2011.

- [75] M. Muffato, A. Louis, C. E. Poisnel, and H. Roest Crolius, “Genomicus: a database and a browser to study gene synteny in modern and ancestral genomes,” *Bioinformatics*, vol. 26, pp. 1119–1121, Apr 2010.
- [76] S. J. McKay, I. A. Vergara, and J. E. Stajich, “Using the Generic Synteny Browser (GBrowse\_syn),” *Curr Protoc Bioinformatics*, vol. Chapter 9, p. Unit 9.12, Sep 2010.
- [77] V. Curcin and M. Ghanem, “Scientific workflow systems - can one size fit all?,” in *2008 Cairo International Biomedical Engineering Conference*, pp. 1–9, Dec 2008.
- [78] A. Oram, *Managing Projects with make*. OReilly Associates Inc., 1991.
- [79] D. Merkel, “Docker: Lightweight linux containers for consistent development and deployment,” *Linux J.*, vol. 2014, Mar. 2014.
- [80] J. Leipzig, “A review of bioinformatic pipeline frameworks,” *Brief. Bioinformatics*, vol. 18, pp. 530–536, 05 2017.
- [81] J. Köster and S. Rahmann, “Snakemake—a scalable bioinformatics workflow engine,” *Bioinformatics*, vol. 28, pp. 2520–2522, 08 2012.
- [82] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, “Jupyter notebooks – a publishing format for reproducible computational workflows,” in *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (F. Loizides and B. Schmidt, eds.), pp. 87 – 90, IOS Press, 2016.
- [83] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, J. Bhagat, K. Belhajjame, F. Bacall, A. Hardisty, A. Nieva de la Hidalga, M. P. Balcazar Vargas, S. Sufi, and C. Goble, “The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud,” *Nucleic Acids Res.*, vol. 41, pp. W557–561, Jul 2013.
- [84] P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, “Nextflow enables reproducible computational workflows,” *Nat.*

- Biotechnol.*, vol. 35, pp. 316–319, 04 2017.
- [85] E. Afgan, D. Baker, B. Batut, M. van den Beek, D. Bouvier, M. Cech, J. Chilton, D. Clements, N. Coraor, B. A. Gruning, A. Guerler, J. Hillman-Jackson, S. Hiltmann, V. Jalili, H. Rasche, N. Soranzo, J. Goecks, J. Taylor, A. Nekrutenko, and D. Blankenberg, “The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update,” *Nucleic Acids Res.*, vol. 46, pp. W537–W544, Jul 2018.
- [86] J. Goecks, C. Eberhard, T. Too, A. Nekrutenko, and J. Taylor, “Web-based visual analysis for high-throughput genomics,” *BMC Genomics*, vol. 14, p. 397, Jun 2013.
- [87] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra, “Circos: an information aesthetic for comparative genomics,” *Genome Res.*, vol. 19, pp. 1639–1645, Sep 2009.
- [88] J. Goecks, N. Coraor, A. Nekrutenko, and J. Taylor, “NGS analyses by visualization with Trackster,” *Nat. Biotechnol.*, vol. 30, pp. 1036–1039, Nov 2012.
- [89] A. S. Thanki, X. Bian, and R. P. Davey, “TGAC Browser: an open-source genome browser for non-model organisms,” *bioRxiv*, 2019.
- [90] J. Gomez, L. J. Garcia, G. A. Salazar, J. Villaveces, S. Gore, A. Garcia, M. J. Martin, G. Launay, R. Alcantara, N. Del-Toro, M. Dumousseau, S. Orchard, S. Velankar, H. Hermjakob, C. Zong, P. Ping, M. Corpas, and R. C. Jimenez, “BioJS: an open source JavaScript framework for biological data visualization,” *Bioinformatics*, vol. 29, pp. 1103–1104, Apr 2013.
- [91] timpalant, “timpalant/java-genomics-io.” <https://github.com/timpalant/java-genomics-io>. Accessed: 2016-5-17.
- [92] A. S. Thanki, R. C. Jimenez, G. G. Kaithakottil, M. Corpas, and R. P. Davey, “wigExplorer, a BioJS component to visualise wig data,” *F1000Res*, vol. 3, p. 53, 2014.
- [93] S. Wilzbach, “biojs-vis-wigexplorer.” <https://biojs.net/#/component/biojs-vis-wigexplorer>, 2016.

- [94] D. Scofield, “hcluster\_sg, a tool for hierarchically clustering on a sparse graph.” <https://github.com/douglasgscfield/hcluster>, 2019. Maintained by Douglas G. Scofield, but originally developed by Heng Li.
- [95] H. Li, *Constructing the TreeFam database*. PhD thesis, Chinese Academy of Sciences Beijing, 2006.
- [96] C. Notredame, D. G. Higgins, and J. Heringa, “T-Coffee: A novel method for fast and accurate multiple sequence alignment,” *J. Mol. Biol.*, vol. 302, pp. 205–217, Sep 2000.
- [97] “TreeBeST: Tree Building guided by Species Tree (Ensembl Compara modifications).” <https://github.com/Ensembl/treebest>, 2019.
- [98] A. S. Thanki, N. Soranzo, W. Haerty, and R. P. Davey, “GeneSeqToFamily: a Galaxy workflow to find gene families based on the Ensembl Compara GeneTrees pipeline,” *GigaScience*, vol. 7, pp. 1–10, 03 2018.
- [99] “BLAST parser: Galaxy Tool Shed.” [https://toolshed.g2.bx.psu.edu/view/earlhaminst/blast\\_parser/](https://toolshed.g2.bx.psu.edu/view/earlhaminst/blast_parser/), 2019.
- [100] “hcluster\_sg parser: Galaxy Tool Shed.” [https://toolshed.g2.bx.psu.edu/view/earlhaminst/hcluster\\_sg\\_parser/](https://toolshed.g2.bx.psu.edu/view/earlhaminst/hcluster_sg_parser/), 2019.
- [101] “A suite of Ensembl-REST tools: Galaxy Tool Shed.” [https://toolshed.g2.bx.psu.edu/repos/earlhaminst/ensembl\\_rest/](https://toolshed.g2.bx.psu.edu/repos/earlhaminst/ensembl_rest/), 2019.
- [102] “NCBI BLAST plus: Galaxy Tool Shed.” [https://toolshed.g2.bx.psu.edu/view/devteam/ncbi\\_blast\\_plus](https://toolshed.g2.bx.psu.edu/view/devteam/ncbi_blast_plus), 2019.
- [103] “T-Coffee: Galaxy Tool Shed.” [https://toolshed.g2.bx.psu.edu/view/earlhaminst/t\\_coffee/](https://toolshed.g2.bx.psu.edu/view/earlhaminst/t_coffee/), 2019.
- [104] “Filter FASTA on the headers and/or the sequences : Galaxy Tool Shed.” [https://toolshed.g2.bx.psu.edu/repos/galaxyp/filter\\_by\\_fasta\\_ids](https://toolshed.g2.bx.psu.edu/repos/galaxyp/filter_by_fasta_ids), 2019.
- [105] “UniProt ID mapping and sequence retrieval : Galaxy Tool Shed.” [https://toolshed.g2.bx.psu.edu/repos/bgruening/uniprot\\_rest\\_interface](https://toolshed.g2.bx.psu.edu/repos/bgruening/uniprot_rest_interface), 2019.

- [106] A. Kuzniar, R. C. van Ham, S. Pongor, and J. A. Leunissen, “The quest for orthologs: finding the corresponding gene across genomes,” *Trends Genet.*, vol. 24, pp. 539–551, Nov 2008.
- [107] “faSplit : Galaxy Tool Shed.” [https://toolshed.g2.bx.psu.edu/repos/iuc/ucsc\\_fasplit](https://toolshed.g2.bx.psu.edu/repos/iuc/ucsc_fasplit), 2019.
- [108] “The Ensembl Browser.” <http://www.ensembl.org>, 2019.
- [109] A. S. Thanki, N. Soranzo, J. Herrero, W. Haerty, and R. P. Davey, “Aequatus: an open-source homology browser,” *GigaScience*, vol. 7, 11 2018.
- [110] “Genoverse - interactive HTML5 genome browser.” <http://wtsi-web.github.io/Genoverse/>, 2019.
- [111] D. G. Grimm, D. Roqueiro, P. A. Salome, S. Kleeberger, B. Greshake, W. Zhu, C. Liu, C. Lippert, O. Stegle, B. Scholkopf, D. Weigel, and K. M. Borgwardt, “easyGWAS: A Cloud-Based Platform for Comparing the Results of Genome-Wide Association Studies,” *Plant Cell*, vol. 29, pp. 5–19, 01 2017.
- [112] C. P. Ponting, J. Schultz, F. Milpetz, and P. Bork, “SMART: identification and annotation of domains from signalling and extracellular protein sequences,” *Nucleic Acids Res.*, vol. 27, pp. 229–232, Jan 1999.
- [113] “ColorBrewer 2.0 - color advice for cartography.” <http://colorbrewer2.org/>, 2019.
- [114] M. Buljan and A. Bateman, “The evolution of protein domain families,” *Biochem. Soc. Trans.*, vol. 37, pp. 751–755, Aug 2009.
- [115] E. L. Sonnhammer, S. R. Eddy, and R. Durbin, “Pfam: a comprehensive database of protein domain families based on seed alignments,” *Proteins*, vol. 28, pp. 405–420, Jul 1997.
- [116] A. B. W. Kennedy and H. R. Sankey, “THE THERMAL EFFICIENCY OF STEAM ENGINES. Report of the Committee appointed on the 31st March, 1896, to Consider and Report to the council upon the Subject of the Definition of a Standard or Standards of Thermal Efficiency for Steam-Engines: with an

- Introductory Note,” *Minutes of the Proceedings of the Institution of Civil Engineers*, vol. 134, pp. 278–312, Jan 1898.
- [117] A. S. Thanki, “Aequatus.js, visualisation javascript library for homologous genes.” <https://github.com/anilthanki/aequatus.js>, 2019.
- [118] P. Aiewsakun, E. M. Adriaenssens, R. Lavigne, A. M. Kropinski, and P. Simmonds, “Evaluation of the genomic diversity of viruses infecting bacteria, archaea and eukaryotes using a common bioinformatic platform: steps towards a unified taxonomy,” *J. Gen. Virol.*, vol. 99, pp. 1331–1343, Sep 2018.
- [119] “ICTV: Taxonomy Release History.” [https://talk.ictvonline.org/taxonomy/p/taxonomy\\_releases](https://talk.ictvonline.org/taxonomy/p/taxonomy_releases), 2019.
- [120] C. Lauber and A. E. Gorbalenya, “Partitioning the genetic diversity of a virus family: approach and evaluation through a case study of picornaviruses,” *J. Virol.*, vol. 86, pp. 3890–3904, Apr 2012.
- [121] Y. Bao, V. Chetvernin, and T. Tatusova, “Improvements to pairwise sequence comparison (PASC): a genome-based web tool for virus classification,” *Arch. Virol.*, vol. 159, pp. 3293–3304, Dec 2014.
- [122] Y. Nishimura, T. Yoshida, M. Kuronishi, H. Uehara, H. Ogata, and S. Goto, “ViPTree: the viral proteomic tree server,” *Bioinformatics*, vol. 33, pp. 2379–2380, Aug 2017.
- [123] A. S. Thanki, “ViCTreeView, a visualisation plugin for victree.” <https://github.com/anilthanki/ViCTreeView>, 2019.
- [124] S. Modha, A. S. Thanki, S. F. Cotmore, A. J. Davison, and J. Hughes, “ViCTree: an automated framework for taxonomic classification from protein sequences,” *Bioinformatics*, vol. 34, no. 13, pp. 2195–2200, 2018.
- [125] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li, “CD-HIT: accelerated for clustering the next-generation sequencing data,” *Bioinformatics*, vol. 28, pp. 3150–3152, Dec 2012.
- [126] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Soding, J. D. Thompson, and D. G. Higgins,

- “Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega,” *Mol. Syst. Biol.*, vol. 7, p. 539, Oct 2011.
- [127] A. Stamatakis, “RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies,” *Bioinformatics*, vol. 30, pp. 1312–1313, May 2014.
- [128] P. Kapli, S. Lutteropp, J. Zhang, K. Kobert, P. Pavlidis, A. Stamatakis, and T. Flouri, “Multi-rate Poisson tree processes for single-locus species delimitation under maximum likelihood and Markov chain Monte Carlo,” *Bioinformatics*, vol. 33, pp. 1630–1638, Jun 2017.
- [129] A. Michael, “Github.js.” <https://github.com/github-tools/github>, 2019.
- [130] S. F. Cotmore, M. Agbandje-McKenna, J. A. Chiorini, D. V. Mukha, D. J. Pintel, J. Qiu, M. Soderlund-Venermo, P. Tattersall, P. Tijssen, D. Gatherer, and A. J. Davison, “The family Parvoviridae,” *Arch. Virol.*, vol. 159, pp. 1239–1247, May 2014.
- [131] S. F. Cotmore, M. Agbandje-McKenna, M. Canuti, J. A. Chiorini, A. M. Eishubinger, J. Hughes, M. Mietzsch, S. Modha, M. Ogliastro, J. Péntzes, D. Pintel, J. Qiu, M. Soderlund-Venermo, P. Tattersall, P. Tijssen, and ICTV Report Consortium, “ICTV Virus Taxonomy Profile: Parvoviridae,” *J. Gen. Virol.*, Jan 2019.
- [132] A. S. Thanki, X. Bian, and R. P. Davey, “TGAC Browser, an open-source genome browser.” <https://github.com/TGAC/TGACBrowser>, 2019.
- [133] D. Blankenberg, G. Von Kuster, E. Bouvier, D. Baker, E. Afgan, N. Stoler, J. Taylor, and A. Nekrutenko, “Dissemination of scientific software with Galaxy ToolShed,” *Genome Biol.*, vol. 15, p. 403, Feb 2014.
- [134] “Repositories Owned by earlhaminst.” [https://toolshed.g2.bx.psu.edu/repository/browse\\_repositories\\_by\\_user?user\\_id=e7d772b367350baf](https://toolshed.g2.bx.psu.edu/repository/browse_repositories_by_user?user_id=e7d772b367350baf), 2019.
- [135] “Galaxy tools developed at the Earlham Institute.” <https://github.com/TGAC/earlham-galaxytools>, 2019.



- [136] “Galaxy Release 19.01.” <https://github.com/galaxyproject/galaxy/releases/tag/v19.01>, 2019.
- [137] “We passed the 5.000.000th jobs and 5.000 users!” <https://galaxyproject.eu/posts/2019/04/03/5k-users-5M-jobs/>, 2019.
- [138] “ViCTree - GitHub Repository.” <https://github.com/josephhughes/ViCTree>, 2019.
- [139] E. L. Baggs, A. S. Thanki, R. O’Grady, C. Schudoma, W. Haerty, and K. V. Krasileva, “Convergent gene loss in aquatic plants predicts new components of plant immunity and drought response,” *bioRxiv*, 2019.
- [140] M. Spannagl, M. Alaux, M. Lange, D. M. Bolser, K. C. Bader, T. Letellier, E. Kimmel, R. Flores, C. Pommier, A. Kerhornou, B. Walts, T. Nussbaumer, C. Grabmuller, J. Chen, C. Colmsee, S. Beier, M. Mascher, T. Schmutzer, D. Arend, A. Thanki, R. Ramirez-Gonzalez, M. Ayling, S. Ayling, M. Caccamo, K. F. Mayer, U. Scholz, D. Steinbach, H. Quesneville, and P. J. Kersey, “trans-PLANT Resources for Triticeae Genomic Data,” *Plant Genome*, vol. 9, 03 2016.
- [141] “Sequencing the genomes of a number of native vietnam rice lines.” <http://www.riceagi.org.vn/phenotype/home/index.php?language=2>.
- [142] P. A. Wilkinson, M. O. Winfield, G. L. Barker, A. M. Allen, A. BurrIDGE, J. A. Coghill, and K. J. Edwards, “CerealsDB 2.0: an integrated resource for plant breeders and scientists,” *BMC Bioinformatics*, vol. 13, p. 219, Sep 2012.
- [143] “Vertebrate Gnome Project.” <https://vertebrategenomesproject.org/>.
- [144] “Earth Biogenome Project.” <https://www.earthbiogenome.org/>.
- [145] “The Darwin Tree of Life Project.” <https://www.sanger.ac.uk/news/view/genetic-code-66000-uk-species-be-sequenced>.
- [146] B. Buchfink, C. Xie, and D. H. Huson, “Fast and sensitive protein alignment using DIAMOND,” *Nat. Methods*, vol. 12, pp. 59–60, Jan 2015.
- [147] J. Castresana, “Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis,” *Mol. Biol. Evol.*, vol. 17, pp. 540–552, Apr 2000.

- [148] “Analyse phylogenetic trees using the ETE toolkit: Galaxy Tool Shed.” <https://github.com/TGAC/earlham-galaxytools/tree/master/tools/ete>, 2019.
- [149] I. Letunic, R. R. Copley, and P. Bork, “Common exon duplication in animals and its role in alternative splicing,” *Hum. Mol. Genet.*, vol. 11, pp. 1561–1567, Jun 2002.
- [150] A. Prlić and J. B. Procter, “Ten simple rules for the open development of scientific software,” *PLOS Computational Biology*, vol. 8, pp. 1–3, 12 2012.
- [151] M. List, P. Ebert, and F. Albrecht, “Ten simple rules for developing usable software in computational biology,” *PLOS Computational Biology*, vol. 13, pp. 1–5, 01 2017.
- [152] M. Taschuk and G. Wilson, “Ten simple rules for making research software more robust,” *PLOS Computational Biology*, vol. 13, pp. 1–10, 04 2017.

## Appendix I: Letters of Support

---

31 May 2019

To Whom it May Concerns,

Mr Anil Thanki is the lead developer of the Primula **TGAC Browser** I have been using. To me (and other members in our research group who are non-bioinformaticians), the TGAC Browser is particularly useful to analyse the assembled novel sequence data and to understand the meaning of it, which enable me to further investigate any interesting findings in biological experiments. I would like to describe The Primula TGAC Browser is a bridge tool between bioinformaticians and experimental biologists, which help the latter to have a better understanding of sequence data, to provide feedback to the bioinformaticians on the tools could be further improved. I believe a bioinformatic tool such as the TGAC Browser should have a wider application potential because understanding the sequence data by biologists is fundamentally important toward functional genomics. The Primula TGAC Browser is successful and very useful, which demonstrated Mr Thanki's capacity, knowledge and experience in this field toward a PhD degree. The TGAC Browser helped me to identify the Primula Super Gene from our sequence data, which was published in Nature Plants (see publication details below). More importantly, I am working closely with him to further improve the performance of the TGAC Browser for our ongoing collaboration in research.

I fully support Mr Thanki's submission to gain a PhD by publication. Title of his thesis: Development of computational techniques for genomic data analysis and visualisation in the model and non-model organisms.

Ref: Jinhong Li, Jonathan M. Cocker, Jonathan Wright, Margaret A. Webster, Mark McMullan, Sarah Dyer, David Swarbreck, Mario Caccamo, Cock van Oosterhout & Philip M. Gilmartin. Genetic architecture and evolution of the S locus supergene in *Primula vulgaris*, Nature Plants Vol 2 (2016) DOI: 10.1038/NPLANTS.2016.188 *Acknowledgement*: We thank ... **A. Thanki** for TGAC Browser support; ...

Please do not hesitate to contact me for more information.

Yours sincerely,



Dr Jinhong Li  
Research Fellow  
Biology  
School of Environmental Science  
Faculty of Science and Engineering  
University of Hull  
Cottingham Road  
Hull HU6 7RX  
Email: [Jinhong.li@earlham.ac.uk](mailto:Jinhong.li@earlham.ac.uk); [Jinhong.li@hull.ac.uk](mailto:Jinhong.li@hull.ac.uk)  
Telephone: 01603 450 963

Cambridge 29-05-19

Re: Anil Thanki's PhD thesis:  
Development of computational techniques for genomic data analysis and visualisation in the model  
and non-model organisms

To whom it may concern,

During the years 2014-2016 I've had the pleasure to interact with Anil Thanki in my role as the BioJavaScript (BioJS) community coordinator. BioJS is an open source community of developers whose objective is the sharing of code for visualisation of biological data, mainly through the means of JavaScript, the leading scripting language for the web. With regards to the publication below, I certify that Anil played a leading role in the development of wigExplorer, which is reflected as him being the lead author in this article.

1. wigExplorer, a BioJS component to visualise wig data Anil S. Thanki, Rafael C. Jimenez, Gemy G. Kaithakottil, Manuel Corpas, Robert P. Davey. F1000 Research, F1000Research 2016,

It is worth mentioning that his contribution was beyond purely technical. Anil has also contributed to BioJS as a mentor during Google Summer of Code (GSoc) 2014.

I am pleased to endorse Anil in his bid to obtain a PhD by publication.

Yours faithfully



Manuel Corpas

Chief Scientist  
Cambridge Precision Medicine

+44 7939 807 507  
m.corpas@cpm.onl

ideaSpace South  
University of Cambridge Biomedical Innovation Hub  
Hills Road, Cambridge CB2 0AH  
<https://www.cpm.onl>

UNIVERSITY OF CALIFORNIA, BERKELEY

BERKELEY • DAVIS • IRVINE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO



SANTA BARBARA • SANTA CRUZ

May 8th, 2019

To Whom It May Concern:

Re: Convergent gene loss in aquatic plants predicts new components of plant immunity and drought response.  
EL Baggs, AS Thanki, R O'Grady, C Schudoma, W Haerty, KV Krasileva, 2019 March ;  
doi: <https://doi.org/10.1101/572560>

It is my pleasure to endorse Mr. Anil Thanki bid to gain a PhD by publication. I worked closely with Anil on a project that led to the paper cited above whilst at the Earlham Institute. Anil was instrumental in the application of the GeneSeqToFamily pipeline to analyse the gene families identifiable among the 18 genomes in the aforementioned study. His advice on the method and support throughout the process was professional and extremely useful to the project.

Please do not hesitate to contact me for more information.

Sincerely,

A handwritten signature in black ink that reads 'Erin Baggs'.

Erin Baggs  
(UC Berkeley Graduate student)

320 Koshland Hall,  
Berkeley, CA 94709  
[Erinbaggs95@berkeley.edu](mailto:Erinbaggs95@berkeley.edu)



May 28<sup>th</sup>, 2019

Re: Anil Thanki PhD thesis: Development of computational techniques for genomic data analysis and visualisation in the model and non-model organisms

To whom it may concern,

I am writing to express my wholehearted support of Anil Thanki's project. We have collaborated with Anil to use his GeneSeqToFamily workflow for our research. The results from his pipeline proved to be robust and are included in our publication with Anil as a co-author (Baggs, Thanki, O'Grady, Schudoma, Haerty and Krasileva, "Convergent gene loss in aquatic plants predicts new components of plant immunity and drought response" under review). During this work, we were able to directly compare his pipeline to a standard OrthoMCL analysis methods. Anil's pipeline was not only easier to run, but also it required less computational resources allowing us to process larger datasets.

I recommend Anil's work most favorably. If you have further questions, please, feel free to contact me at [kseniak@berkeley.edu](mailto:kseniak@berkeley.edu) or at (510)-820-4991.

Sincerely yours,

A handwritten signature in black ink, appearing to read 'Ksenia'.

**Ksenia Krasileva**  
Assistant Professor  
Department of Plant and Microbial Biology  
University of California, Berkeley

231 Koshland Hall  
Berkeley, CA 94710

<https://krasilevalab.org/>  
<http://plantandmicrobiology.berkeley.edu>  
@kseniakrasileva

Nicola Soranzo, PhD  
Data Infrastructure and  
Algorithms  
Earlham Institute  
[nicola.soranzo@earlham.ac.uk](mailto:nicola.soranzo@earlham.ac.uk)

3<sup>rd</sup> June 2019

To whom it may concern

**Re: Anil S. Thanki's PhD thesis "Development of computational techniques for genomic data analysis and visualisation in the model and non-model organisms"**

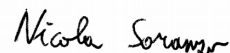
I worked closely with Anil Thanki on the 2 peer-reviewed papers listed below. For both of these projects, Anil was involved in every step from design to the final publication. He was the main developer of the tools presented and the main author of the papers.

1. A.S. Thanki, N. Soranzo, W. Haerty, and R.P. Davey. GeneSeqToFamily: a Galaxy workflow to find gene families based on the Ensembl Compara GeneTrees pipeline. *GigaScience*, 7(3):giy005, 2018.
2. A.S. Thanki, N. Soranzo, J. Herrero, W. Haerty, and R.P. Davey. Aequatus: an open-source homology browser. *GigaScience*, 7(11):giy128, 2018.

I fully endorse Anil's submission to obtain a PhD by publication.

Please don't hesitate to contact me for more information.

Yours faithfully,



Nicola Soranzo



London, 29<sup>th</sup> of April 2019

To whom it concerns

**Re: Aequatus: an open-source homology browser. Anil S Thanki, Nicola Soranzo, Javier Herrero, Wilfried Haerty, Robert P Davey. *GigaScience*, Nov 2018, 7(11):giy128, 10.1093/gigascience/giy128.**

Anil Thanki was the main developer of the tool presented in this paper. My involvement in the work was mainly to discuss the progress and new ideas with him and Dr. Robert Davey but Mr Thanki was certainly leading the development of the tool, which is reflected as him being the lead author in this article.

It is worth mentioning that his contribution was beyond purely technical. It is thanks to the combination of his understanding of the biological questions, his programming abilities and knowledge of the Ensembl API that this project has been a success.

Since the publication is so recent, there has not been enough time for other articles to cite this work, however the peer reviews are available on the Journal web page. I would like to highlight a couple of sentences from these:

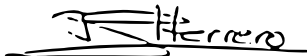
“The key innovation, though, lies in a comparative display of gene structure, with an annotated gene tree on the left and a display of matching exons on the right.” *Dr. Christophe Dessimoz*

“This software will be a useful tool for the visualization of detailed syntenic relationships at the sub-gene level and at the localized gene level”. *Dr Deborah Weighill*

Clearly both reviewer value the novelty of Aequatus. Surely, there are many different use cases for the detailed comparison of exon structure among orthologous genes, including gene annotation QC, evolutionary studies on birth/death of exons, etc.

In summary, this article shows the capacity of Mr Thanki and fits well with the rest of his work presented in his thesis ‘Development of computational techniques for genomic data analysis and visualisation in the model and non-model organisms’. I fully support Mr Thanki’s submission to obtain a PhD by publication.

Sincerely yours



Prof. Javier Herrero  
Head of the Bill Lyons Informatics Centre  
UCL Cancer Institute  
University College London



*To Whom It May Concern:*

Dear Colleague,

Re: Modha, S., Thanki, A. S., Cotmore, S. F., Davison, A. J., & Hughes, J. (2018). ViCTree: an automated framework for taxonomic classification from protein sequences. *Bioinformatics*, 34(13), 2195–2200. <https://doi.org/10.1093/bioinformatics/bty099>

Anil Thanki was the main developer of the ViCTreeView tool included in this paper and integrated in the ViCTree analysis pipeline. I have developed the ViCTree analysis backend pipeline with help from other collaborators and Mr Anil Thanki led and executed the ViCTreeView visualisation component successfully.

It is notable that his contribution was a key part of the project. Mr Thanki has demonstrated his expertise throughout the development and publication process. He has a very strong background in the scientific programming and his contribution in this project reaffirms his proficiency in this area of research.

It is my pleasure to support Anil Thanki in his bid to gain a PhD by publication.

Yours sincerely,

Sejal Modha  
Post-graduate Research Student  
telephone: +44 (0)141 330 2886  
e-mail: [s.modha.1@research.gla.ac.uk](mailto:s.modha.1@research.gla.ac.uk)

**Medical Research Council-University of Glasgow Centre for Virus Research**  
Sir Michael Stoker Building, Garscube Campus, 464 Bearsden Road, Glasgow G61 1QH  
[cvr.ac.uk](http://cvr.ac.uk) @CVRInfo

The University of Glasgow, charity number SC004401.



Joseph Hughes  
MRC – University of Glasgow Centre for Virus Research  
Garscube Estate, Bearsden Road  
Glasgow  
G61 1QH  
UK

26 March 2019

Dear Colleague,

Re: Modha S, **Thanki AS**, Cotmore SF, Davison AJ, Hughes J. ViCTree: an automated framework for taxonomic classification from protein sequences. *Bioinformatics*. 2018 Jul 1;34(13):2195-2200. doi: 10.1093/bioinformatics/bty099.

Anil Thanki was critical in the successful publication of the work listed above. Anil developed the complete visualization front-end for our pipeline that automates the generation of virus family phylogenies. He worked closely with us as we developed the pipeline and was always responsive to developing feature enhancements for the visualization, both prior to publication and following the reviewer's comments.

I am very happy to support Anil in his bid to gain a PhD by publication.

Yours sincerely,

Joseph Hughes

Senior Bioinformatician

telephone: +44 (0)141 330 4019

e-mail: joseph.hughes@glasgow.ac.uk

**Medical Research Council-University of Glasgow Centre for Virus Research**  
Sir Michael Stoker Building, Garscube Campus, 464 Bearsden Road, Glasgow G61 1QH  
cvr.ac.uk @CVRInfo

The University of Glasgow, charity number SC004401.



New Road, East Malling  
Kent ME19 6BJ

T. +44 (0)1732 843833  
enquiries@emr.ac.uk  
@emrcomms  
www.emr.ac.uk

20 June 2019

To Whom It May Concern

**Re: Anil Thanki – PhD candidate**

I am writing this letter in support of Mr Anil Thanki as a candidate for a PhD degree at the University of East Anglia. I was responsible of the Bioinformatics Department at The Genome Analysis Centre (TGAC, now Earlham Institute) when we recruited Anil. Since then I had the opportunity to work with Anil on a number of projects, initially from my role as Head of Bioinformatics at TGAC and later as Director of the organization until my departure in August 2015. This work supported a number of manuscripts as detailed below.

A. S. Thanki, N. Soranzo, J. Herrero, W. Haerty, and R. P. Davey, Aequatus: an open-source homology browser. Gigascience 2018 - Selected in GigaScience Prize Track

A. S. Thanki, X Bian, R. P. Davey, TGAC Browser: An open-source genome browser for non-model organisms. bioRxiv 2019

M. Spannagl, M. Alaux, M. Lange, D. M. Bolser, [and 24 others, including M. Caccamo and A. S. Thanki], transPLANT Resources for Triticeae Genomic Data. The Plant Genome 2015

I have always been impressed with Anil's commitment to scientific excellence and by his expertise in computational methods which add to his strength as a scientist.

From these achievements I believe Anil is an exceptionally strong candidate for a PhD and therefore I am very happy to write this letter of support.

Yours faithfully

A handwritten signature in black ink, appearing to read "Mario Caccamo".

Professor Mario Caccamo  
Managing Director  
NIAB EMR



## Appendix II: Publications submitted

---

1. **A. S. Thanki**, X Bian, and R. P. Davey, “TGAC Browser: An open-source genome browser for non-model organisms,” bioRxiv 2019
2. **A. S. Thanki**, R. C Jimenez, G. G. Kaithakottil, M. Corpas, and R. P. Davey “wigExplorer, a BioJS component to visualise wig data,” F1000Research 2014
3. **A. S. Thanki**, N. Soranzo, W. Haerty, and R. P. Davey, “GeneSeqToFamily: a Galaxy workflow to find gene families based on the Ensembl Compara GeneTrees pipeline,” GigaScience 2018
4. **A. S. Thanki**, N. Soranzo, J. Herrero, W. Haerty, and R. P. Davey, “Aequatus: an open-source homology browser,” GigaScience 2018 - Selected in GigaScience Prize Track
5. S. Modha, **A. S. Thanki**, S. F. Cotmore, A. J. Davison, and J. Hughes, “Victree: an automated framework for taxonomic classification from protein sequences,” Bioinformatics 2018
6. M. Spannagl, M. Alaux, M. Lange, D. M. Bolser, [and 25 others, including **A. S. Thanki**.] “transPLANT Resources for Triticeae Genomic Data,” The Plant Genome 2015

**TGAC Browser: An open-source  
genome browser for non-model  
organisms**

**A. S. Thanki, X Bian, and R. P. Davey,  
bioRxiv 2019**

---

# TGAC Browser: An open-source genome browser for non-model organisms

Anil S. Thanki<sup>1</sup>, Xingdong Bian<sup>1</sup>, and Robert P. Davey<sup>1</sup>

<sup>1</sup>Earlham Institute, Norwich Research Park, NR4 7UZ United Kingdom

Genome browsers play a vital role to provide visualisation for genomic data. It is often the case that bespoke genome browser customisations are required between different research groups, with an obvious necessity to update, upgrade and tailor tracks and features on a potentially frequent basis. However, most of the current genome browsers require highly curated data held in public repositories. Besides, these genome browsers often rely on particular dependencies, where writing plug-in or modifying existing code can be troublesome and resource expensive.

We present TGAC Browser, a new open-source web-based genome browser designed to overcome shortcomings in available approaches. It uses a locally installed Ensembl Core Database schema and is also able to visualise data from well-known NGS data formats. We also added simple analysis functionality to perform BLAST searches within TGAC Browser. TGAC Browser also allows uploading your genomic data. TGAC Browser is an open-source, easy to set up, and user-friendly genome browser with minimal, lightweight configuration details.

Genome Browser, Genomics, Ensembl, NGS data, Visualisation

Correspondence: Anil.Thanki@earlham.ac.uk and Robert.Davey@earlham.ac.uk

## Introduction

Genome browsers (1–3) typically present spatial relationships between different pieces of biological information by providing graphical visualisations of the genomic data. Despite advances in data production and analysis methods, genome browsers play an important role in examining data to explore the results of new analysis and generating hypotheses (4). The principal function of the genome browser is to aggregate different types of genomic annotation data together

and integrate them into an abstract graphical view (5). It allows researchers to visualise and explore predicted genes, transcripts, gene expression, variation, comparative analysis, and alignments. Because of so many of these reasons to use genome browser, many software has been developed which are widely used and essential, for example, Ensembl genome browser (3), GBrowse (1), JBrowse (6), and IGV (7).

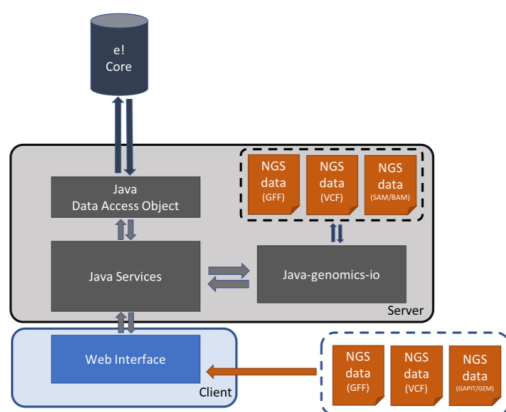
In general, genome browsers can be divided into two categories: standalone browsers and web-based browsers. Standalone browsers are used on a local computer, which tends to focus on heavyweight applications to run with a large dataset. While web-based browsers are generally installed on host institutional server and can be used over the internet. Here we are focusing on a web-based genome browser, which is more popular due to its flexible accessibility and performance.

Many of the available web-based genome browsers require heavily curated data in public repositories as well as in a specific format supported by the particular browser. With the democratisation of sequencing technologies, smaller research labs are generating an increasing amount of sequencing data and performing analyses, especially for non-model organisms. Biological analyses can be performed in many alternative ways providing results in various formats; thus it can be a tedious process to convert data in order for it to be supported by a particular browser and data needs to be curated before making them available from a public repository. Among various available genomic formats, the Ensembl database system (8) is standard format containing various genomic annotation, and it is widely accepted in both companies as well as academic sites and also provides a framework to load any standard NGS formatted data into Ensembl databases.

To better simplify genomic data visualisation, we present the TGAC Browser, an open-source genome browser. It retrieves and visualises data directly from a local instance of the Ensembl core database (8) as well as well-known NGS data formats. TGAC Browser can also perform BLAST (Basic Local Alignment Search Tool) (9) analysis within.

## Materials and Methods

TGAC Browser is designed with a typical server-client architecture (see Figure 1) to utilise the server for data retrieval and use clients' computational resources to generate the visualisation. This approach provides a consistent experience to users when the TGAC Browser is being used by multiple users simultaneously. Client and server transfer data asynchronously using Ajax (Asynchronous JavaScript And XML).



**Fig. 1.** The TGAC Browser infrastructure, showing the interactions between the server-side implementation, connected to Ensembl core database using Java Data Access Objects and NGS files via Java-Genomics-IO, and the client-side implemented using popular techniques such as JavaScript, jQuery, d3.js and jQuery DataTables.

The server side of TGAC Browser is implemented in Java programming language, which retrieves data from a local Ensembl Core database using Java DAO (Data Access Objects) and NGS formatted files using Java-Genomics-IO library (10), a Java library developed by Timothy Palpant. TGAC Browser retrieves references from Ensembl database and visualises genomic annotation from the Ensembl

database as well as NGS formatted files such as SAM (Sequence Alignment/Map format) (11), BAM (Binary equivalent of SAM), GFF (Generic Feature Format) (12), and VCF (Variant Call Format) (13).

TGAC Browser client-side is implemented in JavaScript, jQuery library, SVG (Scalable Vector Graphics) and D3.js (Data-Driven Documents) (14). By using all these well-known web technologies, we are able to create seamless browsing experience for users, where user can drag, pan and rearrange genomics tracks on a web browser similar to Google Maps. We have implemented lazy loading method, in which TGAC Browser retrieves and visualise data only for the visible and surrounding regions and for any action by the user it retrieves only required additional data. This strategy allows for very dynamic zooming and scrolling to provide smoother and faster users experience.

TGAC Browser allows users to upload (Figure 3 E) genomic annotations such as GFF, GAPIT (Genome Association and Prediction Integrated Tool) (15) and GEM file format containing information about Genes, SNPs and expression data. This provides a collaborative platform for users to visualise their data safely without needing to share or making it available from the server.

TGAC Browser has an integrated BLAST search functionality to add analysis capability. BLAST can be set up to run on local installation or High-performance computing (HPC) cluster using BLAST+ (16), as well as NCBI BLAST server. TGAC Browser keeps track of BLAST analysis using *blast\_manager*, a database system (see Figure 2), and stores result for future reference.

## Results

The layout of the TGAC Browser (see Figure 3) is similar to the many of the genome browser available, making it user-friendly. In this layout, the genomic region spans from left to right and genomic annotations are laid out from top to bottom.

TGAC Browser visualises reference level genomic informa-



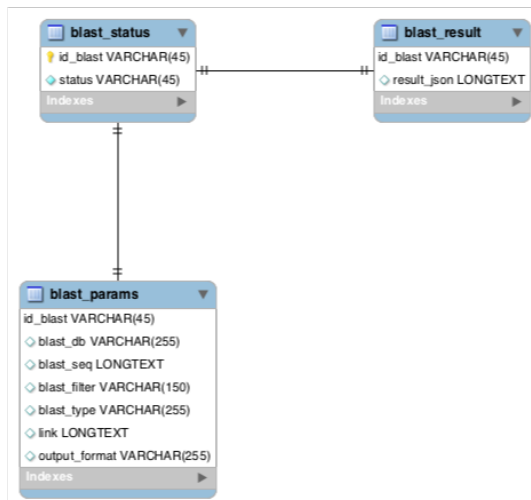


Fig. 2. blast\_manager database schema



Fig. 3. The main view of TGAC Browser. The header on top provides a search box (A) and a link to BLAST Search (B). It is followed by the second panel containing Control bar (J), an option to toggle tracks (C), save session (D) and upload tracks (E). Chromosomal view (F) represents available chromosomes for the species, where the selected chromosome is coloured in red. Below there is a horizontal view of reference (G), followed by a zoomed area of the reference (H). All genomic tracks (I) are laid out in order after that. The figure is also showing an example of a typical popup (K)

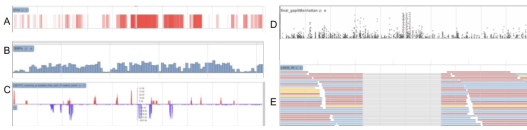
tion on top using chromosome information (Figure 3 F), if available. as well as horizontal selectable region (Figure 3 G). User can move selector on the selectable region, and the respective region will be shown below (Figure 3 H), which can be visualised as nucleotide sequences and three forward frame translation if zoomed enough. The chromosomal view gives user overview of the reference species as well as provide a visual guide of the current viewing region on the chromosome and let user change region of interest. Chromosomal view also visualises available genomic markers on the side.

**Interface features.** TGAC Browser has implemented various browsing functionalities to provide a seamless browsing experience. Navigation controls are in the top control bar (Figure 3 J) for panning and zooming. It also contains an expand button for an overview of the whole reference and reset button to focus on the centre point of reference. In addition, TGAC Browser also equipped with google maps style panning by dragging the mouse and zooming with a scroll or double click as well as panning with arrow keys on the keyboard.

Genomic annotations can be ordered by dragging them with a label of the track and toggled from Tracks/Settings (Figure 3 C). Primary information for each annotation is visualised next to the genomic track, and additional information can be seen with mouseover, as well as in a popup (detailed below).

**Search.** TGAC Browser is equipped with flexible keyword-based search functionality (Figure 3 A), which searches against chromosome names, assembly information as well as all the relevant genomic features information such as gene symbols, Ensembl stable IDs (unique identifiers in the Ensembl project for each genomic annotation), common names in the database. It visualises results along with Chromosomal view if available or in tabular form with a link to respective browser view.

**Visualisations.** TGAC Browser presents genomic annotations using various types of visualisations automatically chosen by the type and volume of genomic data to be visualised (see Figure 4). For small dataset each annotation visualises independently while large dataset visualises either as a histogram (Figure 4 B) or a heat map (Figure 4 A) representing quantitative information. This method is memory efficient as well as helps the user to look at a glance for a larger region and then focus on a particular segment for a detailed view. TGAC Browser also uses wiggle plots (Figure 4 C) for expression data using wigExplorer (17) from BioJS (18) and Manhattan style (Figure 4 D) visuals for SNPs.



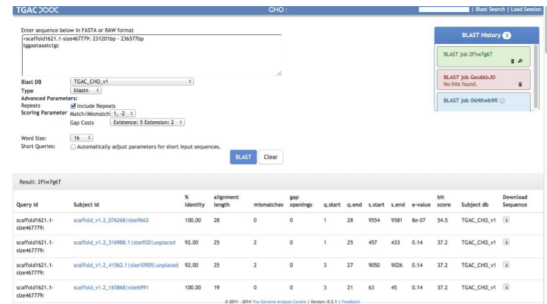
**Fig. 4.** TGAC Browser visualises genomic annotation based on type and amount of data: A: Heat Map presentation of large data (more than 5000 annotation), B: Graphical presentation of large data (from 1000 to 5000 annotation), C: Wiggle plot for expression data using wigExplorer, D: Manhattan plot for GAPIT data, E: Visualising reads directly from SAM/BAM file

**Pop-up.** TGAC Browser provides a contextual menu system via interactive pop-ups (Figure 3 K), which contains additional information for genomic annotation such as analysis type, position on the reference and textual description. Pop-up also contains options to fetch sequence, perform BLAST analysis for the sequence of selected annotation, focus on the annotation, highlighting annotation as well as provides a link to the Ensembl for more information (if the annotation is available in Ensembl). All this information and options are dynamically selected based on the type of annotation.

**BLAST.** Integration of BLAST search within TGAC Browser plays a key role by providing the ability to perform analysis within. A user can utilise BLAST in two ways:

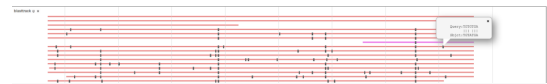
First, the user can perform BLAST search on a sequence of interest and results visualised in tabular view with links to specific result (Figure 5). User can perform multiple BLAST searches, and all the search results are shown as a selectable list, where user can toggle between results (Figure 5). In here user can also choose the type of BLAST (i.e. blastn, tblastn and blastx) as well as change parameters (e.g. scoring parameters, gap penalties and word-size). This feature gives TGAC Browser facility to search with a sequence in addition to the traditional keyword-based search.

Second, the user can perform BLAST search from the pop-up menu of the selected genomic feature and results are presented as a genomic track alongside others (Figure 6). These BLAST Results are coloured based on standard BLAST colour schema for bit-score; it also represents insertion and deletion information. This feature helps to find out other matching regions from references for the selected annotation



**Fig. 5.** BLAST analyses are showing results with links out to associated TGAC Browser instance. on the top right showing BLAST run, allowing previous results to be shown and removed.

in the case of multiple copies of a chromosome.



**Fig. 6.** BLAST analyses are showing results as genomic track. On top right pop-up showing insertions and deletions at the position, BLAST hits are coloured based on the score.

**Sharing.** TGAC Browser allows users to share information using URL as well as session id:

**Persistent URL.** TGAC Browser provides persistent unique Uniform Resource Locator (URL) to enable consistent access to the point of interest. Users can share the link for the specific reference to a given species and chromosome, or a search term. This makes it easy to share information with collaborators, or for use in publications.

**Session.** TGAC Browser also allows the user to share information using session feature (Figure 3 D), where the user can save a running session and share it with collaborator for the same view.

## Conclusions

As more and more genomes are being sequenced, genome browsers are increasing importance. Thus, we developed the TGAC Browser, a genome browser that relies on non-proprietary software but only readily available Ensembl Core database and NGS data formats. The capability of TGAC

Browser to visualise data from multiple sources without any conversion makes it ideally suited to be used with newly sequenced next-generation sequencing datasets of the model and non-model organisms.

TGAC Browser follows many optimisations for data visualisations, making it versatile and robust genome browser. Functionalities to browse, upload, and share genomic information, make it all-rounder genome browser. In addition to exploration tool, TGAC Browser is also able to help scientists pursuing their research by performing analysis using built-in BLAST functionality.

TGAC Browser has been actively used by Primula Research Group at Earlham Institute and the University of Hull, SZN (Stazione Zoologica Anton Dohrn) Napoli, Brassica RIPR community, transPLANT (19), and Vietnamese Rice Community.

The ultimate goal of TGAC Browser is to provide a unique and a single solution to represent genomic data, from known NGS data format(s) for model and non-model organisms.

## Future Directions

We are looking to incorporate TGAC Browser into the virtualization system generated using CyVerse (20) and Docker with Galaxy (21) to provide a complete solution for genome analysis and exploration, where genomic annotation generated from Galaxy can directly be available to visualised using TGAC Browser instance.

We would also like to investigate into implementing, user-friendly annotation method, for users to add or modify genomic annotation. It would help to bring the community together for new genomic annotation as well as validation and curation of existing annotation.

## Availability

Information about the TGAC Browser and a demo instance are currently available at the URL below, and source code for TGAC browser is also available on GitHub. We would be pleased to help any potential users interested in the project.

Demo: <http://browser.earlham.ac.uk>

Source-code: <https://github.com/TGAC/TGACBrowser>

## ACKNOWLEDGEMENTS

This work was supported in part by the NBI Computing Infrastructure for Science Group, which provides technical support and maintenance to EI's high-performance computing cluster and storage systems, which enabled us to develop this tool. We thank the attendees of the 2012 CHO consortium Workshop at Earlham Institute and 2012 NGS meeting at Nottingham, as well as Brassica RIPR community for helpful feedback about TGAC Browser.

## Bibliography

- Lincoln D Stein, Christopher Mungall, Shengqiang Shu, Michael Caudy, Marco Mangone, Allen Day, Elizabeth Nickerson, Jason E Stajich, Todd W Harris, Adrian Arva, and Suzanna Lewis. The generic genome browser: a building block for a model organism system database. *Genome Res.*, 12(10):1599–1610, October 2002.
- W J Kent. The human genome browser at UCSC. *Genome Res.*, 12(6):996–1006, 2002.
- James Stalker, Brian Gibbins, Patrick Meidl, James Smith, William Spooner, Hans-Rudolf Hotz, and Antony V Cox. The ensembl web site: mechanics of a genome browser. *Genome Res.*, 14(5):951–955, May 2004.
- Thomas A Down, Matias Pilpari, and Tim J P Hubbard. Dalliance: interactive genome viewing on the web. *Bioinformatics*, 27(6):889–890, March 2011.
- Jun Wang, Lei Kong, Ge Gao, and Jingchu Luo. A brief introduction to web-based genome browsers. *Brief. Bioinform.*, 14(2):131–143, March 2013.
- Robert Buels, Eric Yao, Colin M Diesh, Richard D Hayes, Monica Munoz-Torres, Gregg Helt, David M Goodstein, Christine G Elsik, Suzanna E Lewis, Lincoln Stein, and Ian H Holmes. JBrowse: a dynamic web platform for genome visualization and analysis. *Genome Biol.*, 17(1):66, April 2016.
- Helga Thorvaldsdóttir, James T Robinson, and Jill P Mesirov. Integrative genomics viewer (IGV): high-performance genomics data visualization and exploration. *Brief. Bioinform.*, 14(2):178–192, March 2013.
- T Hubbard. The ensembl genome database project. *Nucleic Acids Res.*, 30(1):38–41, 2002.
- S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, October 1990.
- timpalpant. timpalpant/java-genomics-io. <https://github.com/timpalpant/java-genomics-io>. Accessed: 2016-5-17.
- Sequence Alignment/Map format specification. <https://samtools.github.io/hts-specs/SAMv1.pdf>. Accessed: 2016-5-18.
- GFF3 - GMOD. <http://gmod.org/wiki/GFF3>. Accessed: 2016-5-18.
- Petr Danecek, Adam Auton, Goncalo Abecasis, Cornelis A Albers, Eric Banks, Mark A DePristo, Robert E Handsaker, Gerton Lunter, Gabor T Marth, Stephen T Sherry, Gilean McVean, Richard Durbin, and 1000 Genomes Project Analysis Group. The variant call format and VCFtools. *Bioinformatics*, 27(15):2156–2158, August 2011.
- Mike Bostock. D3.js - Data-Driven documents. [D3.js - Data-Driven documents](https://d3js.org). [D3.js - Data-Driven Documents \[Internet\]. \[cited 2015 Dec 21\]](https://d3js.org). Available from: <http://d3js.org>. Accessed: 2016-5-18.
- Alexander E Lipka, Feng Tian, Qishan Wang, Jason Peiffer, Meng Li, Peter J Bradbury, Michael A Gore, Edward S Buckler, and Zhiwu Zhang. GAPIT: genome association and prediction integrated tool. *Bioinformatics*, 28(18):2397–2399, September 2012.
- Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. BLAST+: architecture and applications. *BMC Bioinformatics*, 10:421, December 2009.
- Anil S Thanki, Rafael C Jimenez, Gemy G Kaitihakottill, Manuel Corpas, and Robert P Davey. wigexplorer, a BioJS component to visualise wig data. *F1000Res.*, 2014.

18. John Gómez, Leyla J García, Gustavo A Salazar, Jose Villaveces, Swanand Gore, Alexander García, Maria J Martín, Guillaume Launay, Rafael Alcántara, Noemi Del-Toro, Marine Dumousseau, Sandra Orchard, Sameer Velankar, Henning Herbjakob, Chenggong Zong, Peipei Ping, Manuel Corpas, and Rafael C Jiménez. BioJS: an open source JavaScript framework for biological data visualization. *Bioinformatics*, 29(6):1103–1104, April 2013.
19. Manuel Spannagl, Michael Alaux, Matthias Lange, Daniel M Bolser, Kai C Bader, Thomas Letellier, Erik Kimmel, Raphael Flores, Cyril Pommier, Arnaud Kerhornou, Brandon Walts, Thomas Nussbaumer, Christoph Grabmuller, Jinbo Chen, Christian Colmsee, Sebastian Beier, Martin Mascher, Thomas Schmutzer, Daniel Arend, Anil Thanki, Ricardo Ramirez-Gonzalez, Martin Ayling, Sarah Ayling, Mario Caccamo, Klaus F X Mayer, Uwe Scholz, Delphine Steinbach, Hadi Quesneville, and Paul J Kersey. transPLANT resources for triticeae genomic data. *Plant Genome*, 9(1), March 2016.
20. Stephen A Goff, Matthew Vaughn, Sheldon McKay, Eric Lyons, Ann E Stapleton, Damian Gessler, Naim Matasci, Liya Wang, Matthew Hanlon, Andrew Lenards, Andy Muir, Nirav Merchant, Sonya Lowry, Stephen Mock, Matthew Helmke, Adam Kubach, Martha Narro, Nicole Hopkins, David Micklos, Uwe Hilgert, Michael Gonzales, Chris Jordan, Edwin Skidmore, Rion Dooley, John Cazes, Robert McLay, Zhenyuan Lu, Shiran Pasternak, Lars Koesterke, William H Piel, Ruth Grene, Christos Noutsos, Karla Gendler, Xin Feng, Chun-liao Tang, Monica Lent, Seung-Jin Kim, Kristian Kvilekval, B S Manjunath, Val Tannen, Alexandros Stamatakis, Michael Sanderson, Stephen M Welch, Karen A Cranston, Pamela Soltis, Doug Soltis, Brian O'Meara, Cecile Ane, Tom Brutnell, Daniel J Kleibenstein, Jeffrey W White, James Leebens-Mack, Michael J Donoghue, Edgar P Spalding, Todd J Vision, Christopher R Myers, David Lowenthal, Brian J Enquist, Brad Boyle, Ali Akoglu, Greg Andrews, Sudha Ram, Doreen Ware, Lincoln Stein, and Dan Stanzione. The iplant collaborative: Cyberinfrastructure for plant biology. *Front. Plant Sci.*, 2:34, July 2011.
21. Enis Afgan, Dannon Baker, Marius van den Beek, Daniel Blankenberg, Dave Bouvier, Martin Čech, John Chilton, Dave Clements, Nate Coraor, Carl Eberhard, Björn Grüning, Aysam Guerler, Jennifer Hillman-Jackson, Greg Von Kuster, Eric Rasche, Nicola Soranzo, Nitesh Turaga, James Taylor, Anton Nekrutenko, and Jeremy Goecks. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res.*, 44(W1):W3–W10, July 2016.

wigExplorer, a BioJS component to  
visualise wig data

A. S. Thanki, R. C Jimenez, G. G.  
Kaithakottil, M. Corpas, and R. P.  
Davey

F1000Research 2014

---



SOFTWARE TOOL ARTICLE

**REVISED** *wigExplorer*, a BioJS component to visualise wig data  
[version 3; peer review: 1 approved, 2 approved with reservations, 1 not approved]

Anil S. Thanki<sup>1</sup>, Rafael C. Jimenez<sup>2</sup>, Gemy G. Kaithakottil<sup>1</sup>, Manuel Corpas <sup>1</sup>, Robert P. Davey <sup>1</sup>

<sup>1</sup>Earlham Institute, Norwich Research Park, Norwich, NR4 7UH, UK

<sup>2</sup>European Bioinformatics Institute, Hinxton, Cambridge, CB10 1SD, UK

**V3** **First published:** 13 Feb 2014, 3:53 (<https://doi.org/10.12688/f1000research.3-53.v1>)  
**Second version:** 30 May 2014, 3:53 (<https://doi.org/10.12688/f1000research.3-53.v2>)  
**Latest published:** 09 Aug 2016, 3:53 (<https://doi.org/10.12688/f1000research.3-53.v3>)

**Abstract**

**Summary:** *wigExplorer* is a BioJS component whose main purpose is to provide a platform for visualisation of wig-formatted data. Wig files are extensively used by genome browsers such as the UCSC Genome Browser. *wigExplorer* follows the BioJS standard specification, requiring a simple configuration and installation. *wigExplorer* provides an easy way to navigate the visible region of the canvas and allows interaction with other components via predefined events.

**Availability:** <http://biojs.io/d/biojs-vis-wigexplorer>;  
<http://dx.doi.org/10.5281/zenodo.8516>

**Keywords**

BioJS, data visualisation, genome browsers



This article is included in the **International Society for Computational Biology Community Journal gateway**.



This article is included in the **EMBL-EBI gateway**.



This article is included in the **BioJS collection**.

**Open Peer Review**

**Reviewer Status** X ✓ ? ?

	1	2	3	4
<b>version 3</b> published 09 Aug 2016				 report
<b>version 2</b> published 30 May 2014		✓ report	?	?
<b>version 1</b> published 13 Feb 2014	X report		?	

- Robert Buels**, University of California, Berkeley, CA, USA
- Phil Lord**, University of Newcastle, Newcastle, UK
- Stefan Thomas Lang**, Lund University, Lund, Sweden
- Chunlei Wu** , Scripps Research Institute, La Jolla, USA  
**Jiwen Xin**, The Scripps Research Institute, La Jolla, USA

Any reports and responses or comments on the article can be found at the end of the article.

**Corresponding author:** Anil S. Thanki ([Anil.Thanki@earham.ac.uk](mailto:Anil.Thanki@earham.ac.uk))

**Competing interests:** No competing interests were disclosed.

**Grant information:** AT, GK and RD were supported by the UK Biotechnology and Biological Sciences Research Council (BBSRC) National Capability Grant (BB/J010375/1) at Earham Institute.

*The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.*

**Copyright:** © 2016 Thanki AS *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**How to cite this article:** Thanki AS, Jimenez RC, Kaithakottil GG *et al.* *wigExplorer, a BioJS component to visualise wig data [version 3; peer review: 1 approved, 2 approved with reservations, 1 not approved]* F1000Research 2016, 3:53 (<https://doi.org/10.12688/f1000research.3-53.v3>)

**First published:** 13 Feb 2014, 3:53 (<https://doi.org/10.12688/f1000research.3-53.v1>)

**REVISED** Amendments from Version 2

In the new version we have updated the availability sources, and some reference details based on the referees' comments.

See referee reports

## Introduction

Numerous web applications exist for visualisation of biological data. Data can be prepared for visualisation using a variety of formats, one of which is the widely used wiggle (wig) file. A wiggle file contains text that defines either a feature or a data track. The wiggle format was developed by the UCSC genome browser<sup>1</sup> and then quickly adopted by other initiatives<sup>2,3</sup>. Web applications such as genome browsers rely heavily on JavaScript, a popular language for processing and rendering client-side information in a web browser. Despite their widespread use in bioinformatics, biological web applications are usually implemented with no standard reutilisation guidelines in mind, hence BioJS was developed<sup>4</sup>. BioJS code contains proper guidelines on how to use the components and how the API can be implemented to interact with other components.

BioJS is an open source JavaScript library of components for the visualisation of biological data on the web. Here we present *wigExplorer*, a standard, portable BioJS component designed to easily render wig data format files. *wigExplorer* can be integrated and controlled from other applications. To our knowledge, this is the first modular component to visualise wig data that complies with BioJS standards.

## The *wigExplorer* component

*wigExplorer* is fully integrated in the BioJS project. It follows the standards set by the BioJS registry<sup>5</sup>, a centralised repository of BioJS components hosted at the European Bioinformatics Institute (EBI). Having *wigExplorer* in the BioJS registry is advantageous because it promotes i) easy discoverability for the component, ii) collaborative development with other members of the BioJS community and iii) reutilisation by third party applications. In the BioJS registry, component APIs are exposed, i.e., events and methods are defined and documented so that other BioJS components can interact with each other. By following these conventions, *wigExplorer* is able to interact with other components on the same web page, enriching the overall experience for the user. The code below shows how to incorporate *wigExplorer* into a web application. Only three configuration elements are needed: the target HTML element in which the component will be rendered, the background colour of the component, and the file path containing the wig data. Wig files contain minimalistic information of genomic data *wigExplorer* and can handle a large genomic region such as a chromosome (tested

with a single file containing 12 chromosome with average length of 60 Mb), but this depends on the richness of the data rather than the length of the genomic region.

```
var instance = new Biojs.wigExplorer({
  target: "YourOwnDivId",
  selectionBackgroundColor: '<background-colour>',
  dataSet: "<path-to-file>"
});
```

*wigExplorer* uses D3.js, the data-driven documents JavaScript library<sup>6</sup>, to generate graphical representations from wig data. D3.js handles the manipulation of the data documents, the reading of wig data as text format and their conversion to an area chart format. On the top right side, *wigExplorer* contains a dropdown to toggle between different references from the wig file. To control the visual aspect of the wig data, *wigExplorer* contains simple controls for zooming and panning. It is also possible to zoom and pan using provided API.

```
instance._updateDraw(start, end)
```

## Application

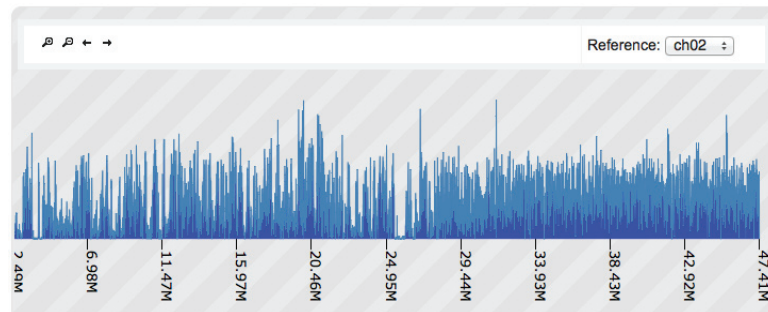
*wigExplorer* can be used to visualise genomic data in different ways. An application is shown in Figure 1, depicting single nucleotide polymorphism (SNP) density data from a genomic annotation in the tomato genome. Here chromosome 2 is zoomed in to show the genomic interval contained between position 2.5M and 47.5M. The SNP density data contained in the wig data file are presented as bins, where the Y axis indicates the number of SNPs contained in each bin. The screenshot shows a dramatic change in the density of SNPs just after the 24M bin mark of the chromosome, suggesting a potential boundary for an introgression segment introduced from a closely related tomato species. Other potential applications of *wigExplorer* may involve the visualisation of gene expression and alignment data.

Third party browsers are also using *wigExplorer*. A screenshot of the TGAC Browser<sup>7</sup> is shown in Figure 2 using *wigExplorer* to depict *Myzus* spp. scaffold 1 zoomed in between regions 714K and 727K. Here strand-specific RNA-Seq paired-end read coverage is shown as a wig track. The track below shows a closely related annotated species gene set for comparison. This comparison suggests a potential gene extension in both forward and reverse orientation.

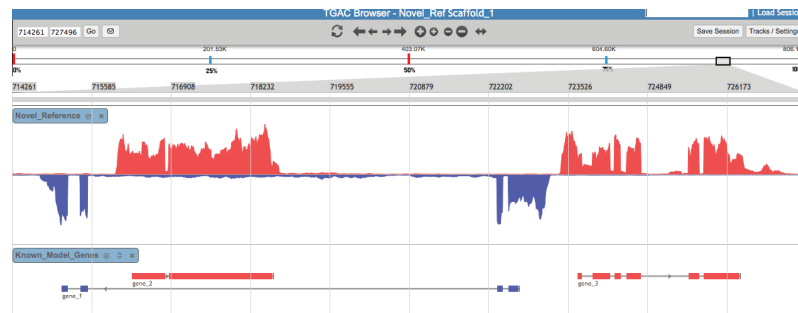
## Conclusions

The *wigExplorer* component provides a platform to visualise biological data in wig format. *wigExplorer* can be easily integrated with other web components or extended to provide new functionality. We expect this component to be particularly useful for visualisation in a variety of data types such as SNP density, alignments and gene





**Figure 1.** *wigExplorer* view of tomato variety Heinz chromosome 2. The top controls are designed to toggle between different references as well as zoom and pan. Peaks show SNP density of 1kb size bins. A change of SNP density can be observed around the 24M mark, with a slightly greater density of SNPs on the right, indicative of a potential introgression segment from another related species.



**Figure 2.** An example of *wigExplorer* integration using the TGAC Browser. The *wigExplorer* track shows read coverage in *Myzus* spp. for scaffold 1. Forward and backward strands are depicted in red and blue respectively. Evidence genes from a closely related species are displayed in the track below.

expression. Like any other BioJS component, *wigExplorer* requires little technical knowledge for its utilisation.

#### Software availability

Zenodo: *wigExplorer*, a BioJS component to visualise wig data\_v2, doi: [10.5281/zenodo.8516](https://doi.org/10.5281/zenodo.8516)<sup>8</sup>

GitHub: BioJS, <http://github.com/biojs/biojs/releases/tag/v1.0>;

#### Author contributions

AT and RJ developed the code for *wigExplorer*. MC and GK created the user cases for [Figure 1](#) and [Figure 2](#) respectively. AT, RD, MC and GK wrote the paper.

#### Competing interests

No competing interests were disclosed.

#### Grant information

AT, GK and RD were supported by the UK Biotechnology and Biological Sciences Research Council (BBSRC) National Capability Grant (BB/J010375/1) at Earlham Institute.

*The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.*

#### Acknowledgements

We are grateful to all BioJS developers who have contributed their work under an open source license. We are thankful to David Swarbreck at Earlham Institute for his advice on the data shown in [Figure 2](#).

## References

---

1. Kent WJ, Sugnet CW, Furey TS, *et al.*: **The human genome browser at UCSC.** *Genome Res.* 2002; **12**(6): 996–1006.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
2. Donlin MJ: **Using the Generic Genome Browser (GBrowse).** *Curr Protoc Bioinformatics.* John Wiley and Sons, Inc., 2009.  
[PubMed Abstract](#) | [Publisher Full Text](#)
3. Skinner ME, Uzilov AV, Stein LD, *et al.*: **JBrowse: A next-generation genome browser.** *Genome Res.* 2009; **19**(9): 1630–1638.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
4. Gómez J, García LJ, Salazar GA, *et al.*: **BioJS: an open source JavaScript framework for biological data visualization.** *Bioinformatics.* 2013; **29**(8): 1103–1104.  
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
5. **BioJS: registry.** <http://biojs.io>, 2016.  
[Reference Source](#)
6. **D3.js data-driven documents.** <http://d3js.org>, 2012.  
[Reference Source](#)
7. Thanki AS, Bian X, Davey RP, *et al.*: **TGAC Browser: visualisation solutions for big data in the genomic era.** 2016.  
[Reference Source](#)
8. Thanki AS, Jimenez RC, Kaithakottil GK, *et al.*: **wigexplorer, a biojs component to visualise wig data\_v2.** *Zenodo.* 2014.  
[Data Source](#)

**GeneSeqToFamily: a Galaxy workflow  
to find gene families based on the  
Ensembl Compara GeneTrees pipeline  
A. S. Thanki, N. Soranzo, W. Haerty,  
and R. P. Davey  
GigaScience 2018**

---



TECHNICAL NOTE

## GeneSeqToFamily: a Galaxy workflow to find gene families based on the Ensembl Compara GeneTrees pipeline

Anil S. Thanki\*, Nicola Soranzo, Wilfried Haerty and Robert P. Davey

Earlham Institute, Norwich Research Park, Norwich NR4 7UZ, UK

\*Correspondence address. Anil S. Thanki, Earlham Institute, Norwich Research Park, Norwich, NR4 7UZ, UK. E-mail: [Anil.Thanki@earlham.ac.uk](mailto:Anil.Thanki@earlham.ac.uk)

### Abstract

**Background:** Gene duplication is a major factor contributing to evolutionary novelty, and the contraction or expansion of gene families has often been associated with morphological, physiological, and environmental adaptations. The study of homologous genes helps us to understand the evolution of gene families. It plays a vital role in finding ancestral gene duplication events as well as identifying genes that have diverged from a common ancestor under positive selection. There are various tools available, such as MSOAR, OrthoMCL, and HomoloGene, to identify gene families and visualize syntenic information between species, providing an overview of syntenic regions evolution at the family level. Unfortunately, none of them provide information about structural changes within genes, such as the conservation of ancestral exon boundaries among multiple genomes. The Ensembl GeneTrees computational pipeline generates gene trees based on coding sequences, provides details about exon conservation, and is used in the Ensembl Compara project to discover gene families. **Findings:** A certain amount of expertise is required to configure and run the Ensembl Compara GeneTrees pipeline via command line. Therefore, we converted this pipeline into a Galaxy workflow, called GeneSeqToFamily, and provided additional functionality. This workflow uses existing tools from the Galaxy ToolShed, as well as providing additional wrappers and tools that are required to run the workflow. **Conclusions:** GeneSeqToFamily represents the Ensembl GeneTrees pipeline as a set of interconnected Galaxy tools, so they can be run interactively within the Galaxy's user-friendly workflow environment while still providing the flexibility to tailor the analysis by changing configurations and tools if necessary. Additional tools allow users to subsequently visualize the gene families produced by the workflow, using the Aequatus.js interactive tool, which has been developed as part of the Aequatus software project.

**Keywords:** Galaxy; Pipeline; Workflow; Genomics; Comparative Genomics; Homology; Orthology; Paralogy; Phylogeny; Gene Family; Alignment; Compara; Ensembl

### Introduction

The phylogenetic information inferred from the study of homologous genes helps us to understand the evolution of gene families (also referred to as “orthogroups”) that comprise genes sharing common descent [1]. This plays a vital role in finding

ancestral gene duplication events as well as in identifying regions under positive selection within species [2]. In order to investigate these low-level comparisons between gene families, the Ensembl Compara GeneTrees gene orthology and paralogy prediction software suite [3] was developed as a pipeline. The Ensembl GeneTrees pipeline uses TreeBest [4, 5] (part of

Received: 30 March 2017; Revised: 31 July 2017; Accepted: 18 January 2018

© The Author(s) 2018. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Table 1: Galaxy tools used in the workflow

Tool name	Tool ID	Version	Developed at Earlham Institute		Toolsheds reference
			Tool	Wrapper	
Get sequences by Ensembl ID	get_sequences	0.1.2	Yes	Yes	[17]
Get features by Ensembl ID	get_feature_info	0.1.2	Yes	Yes	[18]
Select longest coding sequence per gene	ensembl_longest_cds_per_gene	0.0.2	Yes	Yes	[19]
ETE species tree generator	ete_species_tree_generator	3.0.0b35	Yes	Yes	[20]
GeneSeqToFamily preparation	gstf_preparation	0.4.0	Yes	Yes	[21]
Transeq	EMBOSS: transeq101	5.0.0	No	No	[22]
NCBI BLAST+ makeblastdb	ncbi_makeblastdb	0.2.01	No	No	[23]
NCBI BLAST+ blastp	ncbi_blastp_wrapper	0.2.01	No	No	[23]
BLAST parser	blast_parser	0.1.2	Yes	Yes	[24]
hcluster_sg	hcluster_sg	0.5.1.1	No	Yes	[25]
hcluster_sg parser	hcluster_sg_parser	0.2.0	Yes	Yes	[26]
Filter by FASTA IDs	filter_by_fasta_ids	1.0	No	No	[27]
T-Coffee	t.coffee	11.0.8	No	Yes	[28]
Tranalign	EMBOSS: tranalign100	5.0.0	No	No	[22]
TreeBeST best	treebest_best	1.9.2	No	Yes	[29]
Gene Alignment and Family Aggregator	gafa	0.3.0	Yes	Yes	[30]
Unique	tp_sorted_uniq	1.1.0	No	No	[31]
FASTA-to-Tabular	fasta2tab	1.1.0	No	No	[32]
UniProt ID mapping and retrieval	uniprot_rest_interface	0.1	No	No	[33]

TreeFam [6]), which implements multiple independent phylogenetic methods and can merge the results into a consensus tree while trying to minimize duplications and deletions relative to a known species tree. This allows TreeBeST to take advantage of the fact that DNA-based methods are often more accurate for closely related parts of trees, while protein-based trees are better at longer evolutionary distances.

The Ensembl GeneTrees pipeline comprises 7 steps, starting from a set of protein sequences and performing similarity searching and multiple large-scale alignments to infer homology among them, using various tools: BLAST [7], hcluster.sg [8], T-Coffee [9], and phylogenetic tree construction tools, including TreeBeST. While these tools are freely available, most are specific to certain computing environments, are only usable via the command line, and require many dependencies to be fulfilled. Therefore, users are not always sufficiently expert in system administration to install, run, and debug the various tools at each stage in a chain of processes. To help ease the complexity of running the GeneTrees pipeline, we employed the Galaxy bioinformatics analysis platform to relieve the burden of managing these system-level challenges.

Galaxy is an open-source framework for running a broad collection of bioinformatics tools via a user-friendly web interface [10]. No client software is required other than a recent web browser, and users are able to run tools singly or aggregated into interconnected pipelines, called “workflows”. Galaxy enables users to not only create but also share workflows with the community. In this way, it helps users who have little or no bioinformatics expertise to run potentially complex pipelines in order to analyze their own data and interrogate results within a single online platform. Furthermore, pipelines can be published in a scientific paper or in a repository such as myExperiment [11] to encourage transparency and reproducibility.

In addition to analytical tools, Galaxy also contains plugins [12] for data visualization. Galaxy visualization plugins may be

interactive and can be configured to visualize various data types, for example, bar plots, scatter plots, and phylogenetic trees. It is also possible to develop custom visualization plugins and easily integrate them into Galaxy. As the output of the GeneSeqToFamily workflow is not conducive to human readability, we also provide a data-to-visualization plugin based on the Aequatus software [13]. Aequatus.js [14] is a new JavaScript library for the visualization of homologous genes that we extracted from the standalone Aequatus software. It provides a detailed view of gene structure across gene families, including shared exon information within gene families alongside gene tree representations. It also shows details about the type of interrelation event that gave rise to the family, such as speciation, duplication, and gene splits.

## Methods

The GeneSeqToFamily workflow has been developed to run the Ensembl Compara software suite within the Galaxy environment (Galaxy, RRID:SCR.006281), combining various tools alongside preconfigured parameters obtained from the Ensembl Compara pipeline to produce gene trees. Among the tools used in GeneSeqToFamily (listed in Table 1), some were existing tools in the Galaxy ToolShed [15], such as NCBI BLAST (NCBI BLAST, RRID:SCR.004870), TranSeq (Transeq, RRID:SCR.015647), Tranalign, and various format converters. Additional tools that are part of the pipeline were developed at the Earlham Institute (EI) and submitted to the ToolShed, that is, BLAST parser, hcluster\_sg, hcluster\_sg parser, T-Coffee, TreeBeST best, and Gene Alignment and Family Aggregator. Finally, we developed helper tools that are not part of the workflow itself but aid the generation of input data for the workflow, and these are also in the ToolShed, i.e. Get features by Ensembl ID, Get sequences by Ensembl ID, Select longest CDS per gene, ETE species tree generator, and GeneSeqToFamily preparation.



Figure 1: Overview of the GeneSeqToFamily workflow.

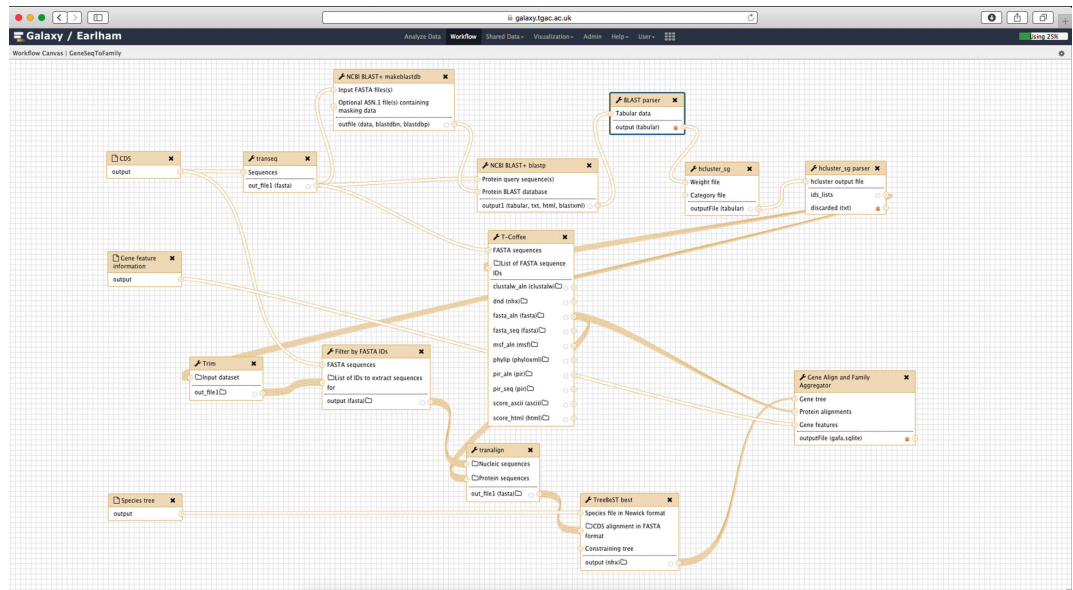


Figure 2: Screenshot from the Galaxy Workflow Editor showing the GeneSeqToFamily workflow.

The workflow comprises 7 main steps (see Figure 1), starting with translation from input coding sequences (CDS) to protein sequences, finding subsequent pairwise alignments of those protein sequences using BLASTP, and then the generation of clusters from the alignments using *hcluster\_sg*. The workflow then splits into 2 simultaneous paths, whereby in one path it performs the multiple sequence alignment (MSA) for each cluster using T-Coffee (T-Coffee, [RRID:SCR.011818](https://www.ebi.ac.uk/EMBL-EBI/TCoffee/)), while in the other it generates a gene tree with TreeBeST taking the cluster alignment and a species tree as input. Finally, these paths merge to aggregate the MSA, the gene tree, and the gene feature information (eg, transcripts, exons) into an SQLite [16] database for visualization and downstream reuse. All the workflow and data preparation steps are shown in Figure 2 and explained in detail below.

## Data generation and preparation

We developed a number of tools that assist in preparing the datasets needed by the workflows.

### Ensembl REST API tools

Galaxy tools were developed that use the Ensembl REST API [34] to retrieve sequence information (*Get sequences by Ensembl ID*) and feature information (*Get features by Ensembl ID*) by Ensembl ID from the Ensembl service. REST (REpresentational State Transfer) is an architecture style for designing networked appli-

cations [35] that encourages the use of standardized HTTP technology to send and receive data between computers. As such, these tools are designed to help users to retrieve existing data from Ensembl rather than requiring them to manually download datasets to their own computers and then subsequently uploading them into the workflow.

### ETE tools

We have developed the *ETE species tree generator* Galaxy tool, which uses the ETE toolkit [36] to generate a species tree from a list of species names or taxon IDs through the NCBI Taxonomy.

## GeneSeqToFamily workflow

### 0. GeneSeqToFamily preparation

Before GeneSeqToFamily can be run, a data preparation step must be carried out. We developed a tool called *GeneSeqToFamily preparation* to preprocess the input datasets (gene feature information and CDS) for the GeneSeqToFamily workflow. It converts a set of gene feature information files in GFF3 [37] and/or JavaScript Object Notation (JSON) [38] format to an SQLite database. It also modifies all CDS FASTA header lines by appending the species name to the transcript identifier, as required by *TreeBeST best*. It can also retain only the longest CDS sequence for each gene, as done in the GeneTrees pipeline.

We decided to use an SQLite database to store the gene feature information because the GFF3 format has a relatively inconvenient and unstructured additional information field (9th column) and because searching is much faster and more memory efficient in a database than in a text file like JSON or GFF3, especially when dealing with feature information for multiple large genomes.

### 1. CDS translation

#### Transeq

Transeq, part of the European Molecular Biology Open Software Suite (EMBOSS) (EMBOSS, [RRID:SCR.008493](#)) [39], is a tool to generate 6-frame translation of nucleic acid sequences to their corresponding peptide sequences. Here, we use Transeq to convert a CDS to protein sequences in order to run BLASTP (BLASTP, [RRID:SCR.001010](#)) and find protein clusters. However, since downstream tools in the pipeline, such as TreeBeST, require nucleotide sequences to generate a gene tree, the protein sequences cannot be directly used as workflow input and are instead generated with Transeq.

### 2. Preclustering alignment

#### BLAST

This workflow uses the BLAST wrappers [40] developed to run BLAST+ tools within Galaxy. BLASTP is run over the set of sequences against the database of the same input, as is the case with BLAST-all, in order to form clusters of related sequences.

#### BLAST parser

BLAST parser is a small Galaxy tool to convert the BLAST output into the input format required by hcluster.sg. It takes the BLAST 12-column output [41] as input and generates a 3-column tabular file, comprising the BLAST query, the hit result, and the edge weight. The weight value is simply calculated as minus  $\log_{10}$  of the BLAST e-value divided by 2, replacing this with 100 if this value is greater than 100. It also removes the self-matching BLAST results and lets the user filter out non-Reciprocal Best Hits.

### 3. Cluster generation

#### hcluster.sg

hcluster.sg performs clustering for sparse graphs. It reads an input file that describes the similarity between 2 sequences, and iterates through the process of grouping 2 nearest nodes at each iteration. hcluster.sg outputs a single list of gene clusters, each comprising a set of sequence IDs present in that cluster. This list needs to be reformatted using the hcluster.sg parser tool in order to be suitable for input into T-Coffee and TreeBeST (see below).

#### hcluster.sg parser

hcluster.sg parser converts the hcluster.sg output into a collection of lists of IDs, 1 list for each cluster. Each of these clusters will then be used to generate a gene tree via TreeBeST. The tool can also filter out clusters with a number of elements outside a specified range. The IDs contained in all discarded clusters are collected in separate output dataset. Since TreeBeST requires at least 3 genes to generate a gene tree, we configured the tool to filter out clusters with less than 3 genes.

Filter by FASTA IDs, which is available from the Galaxy Tool-Shed, is used to create separate FASTA files using the sequence IDs listed in each gene cluster.

### 4. Cluster alignment

#### T-Coffee

T-Coffee is a MSA package but can also be used to combine the output of other alignment methods (Clustal, MAFFT, Probcons, MUSCLE) into a single alignment. T-Coffee can align both nucleotide and protein sequences [9]. We use it to align the protein sequences in each cluster generated by hcluster.sg.

We modified the Galaxy wrapper for T-Coffee to take a single FASTA (as normal) and an optional list of FASTA IDs to filter. If a list of IDs is provided, the wrapper will pass only those sequences to T-Coffee, which will perform the MSA for that set of sequences, thus removing the need to create thousands of intermediate Galaxy datasets.

### 5. Gene tree construction

#### Tranalign

Tranalign [39] is a tool that reads a set of nucleotide sequences and a corresponding aligned set of protein sequences and returns a set of aligned nucleotide sequences. Here, we use it to generate CDS alignments of gene sequences using the protein alignments produced by T-Coffee.

#### TreeBeST "best"

TreeBeST (Tree Building guided by Species Tree) is a tool to generate, manipulate, and display phylogenetic trees and can be used to build gene trees based on a known species tree.

The "best" command of TreeBeST builds 5 different gene trees from a FASTA alignment file using different phylogenetic algorithms, then merges them into a single consensus tree using a species tree as a reference. In GeneSeqToFamily, TreeBeST "best" uses the nucleotide MSAs generated by Tranalign (at least 3 sequences are required) and a user-supplied species tree in Newick format [42] (either produced by a third-party software or through the ETE species tree generator data preparation tool, described above) to produce a GeneTree for each family, represented also in Newick format. The resulting GeneTree also includes useful annotations specifying phylogenetic information of events responsible for the presence/absence of genes, for example, "S" means speciation event, "D" means duplication, and "DCS" denotes the duplication score.

### 6. Gene alignment and family aggregation

#### Gene alignment and family aggregator

Gene alignment and family aggregator (GAFA) is a Galaxy tool that generates a single SQLite database containing the gene trees and MSAs, along with gene features, in order to provide a reusable, persistent data store for visualization of synteny information with Aequatius. GAFA requires gene trees in Newick format, the protein MSAs in fasta.aln format from T-Coffee, and gene feature information generated with the GeneSeqToFamily preparation tool.

Internally, GAFA converts each MSA from fasta.aln format to a simple CIGAR string [43]. An example of CIGAR strings for aligned sequences is shown in Figure 3, in which each CIGAR string changes according to other sequences.

The simple schema [44] for the generated SQLite database is shown in Figure 4.



Sequence1: NLYIQWLKDGPPSSGRPPPS  
 Sequence2: NLYIQWLKDQGPSSGRPPPS  
 Sequence3: GDAYAQWLADGGPSSGRPPPSG

Sequence1: -NLYIQWLKDGPPSSGRPPP-S  
 Sequence2: -NLYIQWLKDQGPSSGRPPP-S  
 Sequence3: GDAYAQWLADGGPSSGRPPPSG

CIGAR1: D19MDM  
 CIGAR2: D19MDM  
 CIGAR3: 22M

Figure 3: Showing how CIGAR for multiple sequence alignment is generated.

## 7. Visualization

### Aequatus visualization plugin

The SQLite database generated by the GAFA tool can be rendered using a new visualization plugin, Aequatus.js. The Aequatus.js library, developed as part of the Aequatus project, has been con-

figured to be used within Galaxy to visualize homologous gene structure and gene family relationships. This allows users to interrogate not only the evolutionary history of the gene family but also the structural variation (exon gain/loss) within genes across the phylogeny. Aequatus.js is available to download from GitHub [44], as visualization plugins cannot yet be submitted to the Galaxy ToolShed.

### Finding homology information for orphan genes

Although the GeneSeqToFamily workflow will assign most of the genes to orthogroups, many genes within a species might appear to be unique without homologous relationship to any other genes from other species. This observation could be the consequence of the parameters selected, choice of species, or incomplete annotations. This could also reflect real absence of homology, such as for rapidly evolving gene families. In addition to the GeneSeqToFamily workflow, we also developed 2 associated workflows to further annotate these genes by:

- 1) Retrieving a list of orphan genes from the GeneSeqToFamily workflow (see Figure 5) as follows:

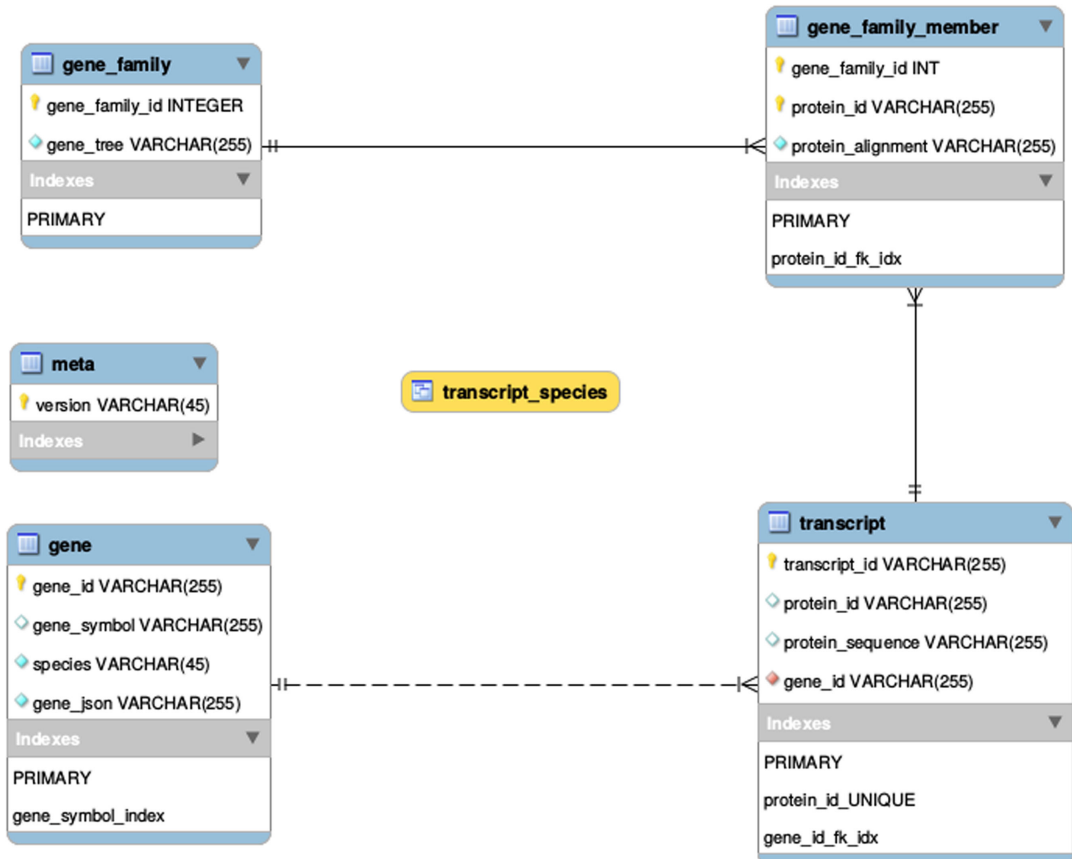


Figure 4: Schema of the gene alignment and family aggregator (GAFA) SQLite database, where transcript.species is a database view.



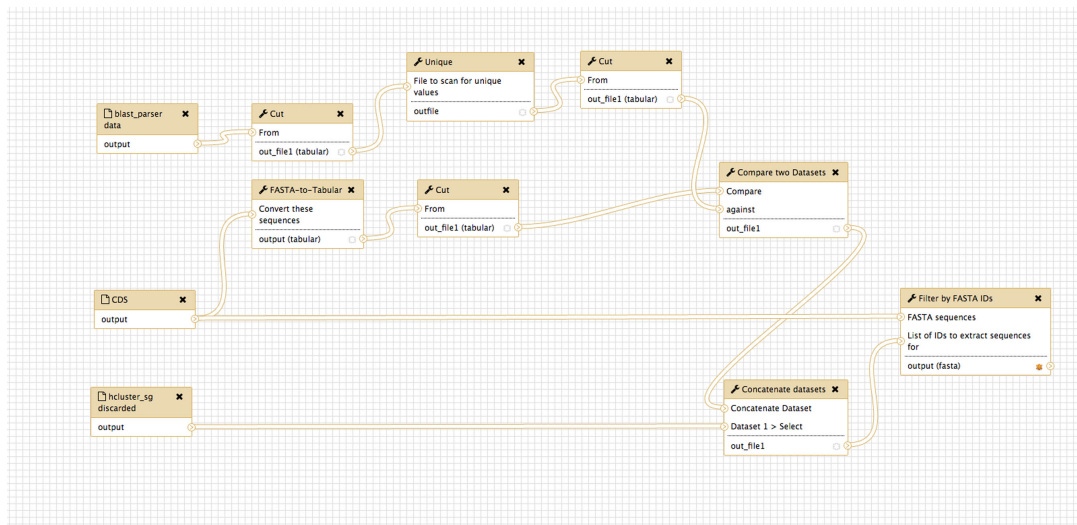


Figure 5: Screenshot from the Galaxy Workflow Editor showing the orphan gene finding workflow.

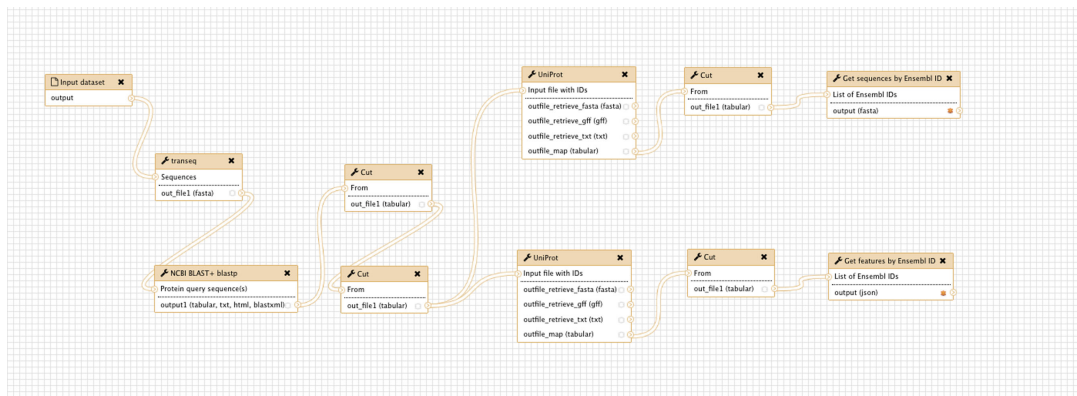


Figure 6: Screenshot from the Galaxy Workflow Editor showing the SwissProt workflow.

- Find the IDs of the sequences present in the input CDS of the GeneSeqToFamily workflow but not in the result of BLAST *parser* from the same workflow.
- Add to this list the IDs of the sequences discarded by *hcluster\_sg parser*.
- From the input CDS dataset, retrieve the respective sequence for each CDS ID (from the step above) using *Filter by FASTA IDs*.
- Retrieve Ensembl IDs (representing genes and/or transcripts) for each UniProt ID using *UniProt ID mapping and retrieval*.
- Get genomic information for each gene ID and CDS for each transcript ID from the core Ensembl database using *Get features by Ensembl ID* and *Get sequences by Ensembl ID*, respectively.

These unique CDS can be fed into the SwissProt workflow below to find homologous genes in other species.

- Finding homologous genes for some genes of interest using SwissProt (see Figure 6) as follows:
  - Translate CDS into protein sequences using *Transeq*.
  - Run BLASTP for the protein sequences against the SwissProt database (from NCBI).
  - Extract UniProt IDs from these BLASTP results, using the preinstalled Galaxy tool *Cut columns from a table* (tool id *Cut1*).

The results from this second workflow can be subsequently used as input to GeneSeqToFamily for familial analysis.

## Results

To validate the biological relevance of results from the GeneSeqToFamily workflow, we analyzed a small set of 23 homologous genes (1 transcript per gene) from *Pan troglodytes* (chimpanzee), *Homo sapiens* (human), *Rattus norvegicus* (rat), *Mus musculus* (mouse), *Sus scrofa* (pig), and *Canis familiaris* (domesticated dog). These genes are a combination of those found in 3 gene families, that is, monoamine oxidases (MAO A and B), insulin

Table 2: Set of parameters used in BLASTP and hcluster.sg to compare results

Tool	Parameter	Parameter set					
		A	B	C	D	E	F
BLASTP	Expectation value cutoff	1e-03	1e-03	1e-03	1e-10	1e-10	1e-10
	Query coverage per hsp	0	0	90	0	0	90
hcluster.sg	Minimum edge weight	0	20	0	0	20	20
	Minimum edge density between a join	0.34	0.50	0.34	0.34	0.50	0.50

BLASTP was configured with maximum number of HSPs set to 1, and hcluster.sg with single link clusters set to "no" and maximum size set to 500.

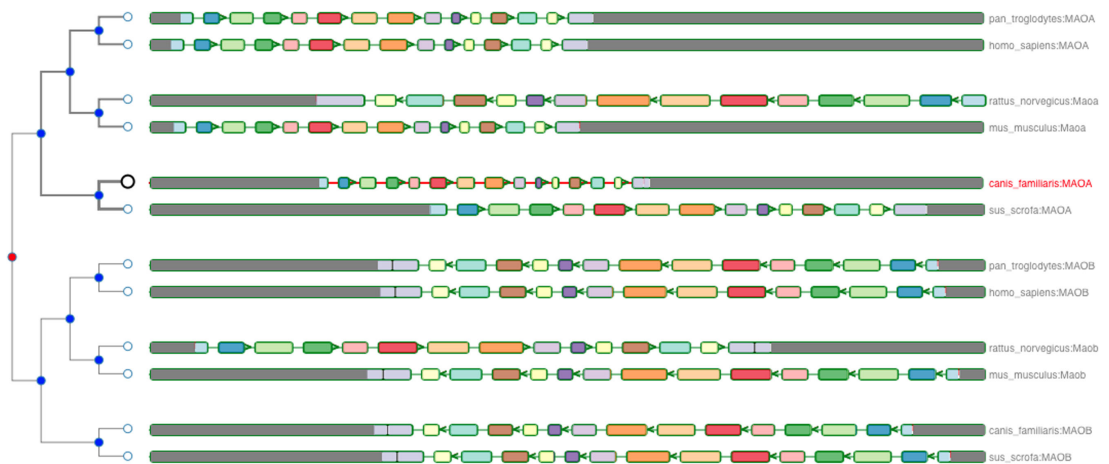


Figure 7: Homologous genes of monoamine oxidase (MAO) of *Canis familiaris* from *Mus musculus*, *Pan troglodytes*, *Homo sapiens*, *Rattus norvegicus*, *Sus scrofa*, and *Canis familiaris*.

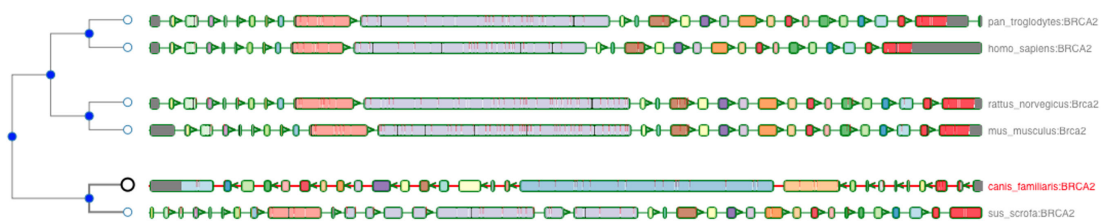


Figure 8: Homologous genes of BRCA2 of *Canis familiaris* from *Mus musculus*, *Pan troglodytes*, *Homo sapiens*, *Rattus norvegicus*, and *Sus scrofa*.

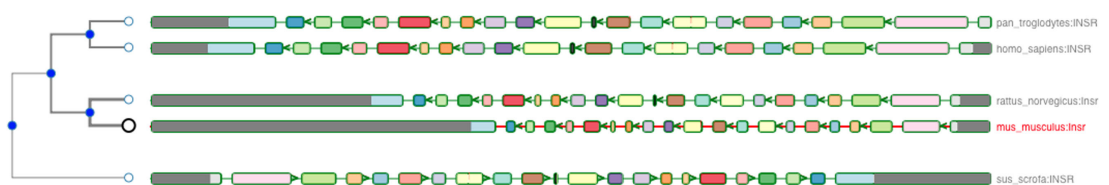


Figure 9: Homologous genes of insulin receptor (INSR) of *Mus musculus* from *Pan troglodytes*, *Homo sapiens*, *Rattus norvegicus*, and *Sus scrofa*.

receptor (INSR), and BRCA2, and were chosen because they are present in all 6 species yet distinct from each other.

Before running the workflow, feature information and CDS for the selected genes were retrieved from the core Ensembl database using the helper tools described above (*Get features by Ensembl ID* and *Get sequences by Ensembl ID*, respectively), and CDS were filtered to keep the longest CDS per gene. A species tree

was generated using *ETE species tree generator*, and inputs were prepared with *GeneSeqToFamily preparation*.

We ran the GeneSeqToFamily workflow on these data using the default parameters of the Ensembl Compara pipeline (Table 2, experiment D). This workflow generated 3 different gene trees, each matching exactly 1 gene family. Figures 7–9 show the resulting gene trees for MAO, BRCA2, and INSR gene

**Table 3:** Results of the GeneSeqToFamily workflow run with 7 different sets of parameters, the complete list of which are shown in Table 2

Analysis	Summary					
	A	B	C	D	E	F
Number of genes	754,149	754,149	754,149	754,149	754,149	754,149
Number of families	58,272	74,252	83,900	63,289	74,309	79,879
Number of larger families (>200)	435	168	56	350	167	46
Number of smaller families (<3)	30,563	40,530	44,295	33,308	40,579	41,794
Families (>3 and <200)	27,274	33,556	39,548	29,628	33,562	38,039
Largest family size	615	567	556	652	561	527
Average family size	11.38	7.36	5.36	10.04	7.35	5.09

families. Different colors of the nodes in each gene tree on the left-hand side of the visualization highlight potential evolutionary events, such as speciation, duplication, and gene splits. Homologous genes showing shared exons use the same color in each representation, including insertions (black blocks) and deletions (red lines). The GeneTrees for these genes are already available in Ensembl; we used them to validate our findings [45–48]. Our gene trees agree perfectly with the Ensembl GeneTrees, showing that the workflow generates biologically valid results. We have provided the underlying data for this example along with the submitted workflow in figshare [49].

We also studied the impact of the most important tool parameters on the gene families reconstructed by the workflow by running it on larger datasets, in particular, the reference proteomes of 754,149 sequences from 66 species established by the Quest for Orthologs (QfO) consortium [50]. We ran GeneSeqToFamily (up to the hcluster.sg step, where gene families are determined) with various sets of parameters (shown in Table 2) and performed statistical analysis on the resulting gene families (Table 3). Our results show that the number of gene families can vary quite distinctly with different BLASTP and hcluster.sg parameters. Stringent parameters (Parameter Set F) result in a large number of smaller families, while relaxed parameters (Parameter Set A) generate a small number of larger families, which may include distantly related genes. The parameters used by Ensembl Compara as default are shown in Parameter Set D.

We also performed benchmarking using the QfO benchmarking service [50]. QfO benchmarking focuses on assessing the accuracy of a tool to predict 1-to-1 orthology, while the GeneSeqToFamily workflow focuses on whole gene families, regardless of the type of homology among the members of a gene family. GeneSeqToFamily performs comparably to other tools benchmarked in QfO, even surpassing them for True Positive ortholog discovery in some parameter spaces. However, we found issues with the QfO service recording 1-to-many orthologs as false positives, hence reducing our overall specificity. Additional information about the corresponding results of benchmarking is available in Additional File 1.

## Conclusion

The ultimate goal of the GeneSeqToFamily is to provide a user-friendly workflow to analyze and discover homologous genes and their corresponding gene families using the Ensembl Compara GeneTrees pipeline within the Galaxy framework, where users can interrogate genes of interest without using the command-line while still providing the flexibility to tailor analysis by changing configurations and tools if necessary. We have shown it to be an accurate, robust, and reusable method to elucidate and analyze potentially large numbers of gene families in a range of model and nonmodel organisms. The workflow

stores the resulting gene families into an SQLite database, which can be visualized using the Aequatus.js interactive tool, as well as shared as a complete reproducible container for potentially large gene family datasets.

We invite the Galaxy community to undertake their own analyses and feedback improvements to various tools, and publish successful combinations of parameters used in the GeneSeqToFamily workflow to achieve better gene families for their datasets. We encourage this process by allowing users to share their own version of GeneSeqToFamily workflow for appraisal by the community.

## Future directions

In terms of core workflow functionality, we would like to incorporate pairwise alignment between pairs of genes for closely related species in addition of the MSA for the gene family, which will help users to compare orthologs and paralogs in greater detail.

We also plan to explicitly include the PantherDB resources [51]. Protein ANalysis THrough Evolutionary Relationships (PANTHER) is a classification system to characterize known proteins and genes according to family, molecular function, biological process, and pathway. The integration of PantherDB with GeneSeqToFamily will enable the automation of gene family validation and add supplementary information about those gene families, which could in turn be used to further validate novel genomics annotation.

Finally, we intend to add the ability to query the GAFA SQLite database using keywords in order to make it easy for users to find gene trees that include their genes of interest without needing to delve into the database itself.

## Availability and requirements

**Project name:** GeneSeqToFamily

**Project home page:** <https://github.com/TGAC/earlham-galaxytools/tree/master/workflows/GeneSeqToFamily>.

**Archived version:** 0.1.0

**Operating system(s):** Platform independent

**Programming language:** JavaScript, Perl, Python, XML, SQL

**Other Requirements:** Web Browser; for development: Galaxy

**Any restrictions to use by non-academics:** None

**License:** The MIT License (<https://opensource.org/licenses/MIT>)

## Availability of supporting data

The example files and additional datasets supporting the results of this article are available in figshare [49]. A virtual

image for Galaxy with necessary tools and installed workflows is available at Earlham repos [52]. Snapshots of the supporting data and code are hosted in the GigaScience GigaDB repository [53].

### Additional files

Table S1: Examples of ortholog pairs that are counted as False Positives by Qf0 benchmarking but are considered orthologs in the Ensembl Compara database.

Table S2: Set of parameters used in BLASTP and hcluster\_sg to compare results. BLASTP was configured with maximum number of HSPs set to 1, hcluster\_sg with single link clusters set to “no,” and maximum size set to 500.

Figure S1: Showing results for benchmarking on Quest for Orthologs using parameters shown in Parameter Set A <http://orthology.benchmarkservice.org/cgi-bin/gateway.pl?f=CheckResults&p1=2569682351ea7dfff3d5b083>.

Figure S2: Showing results for benchmarking on Quest for Orthologs using parameters shown in Parameter Set B <http://orthology.benchmarkservice.org/cgi-bin/gateway.pl?f=CheckResults&p1=1038fba4ba15c369b3d25541>.

Figure S3: Showing results for benchmarking on Quest for Orthologs using parameters shown in Parameter Set C <http://orthology.benchmarkservice.org/cgi-bin/gateway.pl?f=CheckResults&p1=ec4d223d24e0a7f54edd3692>.

Figure S4: Showing results for benchmarking on Quest for Orthologs using parameters shown in Parameter Set D <http://orthology.benchmarkservice.org/cgi-bin/gateway.pl?f=CheckResults&p1=dc81a95f182f5b5bee2dab3f>.

Figure S5: Showing results for benchmarking on Quest for Orthologs using parameters shown in Parameter Set E <http://orthology.benchmarkservice.org/cgi-bin/gateway.pl?f=CheckResults&p1=9d35f843bcae077e917a6452>.

Figure S6: Showing results for benchmarking on Quest for Orthologs using parameters shown in Parameter Set F <http://orthology.benchmarkservice.org/cgi-bin/gateway.pl?f=CheckResults&p1=0cbd5a0267b87491252348d6>.

### Competing interests

All authors report no competing interests.

### Acknowledgements

A.S.T., W.H., and R.P.D. are supported by BBSRC Institute Strategic Program grant funds awarded to E.I. N.S. is funded under the BBSRC Biomathematics and Bioinformatics Training fund (2014) awarded to E.I. This research was supported in part by the NBI Computing Infrastructure for Science Group, which provides technical support and maintenance to E's high-performance computing cluster and storage systems, which enabled us to develop this workflow.

We thank Matthieu Muffato from the European Bioinformatics Institute for his advice during the initial stage of the project.

### References

- Gabaldón T, Koonin EV. Functional and evolutionary implications of gene orthology. *Nat Rev Genet.* 2013;14(5):360–6.
- Jensen JD, Wong A, Aquadro CF. Approaches for identifying targets of positive selection. *Trends in Genetics.* 2007;23(11):568–77.

- Vilella AJ, Severin J, Ureta-Vidal A et al. EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome Research.* 2009;19(2):327–35.
- Ensembl. Ensembl/treebest. GitHub. <https://github.com/Ensembl/treebest>. Accessed 26 Jan 2016.
- Heng L. Constructing the TreeFam database. The Institute of Theoretical Physics, Chinese Academic of Science; 2006. <http://pfigshare-u-files.s3.amazonaws.com/1421613/PhDthesisliheng2006English.pdf>.
- Ruan J, Li H, Chen Z et al. TreeFam: 2008 Update. *Nucleic Acids Res.* 2008;36(Database issue):D735–40.
- Altschul SF, Gish W, Miller W et al. Basic local alignment search tool. *Journal of Molecular Biology.* 1990;215(3):403–10.
- Li H et al. hcluster\_sg: hierarchical clustering software for sparse graphs. <https://github.com/douglasgscotland/hcluster>. Accessed 26 Jan 2016.
- Notredame C, Higgins DG, Heringa J. T-Coffee: a novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology.* 2000;302(1):205–17.
- Afgan E, Baker D, van den Beek M et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res.* 2016;44(W1):W3–W10.
- Goble CA, Bhagat J, Alekseyevs S et al. myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Res.* 2010;38(suppl.2):W677–82.
- Goecks J, Eberhard C, Too T et al. Web-based visual analysis for high-throughput genomics. *BMC Genomics.* 2013;14:397.
- Thanki AS, Ayling S, Herrero J et al. AeQuatus: An open-source homology browser. *bioRxiv.* 2016;055632. doi:10.1101/055632.
- TGAC. TGAC/aequatus.js. GitHub. <https://github.com/TGAC/aequatus.js>. Accessed 26 Jan 2016.
- Blankenberg D, Von Kuster G, Bouvier E et al. Dissemination of scientific software with Galaxy ToolShed. *Genome Biol.* 2014;15(2):403.
- SQLite Home Page. <https://www.sqlite.org/>. Accessed 18 Nov 2016.
- Get sequences by Ensembl ID: Galaxy Tool Shed. [https://toolshed.g2.bx.psu.edu/view/earlhaminst/ensembl\\_get\\_sequences/](https://toolshed.g2.bx.psu.edu/view/earlhaminst/ensembl_get_sequences/). Accessed 20 Dec 2016.
- Get features by Ensembl ID: Galaxy Tool Shed. [https://toolshed.g2.bx.psu.edu/view/earlhaminst/ensembl\\_get\\_feature\\_info/](https://toolshed.g2.bx.psu.edu/view/earlhaminst/ensembl_get_feature_info/). Accessed 20 Dec 2016.
- Select longest CDS per gene: Galaxy Tool Shed. [https://toolshed.g2.bx.psu.edu/view/earlhaminst/ensembl\\_longest\\_cds\\_per\\_gene/](https://toolshed.g2.bx.psu.edu/view/earlhaminst/ensembl_longest_cds_per_gene/). Accessed 8 Mar 2017.
- ETE species tree generator: Galaxy Tool Shed. <https://toolshed.g2.bx.psu.edu/view/earlhaminst/ete/>. Accessed 20 Dec 2016.
- GeneSeqToFamily preparation: Galaxy Tool Shed. [https://toolshed.g2.bx.psu.edu/view/earlhaminst/gstf\\_preparation/](https://toolshed.g2.bx.psu.edu/view/earlhaminst/gstf_preparation/). Accessed 17 Mar 2017.
- EMBOSS: Galaxy Tool Shed. [https://toolshed.g2.bx.psu.edu/view/devteam/emboss\\_5/](https://toolshed.g2.bx.psu.edu/view/devteam/emboss_5/). Accessed 21 Dec 2016.
- NCBI BLAST plus: Galaxy Tool Shed. [https://toolshed.g2.bx.psu.edu/view/devteam/ncbi\\_blast\\_plus/](https://toolshed.g2.bx.psu.edu/view/devteam/ncbi_blast_plus/). Accessed 21 Dec 2016.
- BLAST parser: Galaxy Tool Shed. [https://toolshed.g2.bx.psu.edu/view/earlhaminst/blast\\_parser/](https://toolshed.g2.bx.psu.edu/view/earlhaminst/blast_parser/). Accessed 20 Dec 2016.
- hcluster\_sg: Galaxy Tool Shed. [https://toolshed.g2.bx.psu.edu/view/earlhaminst/hcluster\\_sg/](https://toolshed.g2.bx.psu.edu/view/earlhaminst/hcluster_sg/). Accessed 20 Dec 2016.

26. hcluster.sg parser: Galaxy Tool Shed. <https://toolshed.g2.bx.psu.edu/view/earlhaminst/hcluster.sg.parser/>. Accessed 20 Dec 2016.
27. Filter by FASTA IDs: Galaxy Tool Shed. <https://toolshed.g2.bx.psu.edu/view/galaxyp/filter.by.fasta.ids/>. Accessed 21 Dec 2016.
28. T-Coffee: Galaxy Tool Shed. <https://toolshed.g2.bx.psu.edu/view/earlhaminst/t.coffee/>. Accessed 20 Dec 2016.
29. TreeBeST best: Galaxy Tool Shed. <https://toolshed.g2.bx.psu.edu/view/earlhaminst/treebest.best/>. Accessed 20 Dec 2016.
30. Gene Align and Family Aggregator (GAFA): Galaxy Tool Shed. <https://toolshed.g2.bx.psu.edu/view/earlhaminst/gafa/>. Accessed 21 Dec 2016.
31. text\_processing: Galaxy Tool Shed. <https://toolshed.g2.bx.psu.edu/view/bgruening/text.processing/>. Accessed 19 Apr 2017.
32. FASTA-to-Tabular converter: Galaxy Tool Shed. <https://toolshed.g2.bx.psu.edu/view/devteam/fasta.to.tabular/>. Accessed 19 Apr 2017.
33. uniprot\_rest\_interface: Galaxy Tool Shed. [https://toolshed.g2.bx.psu.edu/view/bgruening/uniprot\\_rest\\_interface/](https://toolshed.g2.bx.psu.edu/view/bgruening/uniprot_rest_interface/). Accessed 20 Mar 2017.
34. Yates A, Beal K, Keenan S et al. The Ensembl REST API: Ensembl data for any language. *Bioinformatics*. 2015;31(1):143–5.
35. Representational State Transfer. <http://www.ietf.org/rfc/rfc2616.txt>. Accessed 4 Feb 2016.
36. Huerta-Cepas J, Serra F, Bork P. ETE 3: Reconstruction, analysis, and visualization of phylogenomic data. *Mol Biol Evol*. 2016;33(6):1635–8.
37. GFF3 - GMOD. <http://gmod.org/wiki/GFF3>. Accessed 4 Feb 2016.
38. JSON. <http://www.json.org>. Accessed 4 Feb 2016.
39. Rice P, Longden I, Bleasby A. EMBOSS: the European molecular biology open software suite. *Trends in Genetics*. 2000;16(6):276–7.
40. Cock PJA, Chilton JM, Grüning B et al. NCBI BLAST+ integrated into Galaxy. *GigaScience*. 2015;4:39.
41. National Center for Biotechnology Information (U.S.), Camacho C. BLAST(r) Command Line Applications User Manual. 2008. <https://www.ncbi.nlm.nih.gov/books/NBK279690/>. Accessed 26 Feb 2018.
42. “Newick’s 8:45” Tree Format Standard. <http://evolution.genetics.washington.edu/phylip/newick.doc.html>. Accessed 8 Apr 2016.
43. Sequence Alignment/Map Format Specification. <http://samtools.github.io/hts-specs/SAMv1.pdf>. Accessed 20 Dec 2016.
44. TGAC. TGAC/earlham-galaxytools. GitHub. <https://github.com/TGAC/earlham-galaxytools>. Accessed 21 Mar 2016.
45. Gene: BRAT1 (ENSG00000106009) - Gene tree - Homo sapiens - Ensembl genome browser 87. [http://dec2016.archive.ensembl.org/Homo\\_sapiens/Gene/Comparative\\_Tree?g=ENSG00000106009;r=7:2537877-2555727](http://dec2016.archive.ensembl.org/Homo_sapiens/Gene/Comparative_Tree?g=ENSG00000106009;r=7:2537877-2555727); Accessed 23 Dec 2016.
46. Gene: INSR (ENSG00000171105) - Gene tree - Homo sapiens - Ensembl genome browser 87. [http://dec2016.archive.ensembl.org/Homo\\_sapiens/Gene/Comparative\\_Tree?g=ENSG00000171105;r=19:7112255-7294034](http://dec2016.archive.ensembl.org/Homo_sapiens/Gene/Comparative_Tree?g=ENSG00000171105;r=19:7112255-7294034); Accessed 23 Dec 2016.
47. Gene: MAOA (ENSG00000189221) - Gene tree - Homo sapiens - Ensembl genome browser 87. [http://dec2016.archive.ensembl.org/Homo\\_sapiens/Gene/Comparative\\_Tree?g=ENSG00000189221;r=X:43654907-43746824](http://dec2016.archive.ensembl.org/Homo_sapiens/Gene/Comparative_Tree?g=ENSG00000189221;r=X:43654907-43746824); Accessed 23 Dec 2016.
48. Gene: MAOB (ENSG00000069535) - Gene tree - Homo sapiens - Ensembl genome browser 87. [http://dec2016.archive.ensembl.org/Homo\\_sapiens/Gene/Comparative\\_Tree?g=ENSG00000069535;r=X:43766611-43882447](http://dec2016.archive.ensembl.org/Homo_sapiens/Gene/Comparative_Tree?g=ENSG00000069535;r=X:43766611-43882447); Accessed 23 Dec 2016.
49. Thanki AS, Soranzo N, Haerty W et al. GeneSeqToFamily.zip. 2017. doi:10.6084/m9.figshare.4484141.v15.
50. Kuzniar A, van Ham RCHJ, Pongor S et al. The quest for orthologs: finding the corresponding gene across genomes. *Trends in Genetics*. 2008;24(11):539–51.
51. Mi H, Lazareva-Ulitsky B, Loo R et al. The PANTHER database of protein families, subfamilies, functions and pathways. *Nucleic Acids Research*. 2004;33(Database issue):D284–8.
52. Galaxy Virtual Image. [http://repos.tgac.ac.uk/vms/Galaxy\\_with\\_GeneSeqToFamily.ova](http://repos.tgac.ac.uk/vms/Galaxy_with_GeneSeqToFamily.ova). Accessed 28 Jul 2017.
53. Thanki AS, Soranzo N, Haerty W et al. Supporting data for “GeneSeqToFamily: a Galaxy workflow to find gene families based on the Ensembl Comparative GeneTrees pipeline”. *Giga-Science Database* 2018. <http://dx.doi.org/10.5524/100402>.

**Aequatus: an open-source homology  
browser**

**A. S. Thanki, N. Soranzo, J. Herrero,  
W. Haerty, and R. P. Davey,**





**GigaScience 2018**

---



## TECH NOTE

## Aequatus: an open-source homology browser

Anil S. Thanki <sup>1,\*</sup>, Nicola Soranzo <sup>1</sup>, Javier Herrero <sup>1,2</sup>,  
Wilfried Haerty <sup>1</sup> and Robert P. Davey <sup>1</sup><sup>1</sup>Earlham Institute, Norwich Research Park, Norwich, NR4 7UZ, UK and <sup>2</sup>Bill Lyons Informatics Centre, UCL Cancer Institute, 72 Huntley St., London, WC1E 6DD, UK\*Correspondence address. Earlham Institute, Norwich Research Park, Norwich, NR4 7UZ, UK. E-mail: [Anil.Thanki@earlham.ac.uk](mailto:Anil.Thanki@earlham.ac.uk)  <http://orcid.org/0000-0002-8941-444X>

## Abstract

**Background:** Phylogenetic information inferred from the study of homologous genes helps us to understand the evolution of genes and gene families, including the identification of ancestral gene duplication events as well as regions under positive or purifying selection within lineages. Gene family and orthogroup characterization enables the identification of syntenic blocks, which can then be visualized with various tools. Unfortunately, currently available tools display only an overview of syntenic regions as a whole, limited to the gene level, and none provide further details about structural changes within genes, such as the conservation of ancestral exon boundaries amongst multiple genomes. **Findings:** We present Aequatus, an open-source web-based tool that provides an in-depth view of gene structure across gene families, with various options to render and filter visualizations. It relies on precalculated alignment and gene feature information typically held in, but not limited to, the Ensembl Compara and Core databases. We also offer Aequatus.js, a reusable JavaScript module that fulfills the visualization aspects of Aequatus, available within the Galaxy web platform as a visualization plug-in, which can be used to visualize gene trees generated by the GeneSeqToFamily workflow.

**Keywords:** alignment, gene family, homology, phylogeny, synteny, visualization

## Introduction

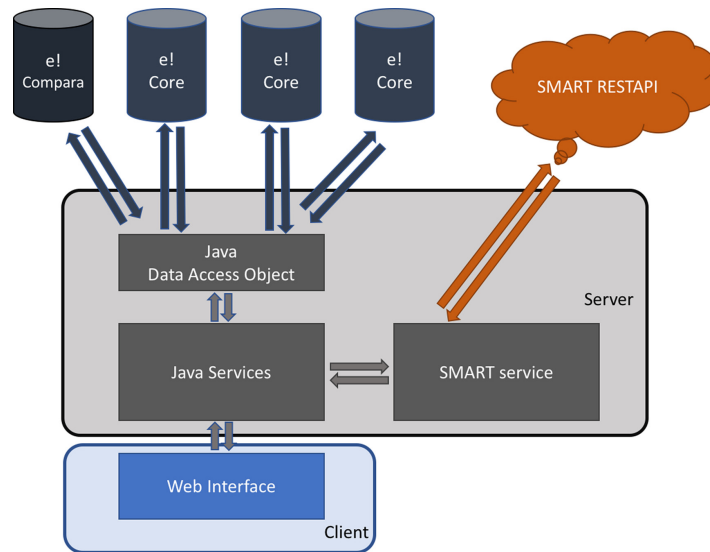
Sequence conservation across populations or species can be investigated at multiple levels from single nucleotides, to discrete sequences (e.g. transcription factor binding sites, exons, introns), genes, genomic blocks, and chromosomes. Analyses at each of these levels inform different evolutionary processes and time scales. While the vast majority of analyses focus on gene evolution, synteny (the conservation of genomic blocks between multiple species) can be used to trace chromosome evolutionary history [1] and infer evolutionary relationships between genes across or within species [2]. Synteny resolution and analysis typically involves carrying out multiple sequence alignments (MSAs) and phylogenetic reconstruction, comprising multiple steps that can be computationally intensive even for relatively small numbers of data points [3].

Many methods are available for the identification of genome-wide orthology (MSOAR [4], OrthoMCL [5], OMA [6], Homolo-

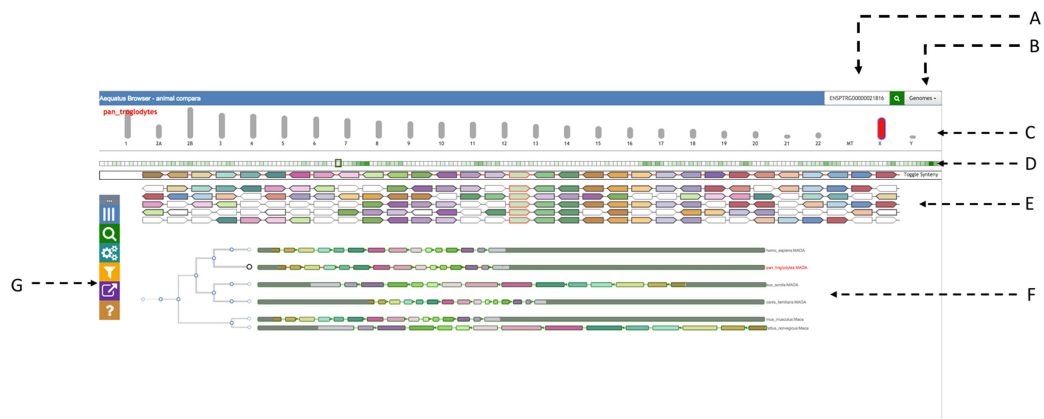
Gene [7], PhyOP [8], TreeFam [9], TreeBeST [10]). However, most of them do not incorporate taxonomic information (typically in the form of a species tree) while finding gene families, nor do they provide any information regarding transcript and protein structural changes across orthogroup members. The Ensembl GeneTrees pipeline [11], a computational workflow developed by the EMBL-EBI Ensembl Compara team, produces familial relationships based on clustering, MSA, and phylogenetic tree inference. The gene trees in Ensembl Compara are inferred with TreeBeST, which relies on a reference species tree to guide the process and calculates the probability of a gene tree in the context of species evolution. The data are stored in a relational database that contains information on gene families, syntenic regions, and protein families. In parallel, the Ensembl Core databases store gene feature information and other genomic annotations at the species level. The Ensembl project (release 90, August

Received: 18 June 2018; Revised: 6 September 2018; Accepted: 17 October 2018

© The Author(s) 2018. Published by Oxford University Press. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.



**Figure 1:** The Aequatus infrastructure, showing the interactions between the server-side implementation, connected to Ensembl Compara and Core database using Java Data Access Objects and Simple Modular Architecture Research Tool (SMART) server via REpresentational State Transfer (REST) application programming interface (API), and the client-side implemented using popular techniques such as JavaScript, jQuery, d3.js, and jQuery DataTables.



**Figure 2:** The main view of Aequatus. The header on top provides a search box (A) and a genome list (B). It is followed by the chromosomal view (C), where the selected chromosome is colored in red. Below there is an overview of genes (D) for the selected chromosome, followed by a zoomed area of the chromosome with genes shown in the gene order view (E) and by gene tree view (F). We are using arbitrary colors to distinguish syntenic genes (in gene order view) and matching exons (in gene tree view). The Aequatus control panel (G) is visible on the far left.

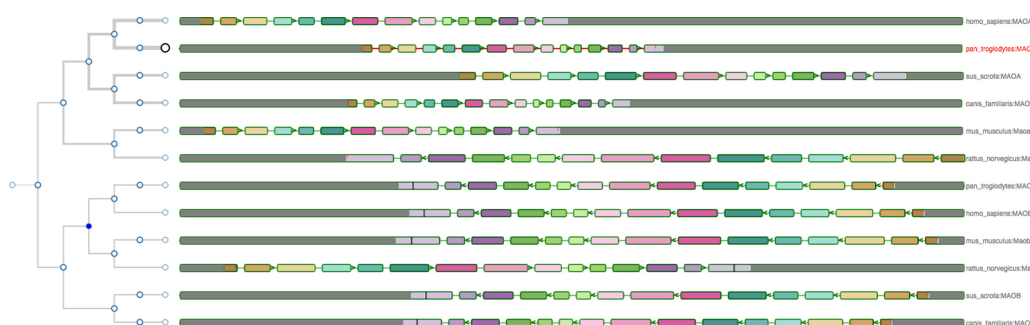
2017) at EMBL-EBI houses 100 vertebrate species [12], along with precomputed MSAs and gene family information.

Phylogenetic reconstruction is the most traditional method to represent and view comparative datasets across a given evolutionary distance, but specific tools such as Ensembl Browser [13], Genomicus [14], SyMAP [15], and MizBee [16] also exist to provide finer-grained information. These tools are able to provide an overview of syntenic regions as a whole, with only Genomicus reaching down to the gene order and orientation level. Conversely, phylogenetic trees retain ancestral information but do not represent the underlying information regarding structural changes within genes, such as the conservation of ancestral

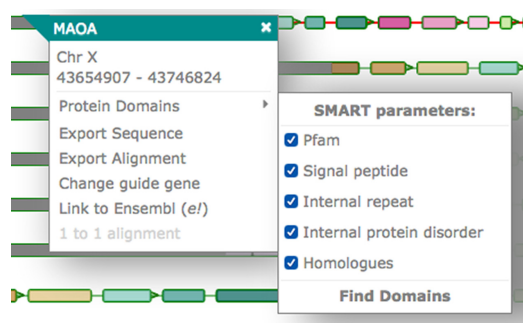
exon boundaries between multiple genomes or variants within genes that can be correlated to phenotypic changes. In order to build these gene-level visualizations, basic genomic feature information is required.

Therefore, we have developed Aequatus to bridge the gap between phylogenetic information and gene feature information. Here, we show that Aequatus allows the identification of exon/intron boundary changes and mutations, informing the user about underlying genetic changes.





**Figure 3:** The gene tree for the monoamine oxidase (MAO) gene, with the Chimp gene as the reference, alongside other homologous genes in the exon-focused view. Considering the gene tree on the left, it is clear that the MAO genes are separated into two clusters, corresponding to the MAOA and MAOB gene families.



**Figure 4:** The pop-up in the gene tree view when clicking on a gene. The pop-up contains the chromosome name and position and options to view the protein domains, export the sequence or the alignment, change the guide gene, connect to the Ensembl page for the gene, and view the pairwise alignment.

## Materials and Methods

Aequatus is built using open-source technologies and is divided into a typical server-client architecture: a web interface and a server backend (see Fig. 1).

The server-side component is implemented using the Java programming language. It retrieves and processes comparative genomics information directly from Ensembl Compara and Ensembl Core databases. Precalculated gene trees and genomic alignments, in the form of CIGAR strings [17], are held in Ensembl Compara, which are cross-referenced by Aequatus to Ensembl Core databases for each species to gather genomic feature information using the unique gene stable IDs.

The Aequatus web interface comprises well-known web technologies such as SVG, jQuery, JavaScript, and D3.js [18] to provide a fast and intuitive web-based browsing experience over complex data. Comparative and feature data are processed and rendered in an intuitive graphical interface to provide a visual representation of the phylogenetic and structural relationships among the set of chosen species.

Aequatus visualizes gene families using a phylogenetic tree generated from gene sequence conservation information, held in an Ensembl Compara database, and gene features from Ensembl Core database. Gene features are presented in the form of exon-intron boundaries and 5' and 3' untranslated regions (UTRs). In this gene tree view, users are able to select a gene from a given species as a “guide gene,” and the homologous genes

discovered through the comparative analysis are shown with respect to this guide gene. The representation of internal similarity among homologues is achieved by comparing the CIGAR strings for homologous genes with the CIGAR of the guide gene and mapping back to the homologous gene structure.

Aequatus is also able to visualize homologous genes in a customized Sankey view, using the d3.js [18] visualization library, and provides feature information in an interactive Tabular view, using the jQuery DataTable [19] library. Statistical information for each member in a set of homologues, such as percentage coverage, positivity, and identity, are fetched from *homology* and *homology\_member* tables of the Ensembl Compara database.

We have integrated a Simple Modular Architecture Research Tool (SMART) [20] service to search for and visualize domain information of a protein sequence. We use the SMART REpresentational State Transfer (REST) application programming interface (API) to retrieve protein domains, motifs, signal, and repeats information from the SMART server using protein sequences.

Finally, to complement these various visualizations for the homologous genes and their gene trees, Aequatus provides gene order information in the form of a syntenic view (see the “Gene Order” section below). For a selected gene, homologues are fetched from *homology* and *homology\_member* tables of the Ensembl Compara database. The neighboring genes for these homologous genes are retrieved from the Ensembl Core databases using positional information and organized into a syntenic representation. Much like the shared conserved exon depiction in the gene tree view, syntenic genes are colored based on the shared homology.

## Results

The landing page of Aequatus (see Fig. 2) contains a header with a search box (2A) and a dropdown list of species (2B), followed by a selectable chromosomal view underneath (2C).

Aequatus has a draggable control panel (2G) on the left-hand side that contains buttons to show/hide the chromosome selector on top, modify gene views and labels, access the search box, and the export options, as well as a link to the help pages.

### Aequatus user interface

Aequatus provides various ways to visualize gene trees and the inferred orthology/paralogy from them.

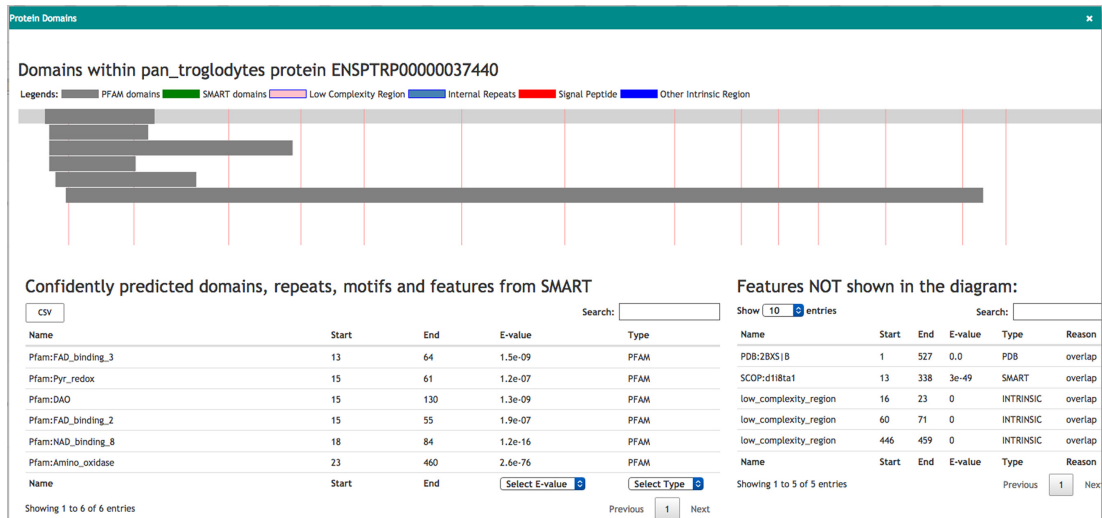


Figure 5: Visualization of the protein domain information for the protein ENSPTRP00000037440 retrieved from the SMART server. On the top, drawings of domains mapped on exons (shown with red lines). The tables below list the features shown in the diagram as well as hidden features.

#### Main gene trees view

The gene tree view (see Fig. 3) comprises a phylogenetic tree on the left, built from GeneTree information stored in a Ensembl Compara database [11]. Aequatus relates the genes through different events (e.g. duplication, speciation, and gene split) for the gene family and homologous genes against each respective node, which are colored based on the potential evolutionary event. Homologous genes are visualized by aligning them against a given guide gene. The selected guide gene is depicted as a larger circle black leaf node in the tree, with a red label on the right, while the other genes have a smaller circle leaf node and a gray label.

On the right, Aequatus depicts the internal gene structure, using a shared color scheme for coding regions, to represent similarity across homologues. Homologous genes are visualized by aligning them against a given guide gene. Aequatus is also able to indicate insertions and deletions in homologous genes with respect to shared ancestors. Black bars within exons represent insertions, while red lines represent deletions specific to a given gene compared with the guide.

Aequatus provides two view types for gene families. The first (default) view is exon focused (as in Fig. 3), where all introns are set to a fixed width, since long introns can adversely affect the visibility of surrounding exons. This provides easier browsing of the actual gene structure, especially when less screen real estate is available. Conversely, in the second view, all homologous genes are resized to the maximum available width in the web browser, showing introns and exons proportional to the real gene size. Users can switch between these views from the “Introns” settings in the control panel.

In gene tree view, gray blocks at the start and end of each gene represent UTRs, black bars within exons indicate insertions, red lines represent deletions specific to a given gene compared with the guide, and tiny arrows denote the coding strand of the gene.

**Pop-ups** Aequatus provides a contextual menu system via interactive pop-up menus, which are displayed when a user clicks on

a gene (see Fig. 4). Each pop-up shows the gene name and its position; a link to find protein domain information using SMART; links to export the protein sequence or the CIGAR alignment; an option to set the current gene as the guide in order to see insertions and deletions in homologous genes relative to the selected guide gene; a link out to the Ensembl page for the gene; and an option to view the pairwise alignment.

**Protein domain** Aequatus can provide an interactive visualization of the protein domains for the selected gene. Aequatus finds the protein domains by connecting to the SMART web server via its REST API and querying the protein sequence for domains, motifs, internal repeats, and similar information. In this view (see Fig. 5), a user can filter and sort domains based on type, E-value, position, and source of domain. The features shown in the diagram can be exported in comma-separated value (CSV) or Excel file format.

#### Homologous genes

The underlying information describing homologous genes contained within the Compara database schema can be visualized using either a tabular view or Sankey plot.

**Tabular view** The tabular view (see Fig. 6) contains statistical information for the homologous relationships. This view is dynamic, allowing the user to search for any homolog using a search box (6A) as well as filter results for the type of homology (6E) (1-to-1 orthologs, 1-to-many orthologs, and paralogs) or one or more specified species (6D). Homologous genes can be exported from the tabular view as Excel, CSV, or PDF.

Extra details for the pairwise alignment between homologues can be shown by using the “+” button for the homologue entry. The first button (6B) will show statistical comparisons for identity, coverage and similarity, while the second button (6C) will visualize the pairwise alignment with the gene structures as detailed in the “1-to-1 alignments” subsection below.

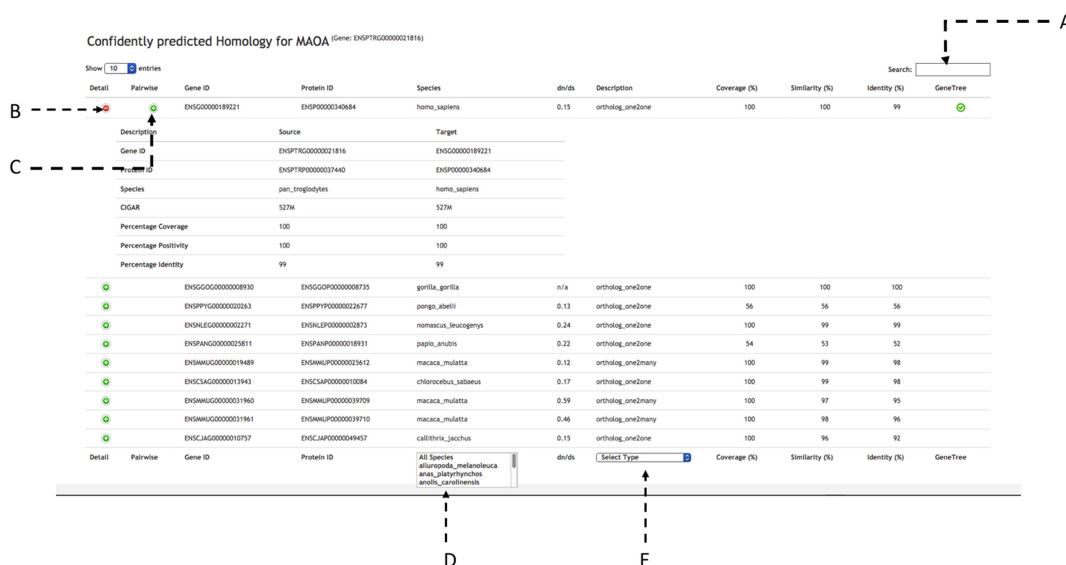


Figure 6: Homologues for the gene MAOA (ENSPTRG0000021816) in tabular view with statistical comparison about homologues. The tabular view contains a search box on top (A). There are two buttons to visualize statistical comparisons (C) and pairwise alignment (D) for each homologue. At the bottom it is possible to select from a list of species (D) and the type of homology (E).

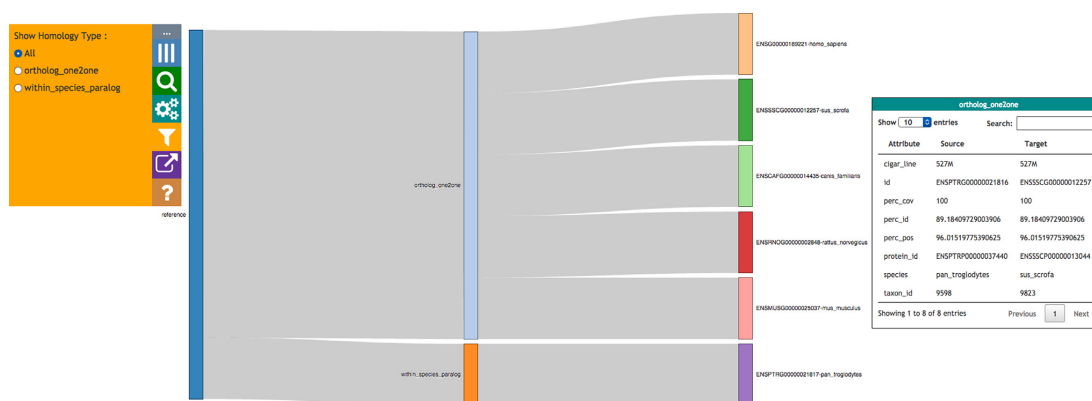


Figure 7: Homologues for a gene in Sankey format, grouped together by type of homology. The control panel on the left shows filters for the view. Additional information for any homologue can be retrieved by clicking on it; the information is then shown in a box on the right.

**Sankey view** The Sankey view (see Fig. 7) visualizes homology as an interactive diagram, where the homologues of a selected gene are distinguished by homology type, i.e. paralogs, 1-to-1 orthologs, or 1-to-many orthologs. The nodes for homologous genes are colored by species, which helps when finding genes from the same species in the case of 1-to-many and many-to-many orthologs.

When clicking on a homologous gene, additional details for the homologous pair are displayed in the info panel on the right-hand side.

### 1-to-1 alignments

Aequatus provides 1-to-1 alignments between homologous genes to facilitate pairwise comparisons. These 1-to-1 align-

ments (Fig. 8) can be seen by clicking on the corresponding option either in the pop-up for the gene tree view or in the homologous genes tabular view. This will fetch the relevant alignment from the homology table of the Ensembl Compara database and visualize it based on the gene structure (8A) together with the pairwise protein sequence alignments (8B).

### Gene order

Genes that share a common ancestor and are part of a consecutive block of genes are likely to have a transcriptional and/or functional relationship [21]. Hence, inferred homologues that are present in all species and in the same order are more likely to be real homologues. In the Gene Order view, neighboring genes are displayed for the selected gene and its homologues (shown

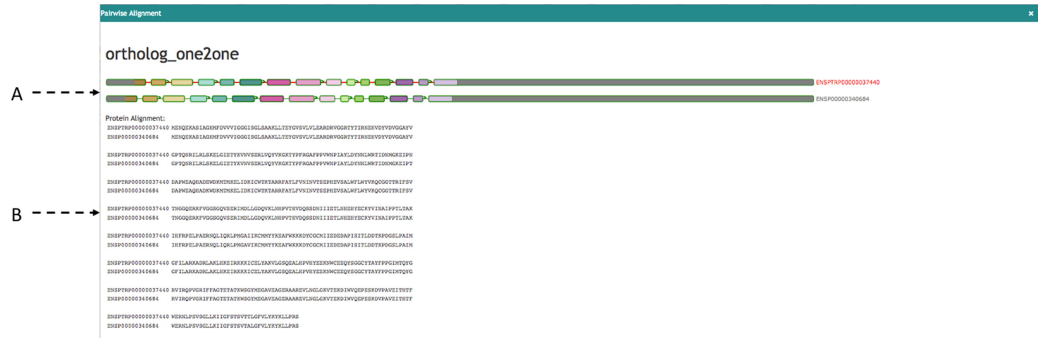


Figure 8: The 1-to-1 alignments between homologous genes. (A) Visualizing alignment on gene structure and (B) visualizing pairwise sequence alignments.

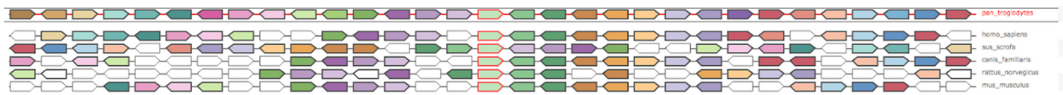


Figure 9: Gene Order for the MAOA gene in Pan troglodytes, where they are colored by homologous genes. The selected gene and its homologous have a red border. White genes are the ones that do not have any homologous in the current visible region.

in Fig. 9). Homologues of the genes in neighboring species are colored based on the matching genes from the reference species. Clicking on a gene feature will open a search panel with various viewing options, and mousing over a given gene will highlight all homologous genes within the same region. The syntenic view complements the main functionality of Aequatus by providing evidence for the conservation information for the genes of interest.

**Search**

Aequatus has keyword-based search functionality, whereby the user can provide search terms and a list of all the relevant genes is returned. Aequatus can query for matching gene symbols, Ensembl stable IDs (unique identifiers in the Ensembl project for each genomic annotation), common names for genes and proteins, or any keyword in the description. Search results then allow the user to visualize the corresponding gene tree view or homologous genes in the tabular or Sankey views.

**Export**

Users can export data at different points in the visualization. In the gene tree view, the underlying genomic data for the gene families can be exported in various forms, such as a list of gene IDs, the sequence alignments, or the gene trees in Newick [22] or JavaScript Object Notation (JSON) [23] format, for use in downstream tools. The tabular view can be exported in CSV, XLS, and PDF format.

**Persistent uniform resource locators**

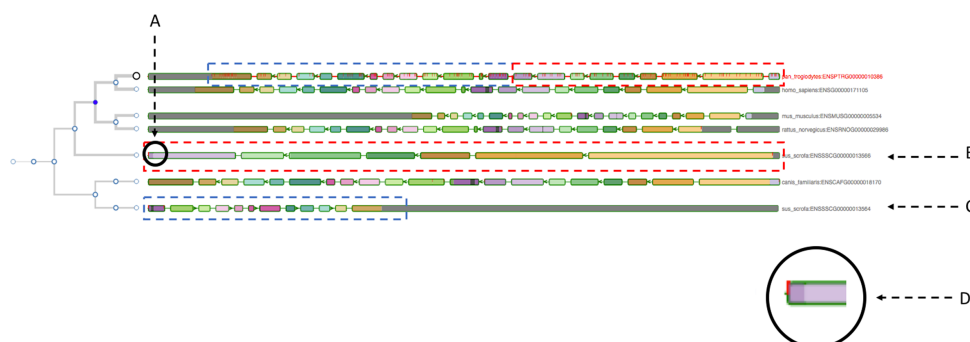
Aequatus provides persistent unique uniform resource locators (URLs) to enable consistent access to genes of interest, making it easy to go back to the results of a previous search, to share information with collaborators, or for use in publications. Users can share the link for the visualization of a specific gene, the results of a search for a term, or a specific reference to a given species and chromosome.

**Discussion**

The ultimate goal of Aequatus is to provide a unique and informative way to render and explore complex relationships between genes from various species at a level of detail that has so far been unrealized in a single platform. Supplementary Table S1 shows a detailed comparison of Aequatus with various phylogenetic visualization tools, which highlights the signature feature of Aequatus, i.e. genetic structural comparison. Supplementary Figs. S1–S3 allow a comparison of the visualizations of monoamine oxidase B (MAOB) genes from the tools offering a gene tree-focused view.

While applicable to species with high-quality gold-standard reference genomes present in core database resources such as human or mouse, Aequatus has been designed to accommodate users who need to explore large, fragmented, nonmodel genome references that are held in institutional databases. Comparing nonmodel organism genes with gold standard genomes allows the identification of exon/intron boundary changes and mutations, informing the user about underlying genetic changes, but can also highlight mis-annotations, pseudogenes [24], or polyploidization (see Fig. 10). We are currently testing Aequatus with a range of nonmodel organisms, such as koala, polyploid crops, and spiny mouse. As Aequatus can visualize relationships using simple CIGAR strings, any tool that outputs this format can use Aequatus to view them. We produce input for Aequatus using the GeneSeqToFamily pipeline, a freely available Galaxy workflow [25] for finding and visualizing gene families for genomes that are not available from Ensembl databases.

In order to make Aequatus more accessible and reusable, the gene tree visualization module from the stand-alone Aequatus browser is available as Aequatus.js [26], an open-source JavaScript library. In this way, it preserves the interactive functionality of the Aequatus browser tool but can be integrated with other third-party web applications. We have demonstrated this by integrating the Aequatus.js library into Galaxy [27], where gene families generated by running the GeneSeqToFamily workflow can be visualized using the Aequatus plug-in within Galaxy.



**Figure 10:** The gene tree view for the insulin receptor (INSR) gene, with the chimp gene as the guide alongside other homologous genes. (B and C) point to two genes from the pig genome, which are matching two different parts of the guide gene (shown with dotted rectangles in corresponding colors). (A) instead indicates an exon of one of the pig genes (enlarged in D) matching two adjacent exons of the guide gene. All these clues may suggest a potential gene split event or just a mis-annotation.

A publicly available instance of the GeneSeqToFamily workflow and the Aequatus plug-in is available on the UseGalaxy.eu server [28].

### Future directions

The main extension to the functionalities of Aequatus is the incorporation of Ensembl REST API functionality [29], where Aequatus will be able to retrieve information directly from Ensembl Compara and Core databases held at the EMBL-EBI, without any need for local database configuration. While this will mean that users will need a reliable Internet connection, it will reduce the need for local storage space for the Core databases, improving the portability of Aequatus.

We also intend to containerize Aequatus using Docker and CyVerse UK [30], and BioConda [31] with Galaxy [25, 27]. We will produce new APIs between Aequatus and TGAC Browser [32] to provide a comprehensive solution for genome analysis and exploration focused on non-model organisms.

### Availability of source code and requirements

- Project name: Aequatus: Earlham Institute's Synteny Browser
- Project home page: <https://github.com/TGAC/Aequatus>
- Demo server: <http://aequatus.earlham.ac.uk/>
- Operating systems: Platform independent
- Programming language: Java, JavaScript
- Other requirements: Java 1.7, Maven 2.0, Apache Tomcat, Ensembl Compara and Core MySQL databases.
- License: GNU General Public License v3 and MIT license.

### Availability of supporting data

Snapshots of the code are available from the GigaScience GigaDB database [33].

### Additional files

**Table S1:** Comparison of various phylogenetic visualisation tools with Aequatus

**Figure S1:** The genetree for the monoamine oxidase (MAO) genes, with the Chimp gene as the reference, alongside other homologous genes in the exon-focused view.

**Figure S2:** Ensembl visualising genetree for MAOB with the chimp gene as the reference, alongside other homologous genes along with alignments.

**Figure S3:** Genomicus visualising syntenic genes for MAOB with the chimp gene as the reference with neighbouring genes.

### Competing interests

The authors declare that they have no competing interests.

### Funding

This work was strategically funded by the Biotechnology and Biological Sciences Research Council (BBSRC)(BBS/E/T/000PR5885, BBS/E/T/000PR9817) and through a EU TransPlant grant (BBS/E/T/000GP006). GeneSeqToFamily and the EI Galaxy platform are funded through the BBSRC-supported EI National Capability in e-Infrastructure (BBS/E/T/000PR9814).

This work was supported in part by the NBI Computing Infrastructure for Science Group, which provides technical support and maintenance to EI's high-performance computing cluster and storage systems, which enabled us to develop this tool.

### Abbreviations

API: application programming interface; CSV: comma-separated values file; MAO: monoamine oxidase; MSA: multiple sequence alignment; REST: REpresentational State Transfer; SMART: Simple Modular Architecture Research Tool; UTR: untranslated region.

### References

1. Myers P. Synteny: Inferring Ancestral Genomes. 2008 <https://www.nature.com/scitable/topicpage/synteny-inferring-ancestral-genomes-44022>. Accessed 9 June 2018
2. Vilella AJ, Severin J, Ureta-Vidal A, et al. EnsemblCompara GeneTrees: complete, duplication-aware phylogenetic trees in vertebrates. *Genome Res.* 2009;19:327–35.
3. Edgar RC, Batzoglou S. Multiple sequence alignment. *Curr Opin Struct Biol.* 2006;16:368–73.
4. Fu Z, Chen X, Vacic V et al. MSOAR: a high-throughput ortholog assignment system based on genome rearrangement. *J Comput Biol.* 2007;14:1160–75.

5. Li L, Stoeckert CJ, Jr, Roos DS. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res.* 2003;13:2178–89.
6. Altenhoff AM, Dessimoz C. Inferring orthology and paralogy. *Methods in Molecular Biology.* 2012, p. 259–79.
7. Wheeler DL, Barrett T, Benson DA, et al. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* 2008;36:D13–21.
8. Goodstadt L, Ponting CP. Phylogenetic reconstruction of orthology, paralogy, and conserved synteny for dog and human. *PLoS Comput Biol.* 2006;2:e133.
9. Li H, Coghlan A, Ruan J, et al. TreeFam: a curated database of phylogenetic trees of animal gene families. *Nucleic Acids Res.* 2006;34:D572–80.
10. TreeSoft: TreeBeST. <http://treesoft.sourceforge.net/treebest.shtml>. Accessed 9 June 2018.
11. Clamp M, Andrews D, Barker D, et al. Ensembl 2002: accommodating comparative genomics. *Nucleic Acids Res.* 2003;31:38–42.
12. Zerbino DR, Achuthan P, Akanni W, et al. Ensembl 2018. *Nucleic Acids Res.* 2018;46:D754–61.
13. Stalker J, Gibbins B, Meidl P, et al. The Ensembl web site: mechanics of a genome browser. *Genome Res.* 2004;14:951–5.
14. Muffato M, Louis A, Poisnel CE, et al. Genomicus: a database and a browser to study gene synteny in modern and ancestral genomes. *Bioinformatics.* 2010;26:1119–21.
15. Soderlund C, Nelson W, Shoemaker A, et al. SyMAP: a system for discovering and viewing syntenic regions of FPC maps. *Genome Res.* 2006;16:1159–68.
16. Meyer M, Munzner T, Pfister H. MizBee: a multiscale synteny browser. *IEEE Trans Vis Comput Graph.* 2009;15:897–904.
17. Sequence Alignment/Map Format Specification. <http://samtools.github.io/hts-specs/SAMv1.pdf>. Accessed 9 June 2018.
18. Bostock M. D3.js - Data-Driven Documents. <http://d3js.org/>. Accessed 9 June 2018.
19. DataTables | Table plug-in for jQuery. <https://datatables.net>. Accessed 9 June 2018.
20. Schultz J, Milpetz F, Bork P, et al. SMART, a simple modular architecture research tool: identification of signaling domains. *Proc Natl Acad Sci.* 1998;95:5857–64.
21. Dávila López M, Martínez Guerra JJ, Samuelsson T. Analysis of gene order conservation in eukaryotes identifies transcriptionally and functionally linked genes. *PLoS One.* 2010;5(5):e10654. doi: 10.1371/journal.pone.0010654.
22. Newick Format. [http://evolution.genetics.washington.edu/phylip/newick\\_doc.html](http://evolution.genetics.washington.edu/phylip/newick_doc.html). Accessed 8 Apr 2018.
23. JSON format. <http://www.json.org>. Accessed 9 June 2018.
24. Vanin EF. Processed pseudogenes: characteristics and evolution. *Annu Rev Genet.* 1985;19:253–72.
25. Thanki AS, Soranzo N, Haerty W, et al. GeneSeqToFamily: a Galaxy workflow to find gene families based on the Ensembl Compara GeneTrees pipeline. *GigaScience.* 2018;7(3):1–10.
26. Thanki AS, Davey RP. TGAC/aequatus.js GitHub Repository. <https://github.com/TGAC/aequatus.js>. Accessed 9 June 2018.
27. Afgan E, Baker D, van den Beek M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res.* 2016;44:W3–10.
28. UseGalaxy.eu <https://usegalaxy.eu>. Accessed 8 June 2018.
29. Yates A, Beal K, Keenan S, et al. The Ensembl REST API: Ensembl data for any language. *Bioinformatics.* 2015;31:143–5.
30. Goff SA, Vaughn M, McKay S, et al. The iPlant Collaborative: cyberinfrastructure for plant biology. *Front Plant Sci.* 2011;2:34.
31. Grüning B, Dale R, Sjödin A, et al. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat Methods.* 2018;15(7):475–6.
32. Thanki AS, Bian X, Davey RP. TGAC Browser: visualisation solutions for big data in the genomic era. <http://browser.earlham.ac.uk/>. Accessed 8 June 2018.
33. Thanki AS, Soranzo N, Herrero J, et al. Supporting data for “Aequatus: An open-source homology browser.” *GigaScience Database.* 2018. <http://dx.doi.org/10.5524/100509>.

**ViCTree: an automated framework for  
taxonomic classification from protein  
sequences**

**S. Modha, A. S. Thanki, S. F. Cotmore,  
A. J. Davison, and J. Hughes**

**Bioinformatics 2018**

---



## Phylogenetics

# ViCTree: an automated framework for taxonomic classification from protein sequences

Sejal Modha<sup>1,\*</sup>, Anil S. Thanki<sup>2</sup>, Susan F. Cotmore<sup>3</sup>, Andrew J. Davison<sup>1</sup> and Joseph Hughes<sup>1</sup>

<sup>1</sup>MRC-University of Glasgow Centre for Virus Research, Glasgow, UK, <sup>2</sup>Earlham Institute, Norwich Research Park, Norwich, UK and <sup>3</sup>Yale University Medical School, New Haven, CT, USA

\*To whom correspondence should be addressed.

Associate Editor: Janet Kelso

Received on July 27, 2017; revised on January 8, 2018; editorial decision on February 16, 2018; accepted on February 20, 2018

### Abstract

**Motivation:** The increasing rate of submission of genetic sequences into public databases is providing a growing resource for classifying the organisms that these sequences represent. To aid viral classification, we have developed ViCTree, which automatically integrates the relevant sets of sequences in NCBI GenBank and transforms them into an interactive maximum likelihood phylogenetic tree that can be updated automatically. ViCTree incorporates ViCTreeView, which is a JavaScript-based visualization tool that enables the tree to be explored interactively in the context of pairwise distance data.

**Results:** To demonstrate utility, ViCTree was applied to subfamily *Densovirinae* of family *Parvoviridae*. This led to the identification of six new species of insect virus.

**Availability and implementation:** ViCTree is open-source and can be run on any Linux- or Unix-based computer or cluster. A tutorial, the documentation and the source code are available under a GPL3 license, and can be accessed at [http://bioinformatics.cvr.ac.uk/victree\\_web/](http://bioinformatics.cvr.ac.uk/victree_web/).

**Contact:** sejal.modha@glasgow.ac.uk

**Supplementary information:** [Supplementary data](#) are available at *Bioinformatics* online.

## 1 Introduction

The increasing rate at which sequence data are being deposited into public databases is providing a tremendous resource for taxonomic classification throughout biology. Phylogenetic analysis provides a key means of integrating these data and inferring the evolutionary relationships that form the basis of classification. However, preparing datasets for such analyses is often time-consuming, and the phylogenies obtained are typically not easy to update. Consequently, systematic approaches are being developed that automate the various steps involved.

The Ensembl Compara GeneTree pipeline (Vilella *et al.*, 2009) provides a comprehensive gene-orientated phylogenetic resource. It has a powerful analytical backend for classifying genes and gene families on the basis of detecting orthology among the complete genomes available in the Ensembl framework. Automated phylogeny-based classification is also implemented in the mor package

(<http://www.clarku.edu/faculty/dhibbett/clarkfungald/>), which has been applied to fungal taxa by aligning 28S rRNA sequences from GenBank and generating a phylogeny that can be updated by a node-based classification approach (Hibbett *et al.*, 2005). The 16S and 18S rRNA sequences can also inform classification, and are employed in tools such as STAP (Wu *et al.*, 2008) and EukRef (<http://eukref.org/curation-pipeline-overview/>). A more general approach is implemented in PUmPER (Izquierdo-Carrasco *et al.*, 2014), which has been applied to the classification of plants (<http://port.noy.iplantcollaborative.org/view/tree/10b17429d13160ac1cd07e30bb42fd9b>). However, PUmPER employs PHLAWD (Smith *et al.*, 2009) to collate sequences and build multiple alignments, which in turn relies on GenBank annotations to retrieve nucleotide sequences.

The tools described above were developed for specific types of non-viral organisms and have limited applications to the classification



of viruses. Viruses exhibit an enormous range of sequence diversity and cannot be integrated into a tree of life because they lack genes that are universally conserved in other organisms and that therefore may be used for barcoding (e.g. those encoding rRNAs or enzymes such as cytochrome c oxidase subunit I and ribulose-bisphosphate carboxylase). Also, GenBank annotations of viral genes are often not standardized and are thus unreliable for retrieving sequences. Moreover, none of the tools mentioned above presents both pairwise distances and phylogenies. This dual facility is important in viral classification because precise distance thresholds are frequently stipulated as demarcation criteria, and these may vary widely among families and even among genera due to differences in evolutionary rate.

Viruses are classified formally by the International Committee on Taxonomy of Viruses (ICTV; <http://www.ictvonline.org/>) into three ranks (family, genus and species), and, in some cases, two further ranks (order and subfamily). The huge diversity of viruses has the effect that the criteria used and the relative emphasis placed on each vary widely from family to family. However, sequence-based criteria (typically based on amino acid, rather than nucleotide, sequences) are prominent, and include simple measures of distance and increasingly powerful phylogenetic measures, as in the case of family *Parvoviridae* discussed below (Cotmore et al., 2014). The rapidly increasing volume of viral sequence information and the limitations of existing tools in relation to viruses necessitates the development of automated bioinformatic solutions that are suited specifically to viruses (Simmonds, 2015; Simmonds et al., 2017). At least two tools in this category have been used in viral classification: PASC provides a web-based interface for visualizing distances among automatically aligned sequences (Bao et al., 2014), and DEARC takes a sophisticated approach to identifying taxonomically significant thresholds in the distribution of distance data (Lauer and Gorbalenya, 2012). These tools were developed for exploring the pairwise distance criteria used to define species and genus boundaries within a taxon. ViPTree, a web-based classification tool, employs a genome-wide similarity method to classify viral sequences, and generates a static viral proteomic tree to illustrate the phylogenetic relationships among the existing sequences available in the GenomeNet/Virus-Host database (Nishimura et al., 2017). Useful as these tools are, none of them presents the results of pairwise distances along with the phylogeny, a feature that is important to viral taxonomists as they transition from using distance-based to phylogeny-based classification methods.

As an aid to integrating GenBank data into taxonomic analyses that can be updated automatically, we present ViCTree, a pipeline that retrieves the relevant viral sequences from GenBank, aligns and clusters them, and generates a maximum likelihood phylogenetic tree combined with distance data. The results are rendered by using ViCTreeView, which is a Javascript-based tool that enables users to visualize and explore distances in the context of the tree. ViCTree is automated so that the phylogenies are synchronized with the GenBank data, and the results are versioned on GitHub. The pipeline is flexible and broadly applicable to examining the phylogenetic relationships that underpin viral taxa.

## 2 Framework

All modules and tools implemented in ViCTree are open-source. The framework is a combination of a Bash shell script for automatically generating multiple sequence alignments and phylogenetic trees, and JavaScript for visualizing and exploring the trees in combination with the underlying distance data.

### 2.1 Phylogeny building

A curated set of seed protein sequences must be provided that spans the known diversity of a viral taxon. These sequences and the relevant GenBank taxonomic ID (specified at any rank) are submitted to the start of the ViCTree pipeline (Fig. 1), and all available protein sequences that bear the taxonomic ID are automatically downloaded from GenBank. Rather than filtering on the basis of GenBank sequence annotations, which are sometimes incomplete or incorrect, BLAST (Altschul et al., 1990) is used to compare the downloaded sequences with all the seed sequences. Significant

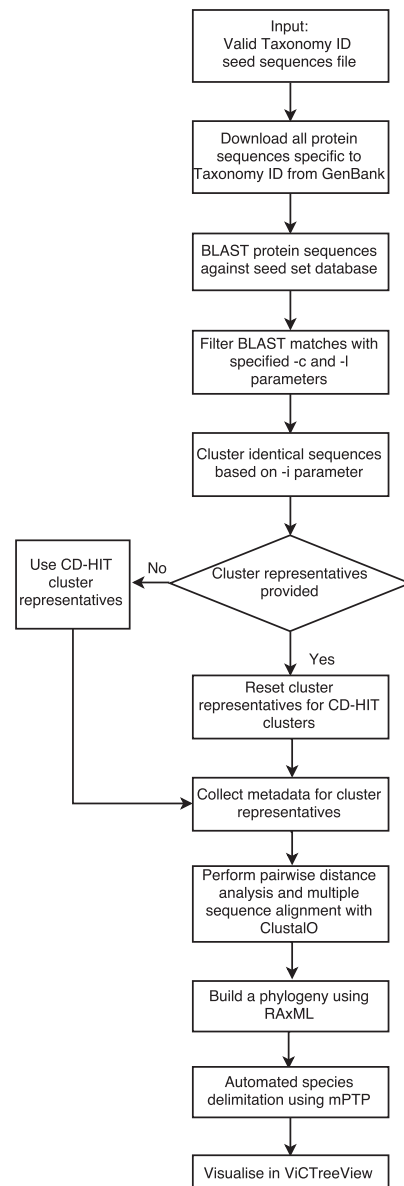


Fig. 1. Data processing workflow of the ViCTree pipeline

matches are extracted from the BLAST output on the basis of user-specified parameters specifying the hit length threshold (the minimum number of amino acid residues in the alignment between the query and subject sequences) and query coverage threshold (the minimum percentage of amino acid residues in the query sequence represented in the alignment). Significant matches are clustered by using CD-HIT (Fu *et al.*, 2012) with a user-specified identity threshold, and a representative sequence from each cluster is selected for downstream analysis in order to reduce the size of the tree to a manageable scale. CD-HIT is used to cluster the sequences below species level to generate a manageable sized phylogeny, with the default clustering threshold set to 0.9. CD-HIT picks the longest sequence as representative by default, and processes the classification of the remaining sequences by comparing them to the representatives. This arrangement has the advantage of not influencing the inter-species or inter-generic relationships. An optional parameter enables the user to provide a list of predefined sequences representing the clusters instead of the default sequences derived by using CD-HIT, thus allowing static tips to be maintained in an expanding phylogeny. A multiple sequence alignment and a distance matrix are generated for the final sequence set by using Clustal Omega (Sievers *et al.*, 2011), and an evolutionary tree is inferred by using RAxML (Stamatakis, 2014) under a user-defined evolutionary model or a default model (PROTGAMMAJTT). The evolutionary tree can then be submitted for automated species delimitation by mPTP (Kapli *et al.*, 2017).

The output files include the tree in Newick format, the alignment in Fasta format, a distance matrix, a comma-separated list of clustered sequences, and a metadata file with GenBank accession numbers and taxonomic names (species and genus) if known. These files are stored on GitHub in a predefined directory structure, thus allowing previous versions of the alignment and tree to be retrieved and enabling changes to be tracked over time. After an initial setup stage for the viral group of interest, the phylogeny can be updated with little or no manual intervention.

## 2.2 Tree visualization

ViCTreeView was inspired by VEG's phylotree (<https://github.com/veg/phylotree.js>) and enables the tree to be visualized. In ViCTreeView, maximum likelihood phylogenetic trees are rendered directly from the GitHub repository and visualized as a phylogram with bootstrap values. It is possible to visualize the tree in ultrametric representation instead of a phylogram. Different phylogenetic tree instances available in the GitHub repository can be browsed and visualized using by using the example menu. Attributes to increase and decrease distances between branches and a zooming function styled after Google maps are implemented in ViCTreeView, thus enabling users to explore specific parts of the phylogeny in detail. This interactive web tool facilitates an integrated dynamic visualization of the tree and the distance data represented as percentages. When a user-defined distance threshold is specified, sequences that fall within it are highlighted in clusters of different colours. This enables users to study and explore the sequences that generate new clusters when a specific pairwise distance criterion is applied. The highlighted versions of the tree with user-defined thresholds are also available for downloading in SVG and PNG formats. The automated phylogeny generated within ViCTree is midpoint rooted and can be re-rooted to any nodes in the phylogeny. Specific branches can be expanded and collapsed in order to explore large phylogenetic trees in modular fashion. In addition, various features are included to allow manipulation of the tree, including options for labelling the tips by GenBank accession number,

taxonomic ID, species name or genus name. These options, along with the alignment files from GitHub repositories, enable users to download the tree files in a specific format, and also facilitate easy incorporation of data for newly discovered viruses into taxonomic proposals. Links are also provided to the NCBI genomes page for representative sequences and to the NCBI proteins page for the representative and non-representative protein sequences clustered in the phylogeny.

ViCTree can be run on any Linux/Unix or OSX Apple computer. It was tested on an Apple iMac with a 4 GHz Intel Core i7 processor and 32 GB RAM.

## 3 Results

### 3.1 Case study

The example framework (<http://bioinformatics.cvr.ac.uk/victree/>) is setup for subfamilies *Densovirinae* and *Parvovirinae* of family *Parvoviridae* and for family *Herpesviridae*, and is updated monthly on an automatic schedule. To illustrate the application of ViCTree, we present the results for subfamily *Densovirinae*.

Within the family *Parvoviridae*, viruses that infect invertebrates and vertebrates are classified into the two subfamilies *Densovirinae* and *Parvovirinae*, respectively. This division is strongly supported by the protein sequence-based phylogeny of viral non-structural protein 1 (NS1) (Cotmore *et al.*, 2014). All viruses within the same species are required to be at least 85% identical to each other in this protein, and at least 15% different from viruses in other species. Viruses within the same genus are required to be monophyletic and to encode NS1 proteins that are at least 30% identical to each other. These demarcation criteria were applied to an analysis of subfamily *Densovirinae* carried out by using ViCTree. The analysis took 552 min 34.983 s real time, 1019 min 38.773 s user time and 29 min 32.492 s system time.

The analysis of subfamily *Densovirinae* runs automatically every month as a Cron job. In June 2017, 916 protein sequences available under the relevant taxonomic ID (40120) were downloaded automatically from GenBank. A subset of 21 NS1 protein sequences was used as the seed set in a BLAST-based similarity search of all downloaded protein sequences. A subset of 187 sequences was generated when hit length and query coverage thresholds of 100 and 50, respectively, were applied to filter the BLAST output. These sequences grouped into 103 distinct clusters when a CD-HIT clustering threshold of 1.0 was applied. Distance analysis and multiple sequence alignment were performed on the 103 representatives of these clusters, and taxonomic and accession metadata were collected from GenBank. A phylogeny was built for the aligned sequences, and the tree file was submitted with metadata and distance matrix files to ViCTreeView for visualization.

ViCTree identified all previously classified species and genera in subfamily *Densovirinae* (Cotmore *et al.*, 2014), including members of genus *Ambidensovirus* that encode the NS1 protein on the opposite strand from members of the other genera. This success was due to the breadth of diversity in the seed set and to the use of protein sequences in conducting the BLAST search. The analysis identified six new species (Table 1), which were recognized subsequently by the ICTV (Adams *et al.*, 2017b) on the basis of proposals made by the ICTV *Parvoviridae* Study Group. Although ViCTree is not a dedicated taxonomic misclassification identification tool such as SATIVA (Kozlov *et al.*, 2016) it is able to identify misclassified sequences. Thus, the misclassification of an isolate of Helicoverpa armigera densovirus (GenBank accession number JQ894784) in the

NCBI taxonomy was readily identified. This virus is described as being a member of genus *Iteradensovirus*, but in fact belongs to genus *Ambidensovirus* (species *Lepidopteran iteradensovirus* 5).

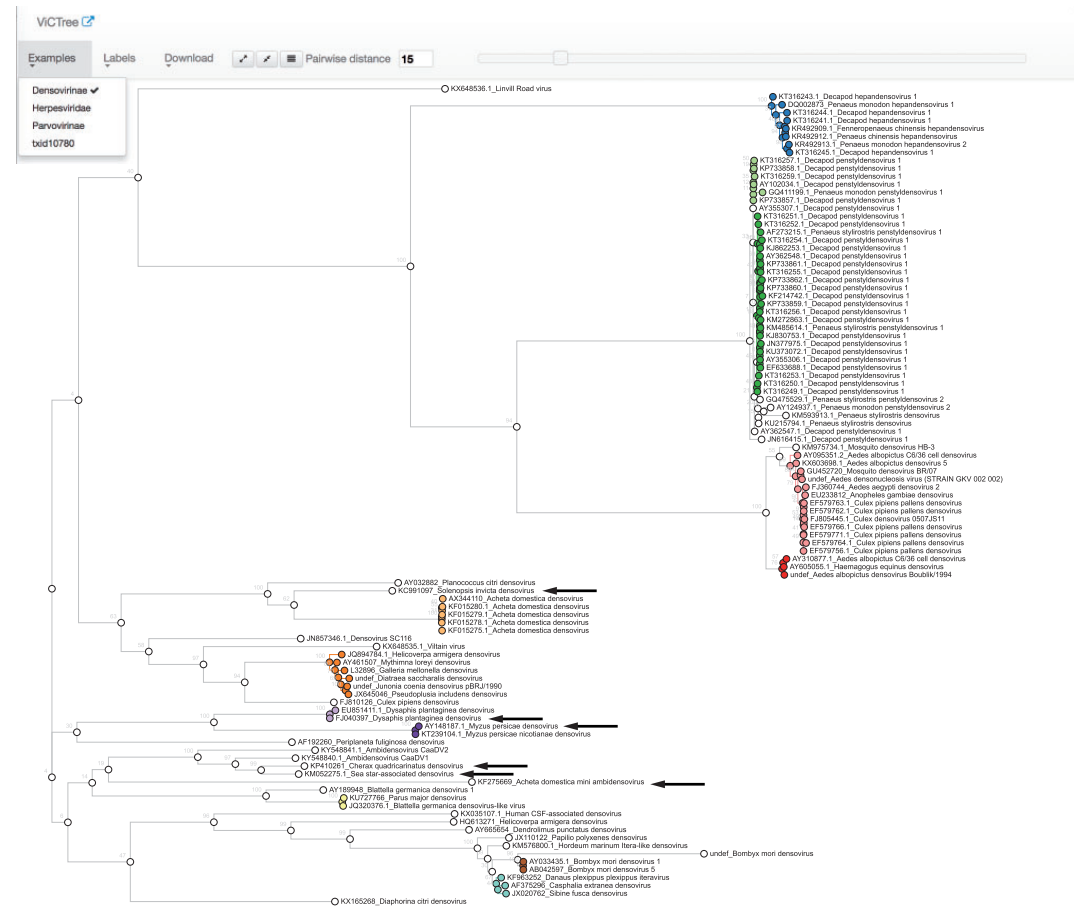
The in-built automated species delimitation using mPTP had identified a total of 25 species from the phylogeny, of which 18 were consistent with the current ICTV classification of subfamily Densovirinae. The automated species delimitation combined *Decapod ambidensovirus* 1 and *Asteroid ambidensovirus* 1 into a

species cluster, and *Lepidopteran iteradensovirus* 1, *Lepidopteran iteradensovirus* 2 and *Lepidopteran iteradensovirus* 4 into another species cluster. It also identified an additional six new species that are not currently recognized by the ICTV (Supplementary Appendix S1). Some of these species are not yet defined as new species by the ICTV as the sequences may be from incomplete genome sequences or may lack multiple sequences from the same species, both of which are requirements for the assignment of new species.

**Table 1.** New species identified in subfamily *Densovirinae* by using ViCTree

Name of new species	Representative isolate	Genus
<i>Asteroid ambidensovirus</i> 1	Sea star-associated densovirus	<i>Ambidensovirus</i>
<i>Decapod ambidensovirus</i> 1	Cherax quadricarinatus densovirus	<i>Ambidensovirus</i>
<i>Hemipteran ambidensovirus</i> 2	Dysaphis plantaginea densovirus 1	<i>Ambidensovirus</i>
<i>Hemipteran ambidensovirus</i> 3	Myzus persicae densovirus 1	<i>Ambidensovirus</i>
<i>Hymenopteran ambidensovirus</i> 1	Solenopsis invicta densovirus	<i>Ambidensovirus</i>
<i>Orthopteran densovirus</i> 1	Acheta domestica mini ambidensovirus	Unassigned

Source: [https://talk.ictvonline.org/ICTV/proposals/2016.003a, bD.A.v1.Densovirinae\\_6sp.pdf](https://talk.ictvonline.org/ICTV/proposals/2016.003a, bD.A.v1.Densovirinae_6sp.pdf)



**Fig. 2.** Phylogenetic tree for subfamily *Densovirinae* based on the NS1 protein and visualized in ViCTreeView. Sequences that fall within the 15% pairwise distance criterion are indicated as distinct clusters in different colours. Black arrows indicate new species identified using ViCTree

### 3.2 Evaluation of accuracy

The accuracy of the ViCTree was tested by determining the proportion of recognized species that it was capable of identifying in subfamily *Densovirinae* (Fig. 2). Three parameters were varied: the number of seed sequences (sets of 5, 10 or 20 randomly selected sequences), the hit length threshold, and the query coverage threshold (Fig. 3). Accuracy increased with the number of seed sequences, and was >95% for all seed sequence sets at a hit length of <400 and a query coverage of <60. Accuracy was compromised by reducing these values, due to increasing numbers of false positives. Hit length and query coverage thresholds of 100 and 50, respectively, were found to be optimal for subfamily *Densovirinae*.

## 4 Discussion

ViCTree is an integrated, automated pipeline for assisting taxonomic classification in an era in which genomic and metagenomic data are being actively accommodated by the ICTV (Adams *et al.*, 2017a; Simmonds, 2015; Simmonds *et al.*, 2017). It is capable of supporting the identification of novel viral species and pinpointing taxonomic errors in public databases. Its automated approach to finding the best reference sequences to represent a viral family or subfamily provides a useful tool for virologists. It implements GitHub-based versioning of alignments and phylogenies of any size,

thus allowing users to monitor taxonomic developments incrementally. The built-in visualization tool (ViCTreeView) enables phylogenies to be explored interactively in a web browser. These features will contribute to the establishment and dissemination of standardized phylogenetic and taxonomic data within the virology community.

The initial setup of ViCTree for a taxonomic group requires several optimization steps, which include setting the thresholds for seed sequences, CD-HIT clustering, and BLAST hit length and query coverage. These parameters were shown to be accurate in the case study of subfamily *Densovirinae*, but will need to be improved iteratively as the taxonomy expands. They will differ for other viral taxa; for example, a single DNA polymerase protein seed sequence was sufficient to identify all species in family *Herpesviridae*. In a wider context, the criteria used to classify viruses vary greatly from family to family, and the flexibility of ViCTree allows appropriate thresholds to be explored interactively. Accuracy determination for a specific viral group of interests is deemed to be an iterative process, as classification parameters depend on the new sequences identified and incorporated into the seed set used as a starting point for ViCTree analysis. The ViCTree GitHub repository provides scripts that enable users to identify optimal BLAST and seed set parameters to study a viral taxonomic group using ViCTree. Novel sequences that are yet to be submitted to GenBank can also be explored using

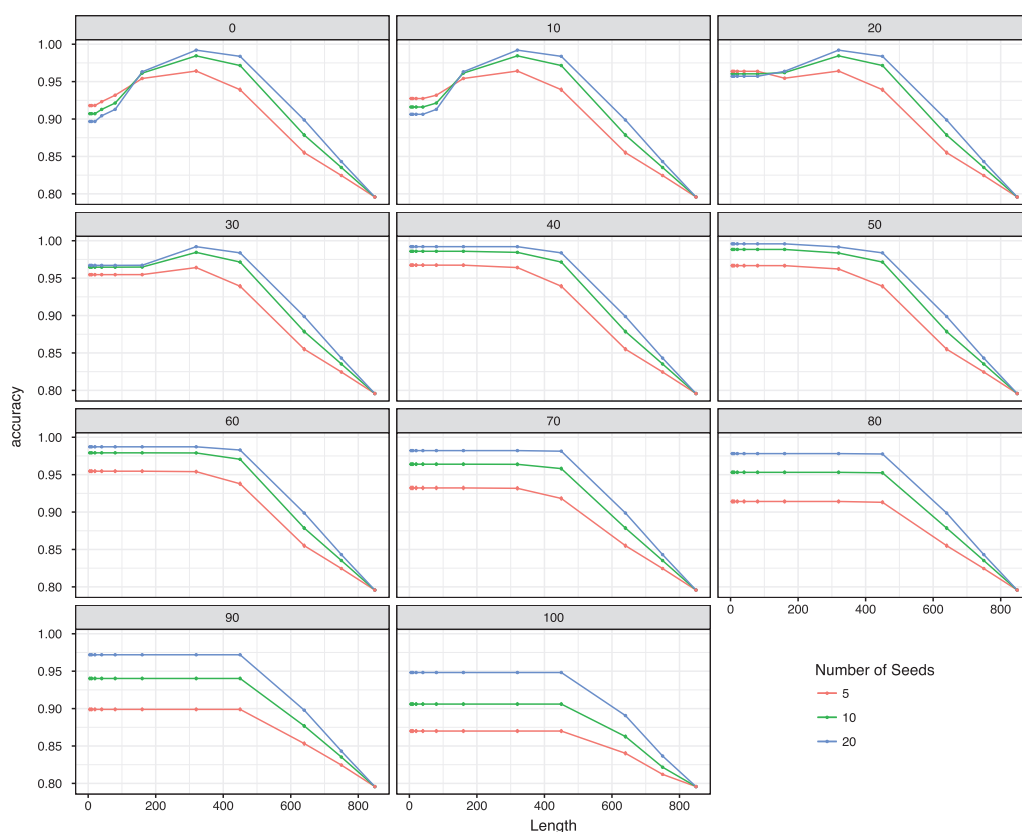


Fig. 3. Accuracy (y-axis) of ViCTree in relation to BLAST query coverage (0–100), BLAST hit length (0–849 amino acid residues) and number of seed sequences (5–20)

the ViCTree framework, as ViCTree allows the incorporation of these sequences by adding them to the seed sequence set.

Since ViCTree is a pipeline that integrates a number of existing tools, it will suffer from their limitations. Clustal Omega was incorporated because it can align large numbers of protein sequences quickly and accurately (Sievers *et al.*, 2011). However, like other progressive algorithms such as CLUSTAL W (Thompson *et al.*, 1994), MAFFT (Katoh *et al.*, 2002; Katoh and Standley, 2013), MUSCLE (Edgar, 2004) and T-COFFEE (Di Tommaso *et al.*, 2011), it may suffer from shortcomings in the handling of insertions and deletions. Other phylogeny-aware methods that have been developed for accurately aligning closely related sequences, such as PRANK (Löytynoja and Goldman, 2005, 2008), are less susceptible to these problems and may improve ViCTree in future, particularly for delimiting genotypes within viral species.

ViCTree adds to a growing number of sequence-based tools that are designed to inform viral classification specifically or that may prove to be adaptable from other areas of biology. Pairwise distance criteria currently being used across the field of viral taxonomy does not provide an objective approach to classify groups of viruses and other methods such as GMYC and PTP/mPTP should be explored further in the context of speciation and identification of novel viral groups. The current version of ViCTree uses protein sequences as input because amino acid sequences are typically used to distinguish taxa from the level of family (or sometimes order) down to species (e.g. *Parvoviridae* and *Herpesviridae*). Although ViCTree was developed with viral classification in mind, it could be used to explore the evolution of any protein.

## Acknowledgements

We thank Dr Quan Gu for testing ViCTree extensively, Dr Richard Orton for providing insightful advice on the manuscript, and other members of the CVR Viral Genomics and Bioinformatics team for offering useful comments.

## Funding

This work was supported by the Medical Research Council [grant number MC\_UU\_12014/12].

*Conflict of Interest:* none declared.

## References

- Adams,M.J. *et al.* (2017a) 50 years of the International Committee on Taxonomy of Viruses: progress and prospects. *Arch. Virol.*, **162**, 1441–1446.
- Adams,M.J. *et al.* (2017b) Changes to taxonomy and the International Code of Virus Classification and Nomenclature ratified by the International Committee on Taxonomy of Viruses (2017). *Arch. Virol.*, **162**, 2505–2538.
- Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
- Bao,Y. *et al.* (2014) Improvements to pairwise sequence comparison (PASC): a genome-based web tool for virus classification. *Arch. Virol.*, **159**, 3293–3304.
- Cotmore,S.F. *et al.* (2014) The family Parvoviridae. *Arch. Virol.*, **159**, 1239–1247.
- Edgar,R.C. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, **32**, 1792–1797.
- Fu,L. *et al.* (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**, 3150–3152.
- Hibbett,D.S. *et al.* (2005) Points of View Automated Phylogenetic Taxonomy: an Example in the Homobasidiomycetes (Mushroom-Forming Fungi). *Syst. Biol.*, **54**, 660–668.
- Izquierdo-Carrasco,F. *et al.* (2014) PUMPER: phylogenies updated perpetually. *Bioinformatics*, **30**, 1476–1477.
- Kapli,P. *et al.* (2017) Multi-rate Poisson Tree Processes for single-locus species delimitation under Maximum Likelihood and Markov Chain Monte Carlo. *Bioinformatics*, **33**, btx025.
- Katoh,K. *et al.* (2002) MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res.*, **30**, 3059–3066.
- Katoh,K. and Standley,D.M. (2013) MAFFT Multiple Sequence Alignment Software Version 7: improvements in Performance and Usability. *Mol. Biol. Evol.*, **30**, 772–780.
- Kozlov,A.M. *et al.* (2016) Phylogeny-aware identification and correction of taxonomically mislabeled sequences. *Nucleic Acids Res.*, **44**, 5022–5033.
- Lauber,C. and Gorbalenya,A.E. (2012) Partitioning the genetic diversity of a virus family: approach and evaluation through a case study of picornaviruses. *J. Virol.*, **86**, 3890–3904.
- Löytynoja,A. and Goldman,N. (2008) Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, **320**, 1632–1635.
- Löytynoja,A. and Goldman,N. (2005) An algorithm for progressive multiple alignment of sequences with insertions. *Proc. Natl. Acad. Sci. USA*, **102**, 10557–10562.
- Nishimura,Y. *et al.* (2017) ViPTree: the viral proteomic tree server. *Bioinformatics*, **33**, 2379–2380.
- Sievers,F. *et al.* (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.*, **7**, 539.
- Simmonds,P. *et al.* (2017) Consensus statement: virus taxonomy in the age of metagenomics. *Nat. Rev. Microbiol.*, **15**, 161–168.
- Simmonds,P. (2015) Methods for virus classification and the challenge of incorporating metagenomic sequence data. *J. Gen. Virol.*, **96**, 1193–1206.
- Smith,S.A. *et al.* (2009) Mega-phylogeny approach for comparative biology: an alternative to supertree and supermatrix approaches. *BMC Evol. Biol.*, **9**, 37.
- Stamatakis,A. (2014) RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, **30**, 1312–1313.
- Thompson,J.D. *et al.* (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, **22**, 4673–4680.
- Di Tommaso,P. *et al.* (2011) T-Coffee: a web server for the multiple sequence alignment of protein and RNA sequences using structural information and homology extension. *Nucleic Acids Res.*, **39**, W13–W17.
- Vilella,A.J. *et al.* (2009) EnsemblCompara GeneTrees: complete, duplication-aware phylogenetic trees in vertebrates. *Genome Res.*, **19**, 327–335.
- Wu,D. *et al.* (2008) An Automated Phylogenetic Tree-Based Small Subunit rRNA Taxonomy and Alignment Pipeline (STAP). *PLoS One*, **3**, e2566.

**transPLANT Resources for Triticeae  
Genomic Data**

**M. Spannagl, M. Alaux, M. Lange, D.  
M. Bolser, [and 25 others, including A.  
S. Thanki.]**

**The Plant Genome 2015**

---

## transPLANT Resources for Triticeae Genomic Data

Manuel Spannagl,\* Michael Alaux, Matthias Lange, Daniel M. Bolser, Kai C. Bader, Thomas Letellier, Erik Kimmel, Raphael Flores, Cyril Pommier, Arnaud Kerhornou, Brandon Walts, Thomas Nussbaumer, Christoph Grabmuller, Jinbo Chen, Christian Colmsee, Sebastian Beier, Martin Mascher, Thomas Schmutzer, Daniel Arend, Anil Thanki, Ricardo Ramirez-Gonzalez, Martin Ayling, Sarah Ayling, Mario Caccamo, Klaus F.X. Mayer, Uwe Scholz, Delphine Steinbach, Hadi Quesneville, and Paul J. Kersey

### Abstract

The genome sequences of many important Triticeae species, including bread wheat (*Triticum aestivum* L.) and barley (*Hordeum vulgare* L.), remained uncharacterized for a long time because their high repeat content, large sizes, and polyploidy. As a result of improvements in sequencing technologies and novel analyses strategies, several of these have recently been deciphered. These efforts have generated new insights into Triticeae biology and genome organization and have important implications for downstream usage by breeders, experimental biologists, and comparative genomicists. transPLANT (<http://www.transplantdb.eu>) is an EU-funded project aimed at constructing hardware, software, and data infrastructure for genome-scale research in the life sciences. Since the Triticeae data are intrinsically complex, heterogenous, and distributed, the transPLANT consortium has undertaken efforts to develop common data formats and tools that enable the exchange and integration of data from distributed resources. Here we present an overview of the individual Triticeae genome resources hosted by transPLANT partners, introduce the objectives of transPLANT, and outline common developments and interfaces supporting integrated data access.

**C**ROPS from the tribe of the Triticeae, including wheat, barley, and rye (*Secale cereale* L.), account for some of the most important nutritional resources in the human diet. Until recently, genomics-informed breeding approaches were limited in Triticeae species, as few genomic data were available. This lack can mainly be attributed to the inherent complexity of their genomes and genetics, especially in species of high economic interest such as barley and bread wheat. With estimated haploid and triploid sizes of 5.3 and 17.1 Gb, respectively, the genomes of these (and other Triticeae) species significantly exceed the size of the haploid human genome (~3 Gb). The high overall repeat content and, in bread wheat,

M. Spannagl, K.C. Bader, T. Nussbaumer, and K.F.X. Mayer, Plant Genome and Systems Biology (PGSB), Helmholtz Center Munich, D-85764, Neuherberg, Germany; M. Alaux, T. Letellier, E. Kimmel, R. Flores, C. Pommier, D. Steinbach, and H. Quesneville, INRA, UR1164 URGI-Research Unit in Genomics-Info, INRA de Versailles, Route de Saint-Cyr, Versailles, 78026, France; M. Lange, J. Chen, C. Colmsee, S. Beier, M. Mascher, T. Schmutzer, D. Arend, and U. Scholz, Leibniz Institute of Plant Genetics and Crop Plant Research (IPK) Gatersleben, Stadt Seeland, D-06466, Germany; D.M. Bolser, A. Kerhornou, B. Walts, C. Grabmuller, and P.J. Kersey, European Molecular Biology Lab., The European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK; A. Thanki, R. Ramirez-Gonzalez, M. Ayling, S. Ayling, and M. Caccamo, The Genome Analysis Centre (TGAC), Norwich Research Park, Norwich, NR4 7UH, UK. Received 8 June 2015. Accepted 14 Oct. 2015. \*Corresponding author ([manuel.spannagl@helmholtz-muenchen.de](mailto:manuel.spannagl@helmholtz-muenchen.de)).

**Abbreviations:** BAC, bacterial artificial chromosome; CSS, chromosome survey sequence; EBI, European Bioinformatics Institute; EST, expressed sequence tag; GO, gene ontology; IBSC, International Barley Genome Sequencing Consortium; IPK, Leibniz Institute of Plant Genetics and Crop Plant Research; IWGSC, International Wheat Genome Sequencing Consortium; PGSB, Plant Genome and Systems Biology unit at the Helmholtz Center Munich; QTL, quantitative trait loci; RNA-seq, RNA sequencing; SNP, single nucleotide polymorphism; TGAC, The Genome Analysis Centre; URGI, Unité de Recherche Génomique Info at the Institut National de la Recherche Agronomique; WheatIS, International Wheat Information System.

Published in The Plant Genome 9  
doi: 10.3835/plantgenome2015.06.0038

© Crop Science Society of America  
5585 Guilford Rd., Madison, WI 53711 USA  
This is an open access article distributed under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).



the allohexaploid genome structure, further complicate every aspect of the collation and analysis of genomic data, including sequencing, assembly, gene calling, and functional analysis.

Nevertheless, great progress has recently been made in deciphering the genome sequences and gene content of these species through coordinated international collaboration. The International Barley Genome Sequencing Consortium (IBSC) reported a draft genome sequence for the barley cultivar Morex obtained using mainly next-generation sequencing technologies, chromosome sorting, an integrated physical and genetic map, gene predictions, and analysis of the transcriptional landscape (IBSC, 2012). For bread wheat, 5× 454 whole-genome sequencing was used to generate assemblies of homeologous genes and to analyze the gene content of hexaploid wheat in a reference-directed approach (Brenchley et al., 2012). In 2014, the International Wheat Genome Sequencing Consortium (IWGSC) released draft genome sequence assemblies for all wheat chromosome arms (again, following the use of a physical sorting strategy before sequencing), with a total of 124,201 annotated gene models and supporting transcriptomics data (IWGSC, 2014). Meanwhile, a French-led consortium has generated a reference sequence and gene annotation for wheat chromosome 3B (Choulet et al., 2014). Genomic data for a number of additional Triticeae species has also been generated recently, including the bread wheat subgenome progenitors *Aegilops tauschii* Coss. (Jia et al., 2013), *A. sharonensis* Eig and *A. speltoides* Tausch (IWGSC, 2014), and *T. urartu* Tumanian ex Gandilyan (Ling et al., 2013), and the tetraploid *T. turgidum* L. *subsp. durum* (Desf.) Husn. (pasta wheat) (IWGSC, 2014). GenomeZippers, a synteny-enabled anchoring approach, have been constructed for the Triticeae species barley (Mayer et al., 2011), wheat (IWGSC, 2014), *A. tauschii* (Luo et al., 2013), and rye (Martis et al., 2013), facilitating the positioning of 10,000s of genes in the absence of finished genome sequences. While the sequence of all these species is incompletely assembled, significant progress has been made toward assembling the gene space, assigning contigs to chromosomes, and ordering them. The data is already sufficient to support analyses of gene families, variation within populations, large scale synteny, and association of genotype with phenotype.

The availability of genomic data from multiple Triticeae species is expected to facilitate powerful comparative genomics approaches and help to enhance understanding of Triticeae biology and evolution. However, the use of multiple approaches to genome sequencing and assembly (e.g., chromosome sorting, GenomeZippers, and genome survey sequencing), the variety of associated data types (e.g., gene predictions, expression data, and molecular markers), and the existence of alternative coordinate systems (e.g., genetic map, physical map, and numerical position in molecular sequence) can make it difficult for users to combine different data sets easily and correctly. Storage, integration, and visualization

of these heterogeneous and complex data are essential to enable efficient research.

Here we describe Triticeae genome data resources maintained by partners in the EU-funded transPLANT project. The transPLANT project aims at producing an integrated, coherent data infrastructure shared among dispersed expert resources with strong interconnections between them (including cross-linking and the use of common formats, tools, and datasets) designed to make it easy for users to switch between different resources according to which one best addresses their current point of interest.

## Results

### transPLANT

The transPLANT project (Table 1, reference no. 1 [Table 1.1]; hereafter, this format is used to reference Table 1) is an integrated infrastructure funded by the Framework 7 program of the European Union. It brings together 11 partners from seven countries with the aim of developing common standards, data, and technologies in the plant genomics area. A major focus of the work is variation data and the development of tools to organize, archive, and analyze this. Another major focus is the definition and use of common reference sequences so that annotation from different resources can be shared and compared. A third focus is the development of a distributed query infrastructure to provide a common point of entry to data held in multiple, dispersed resources.

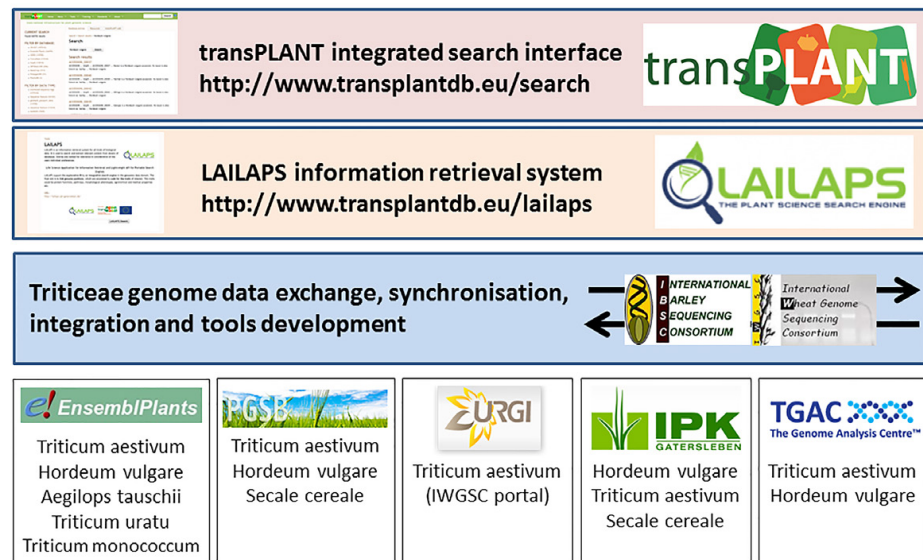
In the context of Triticeae data, five transPLANT partners (The Plant Genome and Systems Biology unit at the Helmholtz Center Munich [PGSB], the European Bioinformatics Institute [EBI], the Unité de Recherche Génomique Info at the Institut National de la Recherche Agronomique [URGI], the Leibniz Institute of Plant Genetics and Crop Plant Research [IPK] and The Genome Analysis Centre [TGAC]) are involved and interacting with other partners engaged in the international consortia (IWGSC and IBSC) coordinating the sequencing and assembly of these genomes. Figure 1 provides an overview over the respective Triticeae data resources hosted by transPLANT partners together with available species and shared search interfaces and services. Data from barley, wheat, and other species have been exported to common data formats in accordance with established standards for data representation, exchanged between partners, and synchronized across the distributed transPLANT resources. Moreover, partners have collaborated in comparative analysis of Triticeae genomes to study ancient duplications, polyploidy, and syntenic relationships.

As a result, end users benefit from a number of specialized tools and interfaces operating on synchronized Triticeae genome data hosted and exchanged by the transPLANT partners. This includes an interface embedded in the transPLANT web portal enabling keyword searches over the data inventories of many transPLANT partners as well as extensive cross-linking between



**Table 1. List of URLs for transPLANT Triticeae resources, tools, and websites. References to specific URLs are given in the format Table 1. REFERENCE no. (e.g. Table 1.1) throughout the text.**

Reference no.	Service provider	URL	Short description
1	transPLANT	<a href="http://www.transplantdb.eu">http://www.transplantdb.eu</a>	transPLANT web hub
2	transPLANT	<a href="http://www.transplantdb.eu/resource-registry">http://www.transplantdb.eu/resource-registry</a>	transPLANT resource registry
3	PGSB	<a href="http://pgsb.helmholtz-muenchen.de/plant/transplant/genomeResources.jsp">http://pgsb.helmholtz-muenchen.de/plant/transplant/genomeResources.jsp</a>	transPLANT resource registry mirror at PGSB
4	PGSB	<a href="http://pgsb.helmholtz-muenchen.de/plant/triticeae/index.jsp">http://pgsb.helmholtz-muenchen.de/plant/triticeae/index.jsp</a>	PlantsDB Triticeae homepage
5	PGSB	<a href="http://pgsb.helmholtz-muenchen.de/plant/crowsNest/index.jsp">http://pgsb.helmholtz-muenchen.de/plant/crowsNest/index.jsp</a>	PlantsDB CrowsNest tool
6	EBI	<a href="http://plants.ensembl.org">http://plants.ensembl.org</a>	Ensembl Plants homepage
7	INRA	<a href="http://wheat-urgi.versailles.inra.fr/">http://wheat-urgi.versailles.inra.fr/</a>	INRA URGI wheat database
8	INRA	<a href="http://wheat-urgi.versailles.inra.fr/Seq-Repository">http://wheat-urgi.versailles.inra.fr/Seq-Repository</a>	IWGSC wheat sequence repository
9	INRA	<a href="https://urgi.versailles.inra.fr/gb2/gbrowse/wheat_phys_pub">https://urgi.versailles.inra.fr/gb2/gbrowse/wheat_phys_pub</a>	Wheat physical maps browser
10	INRA	<a href="https://urgi.versailles.inra.fr/blast/">https://urgi.versailles.inra.fr/blast/</a>	URGI wheat BLAST search tool
11	INRA	<a href="http://urgi.versailles.inra.fr/Wheat3BMine/">http://urgi.versailles.inra.fr/Wheat3BMine/</a>	Wheat 3B data warehouse
12	IPK	<a href="http://lailaps.ipk-gatersleben.de">http://lailaps.ipk-gatersleben.de</a>	LAILAPS search engine
13	IPK	<a href="http://webblast.ipk-gatersleben.de/barley">http://webblast.ipk-gatersleben.de/barley</a>	IPK barley BLAST server
14	IPK	<a href="http://barlex.barleysequence.org">http://barlex.barleysequence.org</a>	Barlex home page
15	TGAC	<a href="http://www.tgac.ac.uk/tools-resources">www.tgac.ac.uk/tools-resources</a>	TGAC tools and resources homepage
16	TGAC	<a href="http://polymarker.tgac.ac.uk">http://polymarker.tgac.ac.uk</a>	TGAC Polymarker
17	TGAC	<a href="http://www.tgac.ac.uk/grassroots-genomics">http://www.tgac.ac.uk/grassroots-genomics</a>	TGAC grassroots genomics website
18	transPLANT	<a href="http://www.transplantdb.eu/training-resources">http://www.transplantdb.eu/training-resources</a>	transPLANT user training resources
19	transPLANT	<a href="http://www.transplantdb.eu/videos">http://www.transplantdb.eu/videos</a>	transPLANT user training videos
20	WheatIS	<a href="http://www.wheatinitiative.org/tools/wheat">http://www.wheatinitiative.org/tools/wheat</a>	Wheat Initiative website
21	WheatIS	<a href="http://www.wheatis.org">www.wheatis.org</a>	Wheat Information System homepage
22	PGSB	<a href="http://pgsb.helmholtz-muenchen.de/plant/plantsdb.jsp">http://pgsb.helmholtz-muenchen.de/plant/plantsdb.jsp</a>	PlantsDB entry page
23	PGSB	<a href="http://pgsb.helmholtz-muenchen.de/plant/barley/gz/tablejsp/index.jsp">http://pgsb.helmholtz-muenchen.de/plant/barley/gz/tablejsp/index.jsp</a>	PlantsDB GenomeZipper view
24	TGAC	<a href="http://browser.tgac.ac.uk/wheat/">http://browser.tgac.ac.uk/wheat/</a>	TGAC wheat browser
25	TGAC	<a href="http://browser.tgac.ac.uk/barley_phys/">http://browser.tgac.ac.uk/barley_phys/</a>	TGAC physical map browser
26	TGAC	<a href="http://browser.tgac.ac.uk/wheat_compara/">http://browser.tgac.ac.uk/wheat_compara/</a>	TGAC Aequatus browser



**Figure 1. Overview of Triticeae genome resources and services provided by transPLANT partners. Species for which data is available in the respective database systems are given under the resource names. Details on data types and tools and modes of access are given in the individual partner sections. Upper panels illustrate both the transPLANT integrated search tool (enabling centralized keyword searches over the data inventories of many transPLANT partners) as well as the LAILAPS search engine (linking plant genomic data to phenotypic traits).**

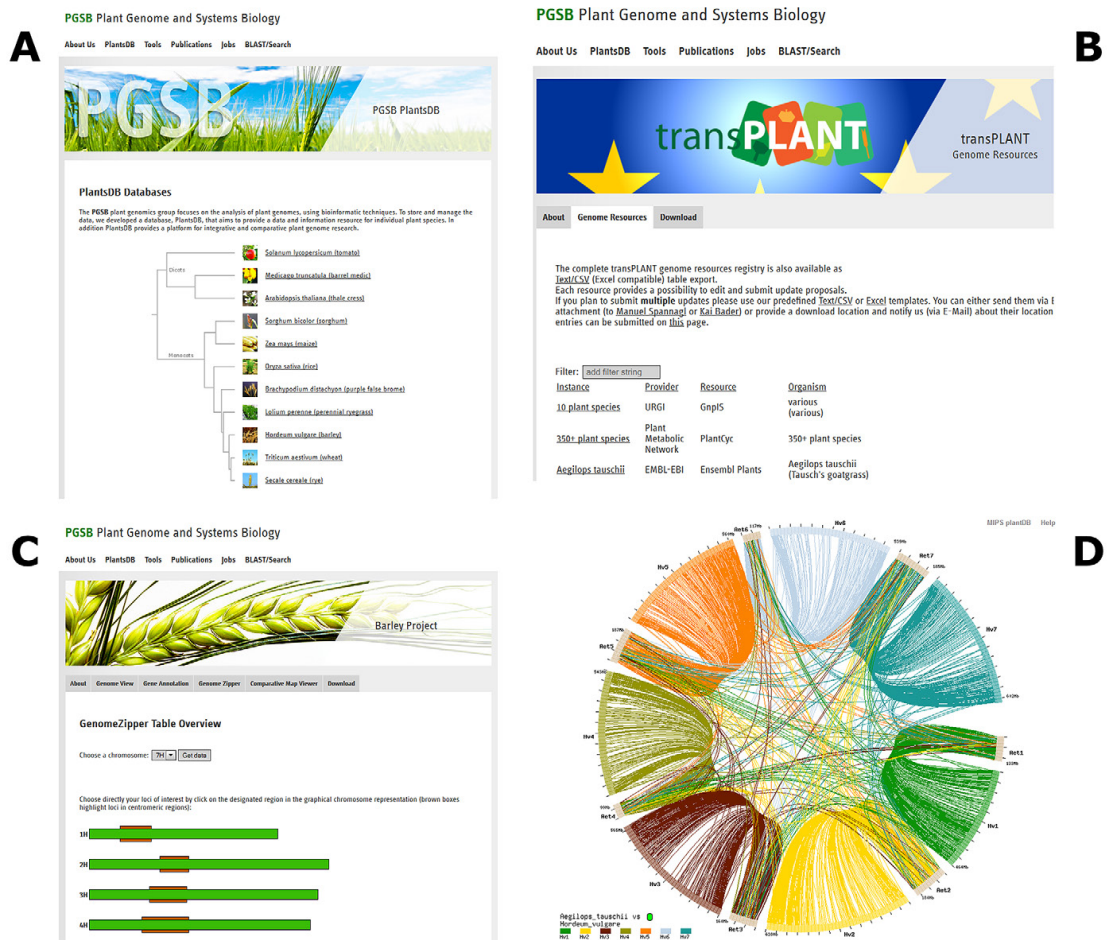


Figure 2. Triticaceae and transPLANT resources hosted by PGSB PlantsDB. (A) Database overview and PlantsDB entry page (Table 1.22). (B) transPLANT genome resources registry giving access to more than 300 different plant genome resources and databases (Table 1.3). (C) GenomeZipper overview representation for the barley genome (Table 1.23). (D) Visualization of syntenic relationships between the genome sequences of *Aegilops tauschii* and barley within the CrowsNest tool (Table 1.5).

partner resources and the provision of web services and data warehouses. To assist users and communities in identifying data critical for their research, transPLANT provides a registry for genome resources (not restricted to Triticeae) and all data is discoverable within a single, integrated search. The registry can be accessed at the websites indicated in Table 1.2 and 1.3 (see also Fig. 2B). The integrated search is available from the website indicated in Table 1.1.

### Plant Genome and Systems Biology PlantsDB: A Platform for the Comparative Analysis of Triticeae Genome Data

Plant Genome and Systems Biology (PGSB, formerly MIPS) PlantsDB provides a platform for the integration, visualization, and comparative analysis of plant genome

data. Currently, genome data from 12 different plant species are available in the public domain of PGSB PlantsDB (Fig. 2A). Access points include search and browse interfaces, BioMoby webservices (Wilkinson et al., 2005), FTP download server, as well as visualization tools. With ongoing involvement in Triticeae genome sequencing and annotation initiatives such as IBSC (IBSC, 2012) and IWGSC (IWGSC, 2014), a major focus of PlantsDB was put on the representation and analysis of complex Triticeae genome data.

To compensate for the lack of reference chromosome sequences, GenomeZippers were constructed for many Triticeae species including barley, wheat, and rye. The GenomeZipper concept integrates data from chromosome sorting, second generation sequencing, array hybridization, and systematic exploitation of conserved

synteny with model grasses to construct virtual ordered gene maps (Mayer et al., 2009, 2011). Besides batch download via FTP, all GenomeZipper data has been integrated into the PGSB PlantsDB database scheme. To assist the interactive exploration of GenomeZippers and the search for anchored elements such as expressed sequence tags (ESTs), genetic markers, and full-length complementary DNA, interfaces for querying and browsing the GenomeZippers for barley, wheat, and rye were constructed (accessible from Table 1.4; Fig. 2C).

Between the genomes of many monocotyledonous plants, including major Triticeae crops and model plants, gene order appears to be conserved over long chromosomal stretches (synteny). This facilitates the potential transfer of knowledge from model plants such as *Brachypodium distachyon* (L.) Beauv. or rice (*Oryza sativa* L.) to the more complex Triticeae crops such as barley and wheat. To assist interactive exploration and visualization of syntenic regions and genes between grass models and Triticeae crop species, the CrowsNest tool (Table 1.5) was developed and populated with the genomes of rice, sorghum [*Sorghum bicolor* (L.) Moench], *B. distachyon*, barley, and *A. tauschii*, the diploid progenitor of the D sub-genome of hexaploid bread wheat. Figure 2D shows a screenshot from the CrowsNest tool visualizing synteny between *A. tauschii* and barley on a whole-genome scale.

Wheat genomic subassembly sequences generated in a reference-directed approach (Brenchley et al., 2012) were integrated with their corresponding genes from *B. distachyon*, sorghum, rice, and barley. Interfaces to query this data include a BLAST server to search for homologous wheat sequences as well as search for reference genes and associated wheat sequences. Genes predicted from chromosome-sorted wheat genome sequence generated within the IWGSC (IWGSC, 2014) have been integrated into PGSB PlantsDB and cross-referenced with the corresponding repositories GnpIS Wheat and Ensembl Plants.

To visualize and search the integrated barley physical and genetic maps, dedicated instances of GBrowse and CrowsNest were set up and in cooperation with IPK Gatersleben populated with markers, bacterial artificial chromosome (BAC) end sequences, BAC sequences, and physical map information. Gene expression data from barley (IBSC, 2012) has been integrated into the RNASeq-ExpressionBrowser (Nussbaumer et al., 2014) and can be queried by keyword, sequence similarity search (BLAST), or gene ontology (GO) and Interpro term and domain.

### Triticeae Genome Data in Ensembl Plants

Ensembl Plants (Table 1.6) is an integrative web portal for plant genomic data, developed by the EBI. The portal provides interactive and programmatic access to data from 39 species through a variety of interfaces including web browser, Perl and RESTful Application Programming Interfaces, FTP, a publicly accessible relational database server, and a data-mining tool implemented using the data-warehousing framework BioMart optimized for gene and variant-centric queries. In addition to participating

in transPLANT, coordination with efforts in the United States is achieved through a formal collaboration with the Gramene project (<http://www.gramene.org>).

Currently, four Triticeae genomes (among 20 cereal genomes) are available in Ensembl Plants: bread wheat, two of its diploid progenitors, *A. tauschii* and *T. urartu*, and barley. In the case of both wheat and barley; additional information (from genetic and physical maps) has been used to assign the genomic contigs to chromosomes and locate them within them. For barley, many of the contiguous sequences generated and assembled by the IBSC have been binned into located clusters according to evidence from the genetic and physical maps, and this information is used to construct a chromosome level view in Ensembl. The initial assembly has recently been revised using POPSEQ (Mascher et al., 2013a) data generated by the IPK. Whole-genome alignments have been performed against *B. distachyon*, rice, bread wheat, and the bread wheat progenitor genomes; collections of barley and wheat ESTs and RNA-sequencing (RNA-seq) reads have been aligned to the barley reference. In addition, intervarietal single nucleotide polymorphisms (SNPs) are represented for eleven varieties of barley as well as sites of variation between wild barley (*H. spontaneum*) and the barley reference.

The core wheat data represented is the chromosome survey sequence (CSS) generated, assembled, and annotated by the IWGSC. However, the CSS assembly of chromosome 3B has been replaced by the BAC-by-BAC assembly constructed by Choulet et al. (2014). In addition to sequence assemblies and gene models, a number of additional data sets have been aligned to the survey sequence, including the complete genomes of *B. distachyon* and rice, wheat unigene clusters from NCBI, and wheat RNA-seq data deposited in the International Nucleotide Sequence Database Collaboration archives. The wheat genome assemblies previously generated by Brenchley et al. (2012) have also been aligned to the survey sequence *B. distachyon* and barley. In addition, a collection of 900,000 polymorphisms from CerealsDB (<http://www.cerealsdb.uk.net>) have been included in the resource as well as data from the wheat HapMap project (Jordan et al., 2015). The chromosome survey sequence (and its annotations), in addition to the complete EST set, are available to search via BLAST and other search alignment algorithms.

For all gene models in Ensembl Plants, functional annotation is inferred using GO, InterPro, and homology metrics. Additionally, the evolutionary history of each protein-coding gene is inferred from comparisons to other plant species, and gene trees and protein alignments are available to browse (see Fig. 3) and download. The three bread wheat genomes have additionally been aligned to each other and linked to assertions of homeology derived from the gene tree analysis to provide supporting evidence. These alignments have been used to identify sites of variation (single nucleotide variants and insertion-deletions) between the A, B, and D genomes (see Fig. 4).

In addition, transcriptome data from another bread wheat precursor species, *T. monococcum* (Fox et al., 2014), have been aligned to the hexaploid reference.

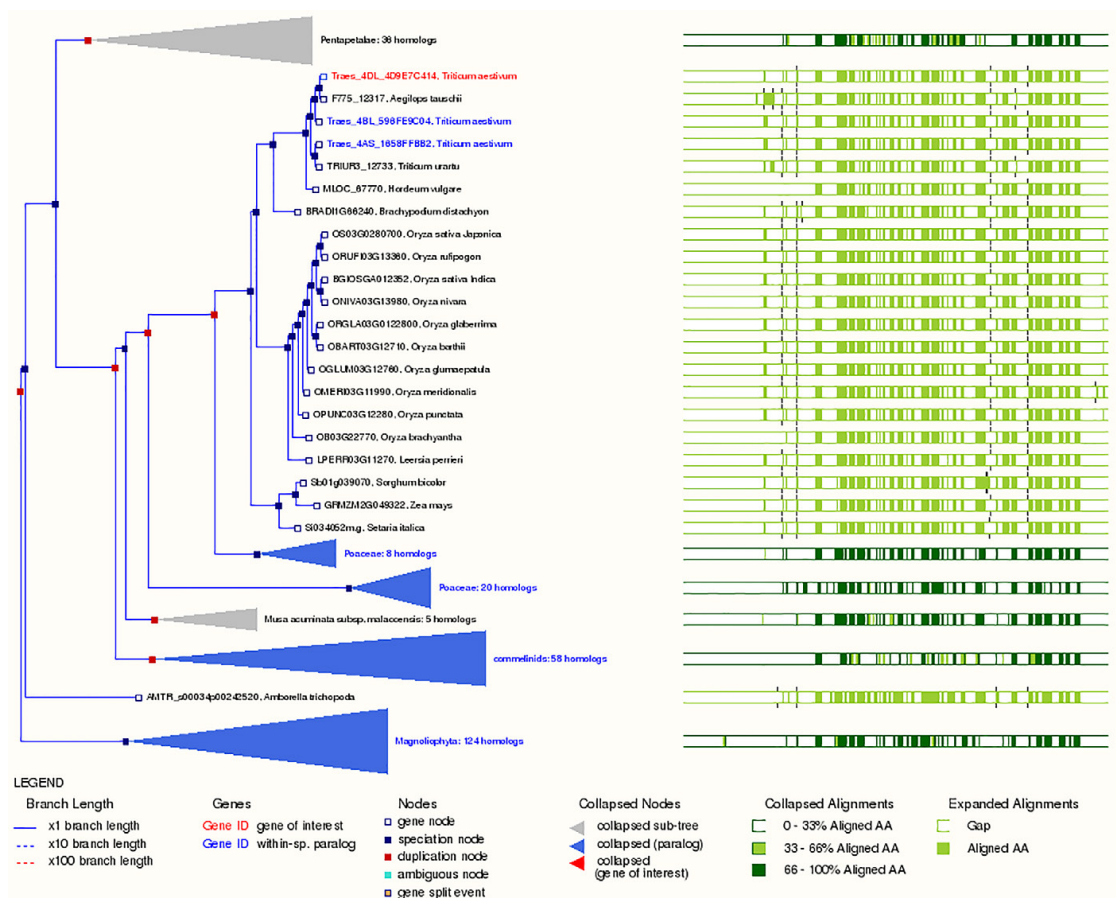


Figure 3. A gene family conserved with 1:1 orthology over 21 *Poaceae* genomes as visualized in Ensembl Plants. The three bread wheat genes are highlighted in red and blue, and the chromosome (and subgenome) are indicated in the prefix of the gene name (e.g., *Traes\_4AL\_X* is the name of a gene from the long arm of chromosome 4 in the A genome). The most related genes are those of the bread wheat precursors, followed by barley. Genes from more distant *Poaceae* species are located below.

## Triticeae Genome Data and Tools at the Unité de Recherches en Génomique Info

### The Official Wheat International Wheat Genome Sequencing Consortium Portal

Unité de Recherches en Génomique Info has been chosen by the IWGSC to be the repository for wheat genomic sequences and physical maps (Table 1.7).

To allow users to download, display, and query the IWGSC sequences and physical map data, a section dedicated to wheat genomics data, the sequence repository (Table 1.8; Fig. 5A), has been set up. Data stored in the sequence repository includes the wheat survey sequence, the chromosome reference sequence (chromosome 3B), the genes and annotations (gene models, GenomeZipper, and POPSEQ), the physical maps, the RNA-Seq, and the variations (HapMap) data.

Users can display the sequence annotation of the 3B reference sequence and the survey sequence in dedicated browsers. The physical maps browser (Table 1.9; Fig. 5B)

is a customized instance of the GBrowse tool developed by the GMod community (Stein et al., 2002).

The *T. aestivum* sequence data, diploid, and tetraploid wheat species sequence data (e.g., *T. durum*, *T. monococcum*, *T. urartu*, *A. speltoides*, *A. sharonensis*, and *A. tauschii*) are searchable using a BLAST tool (Table 1.10). The BLAST server is a customized version of the ViroBLAST tool developed by the University of Washington (Deng et al., 2007).

It also hosts the supplementary data attached to IWGSC publications and we are currently developing a page dedicated to assist the upcoming reference chromosome assemblies.

### Data Warehouse for Wheat Chromosome 3B

To be able to connect reference sequence data from chromosome 3B (Choulet et al., 2014) with genetics and phenomics data, the Wheat3BMine data warehouse (Table 1.11; Fig. 5C) was developed in the framework of transPLANT.



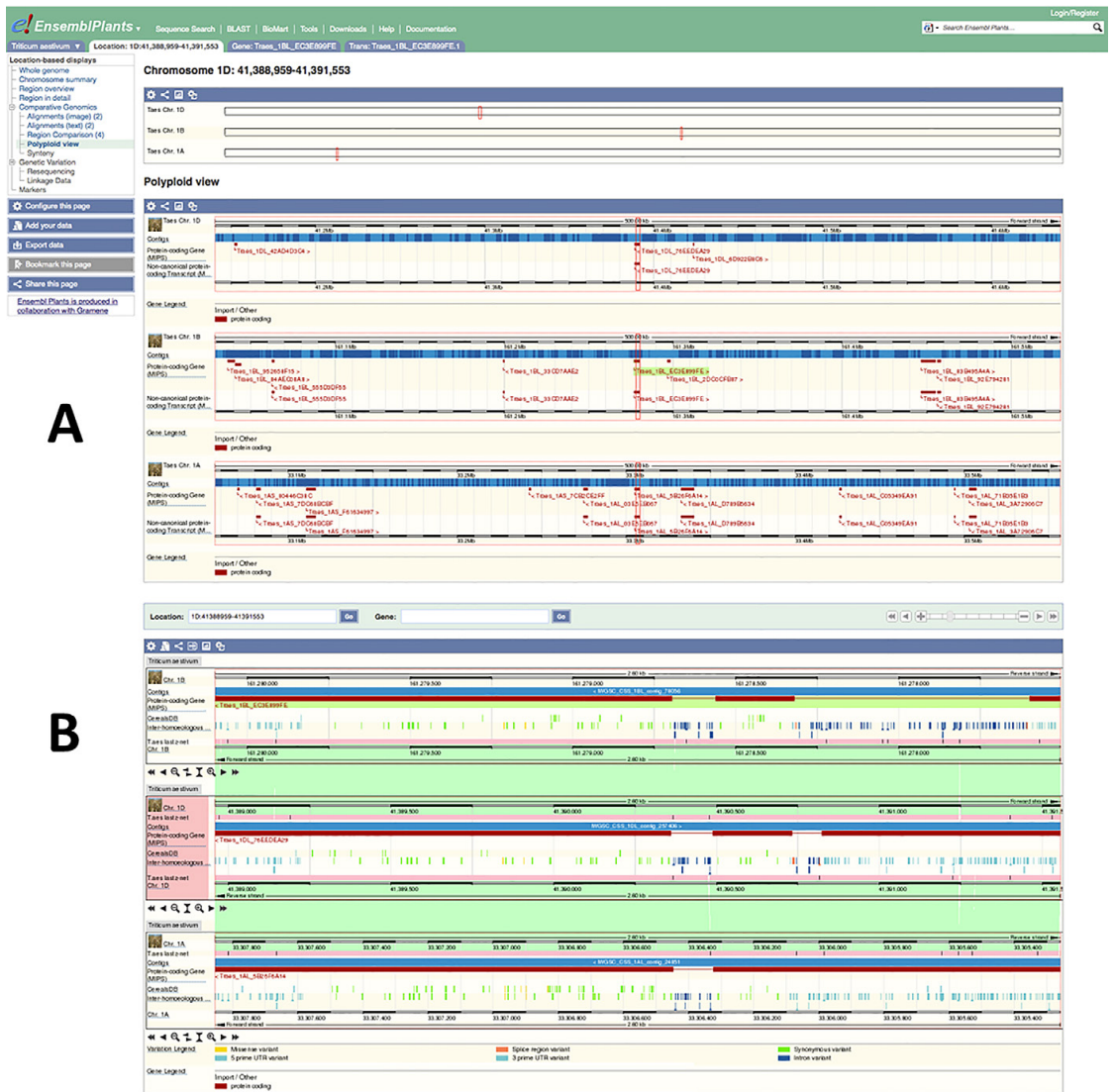


Figure 4. Homeologous regions from the bread wheat A, B, and D genomes as visualized in Ensembl Plants. (A) The top panel shows the annotations made on three regions of contiguous sequence. (B) The lower panel is centered on the homeologous genes and shows gene structures, intervarietal polymorphisms and interhomeologous variants.

The warehouse is implemented using InterMine technology that provides a fast, flexible, and user friendly access to integrated data by multiple ways: a browser, a query builder, and a region search tool. Wheat3BMine users can filter their favorite features, save their own queries, and export results in many different formats (GFF3, BED, or XML). An online documentation and precomputed queries are also available.

The data warehouse contains heterogeneous data and is gene centric. In fact, the typical gene card centralizes relevant information like gene function, ontology terms, and overlapping features. Wheat3BMine provides access to

genomic annotation data (genes, mRNA, polypeptides, and transposable elements), polymorphisms data (markers), genetic mapping data (quantitative trait loci [QTL], meta-QTL), and phenotyping data. Moreover, useful links are available from a gene card to the wheat 3B genome browser (Choulet et al., 2014) and to additional details in GnpIS.

#### Wheat Data in the GnpIS Information System

GnpIS (Steinbach et al., 2013) is an information system that integrates genomic and genetic data for plants and fungi. A “wheat” filter was implemented within GnpIS that allows interconnecting the wheat genomic data

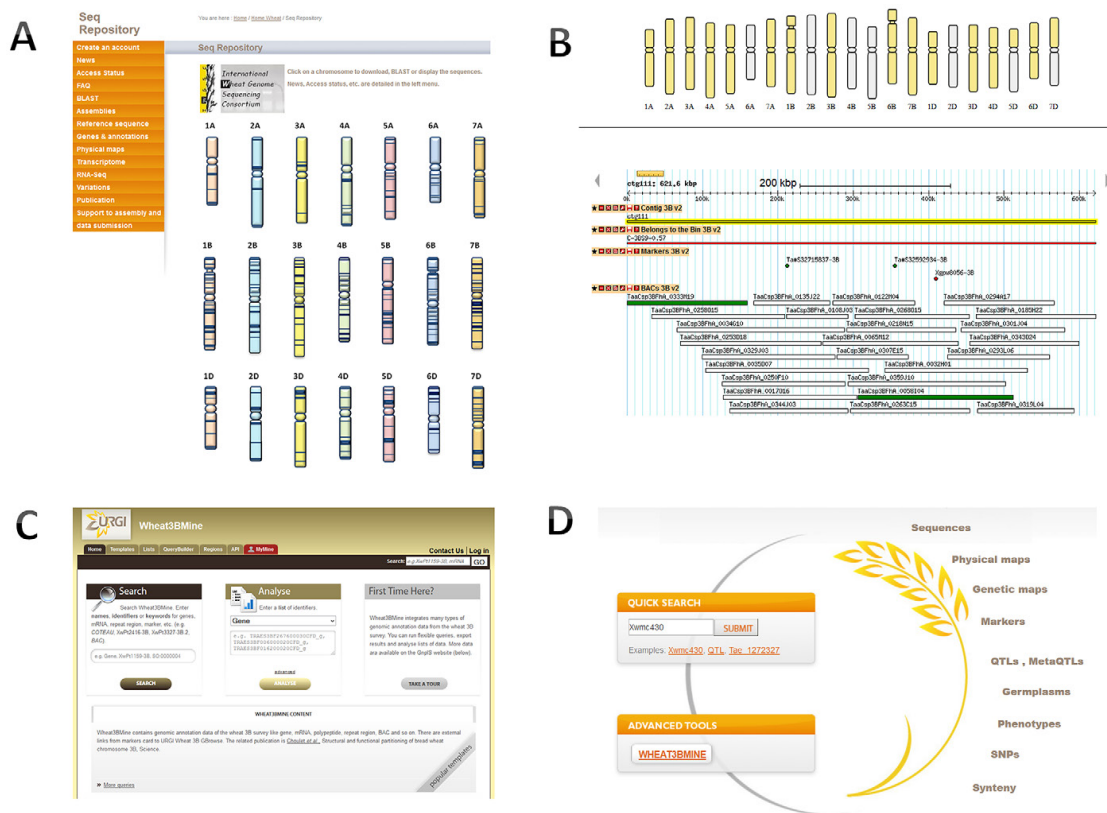


Figure 5. Triticeae resources hosted by URGI. (A) Wheat@URGI website: IWGSC Sequence Repository page (Table 1.8). (B) Wheat physical maps browser (Table 1.9). (C) Wheat3BMine tool homepage (Table 1.11). (D) Wheat data search in GnpIS: quick search, advanced tool and dedicated web interfaces (Table 1.7).

detailed above with the germplasm, markers, QTLs, SNPs, expression, and phenotypes data. Moreover, association and genomic selection data are in the process of integration into the information system. The wheat data in GnpIS (Table 1.7; Fig. 5D) can be queried using the quick search tool (Google-like search), advanced search tool (Wheat3BMine), and the dedicated web interfaces developed in Java and Google Web Toolkit.

### Triticeae Genome Data and Tools at the Leibniz Institute of Plant Genetics and Crop Plant Research Gatersleben

#### LAILAPS Search Engine

LAILAPS (Esch et al., 2014) is an integrated information retrieval system to link plant genomic data in the context of phenotypic traits for a detailed forward genetic research (Table 1.12; Fig. 6). LAILAPS is developed in the framework of the transPLANT project and allows exploratory search for candidate genes linked to specific traits over a loosely integrated system of indexed and interlinked genome databases. Query assistance and an

evidence-based annotation system enable time-efficient and comprehensive information retrieval. An artificial neural network incorporating user feedback and behavior tracking allows relevance sorting of results. Because this enhanced relevance ranking is one of the major innovations to explore millions of database records, a special focus has been set to its training and the inclusion of user feedback. The current LAILAPS release comprises about 91 million indexed database records of trait knowledge within 13 major life science data collections and more than 60 million associations to -omics data sets. To provide an up-to-date user ergonometry, the front end features an interactive query assistance that suggests spelling correction as well as semantic query expansion. This feature makes use of PubMed abstracts to learn vectors of similar words and phrases. Queries are expressed as keyword or phrases that are spell corrected. As query results, a condensed list of relevant hits is rendered that includes a short excerpt of relevant text positions and a list of annotation links to annotated genes in plant genomes. A comprehensive result filter panel and the suggestion of semantic follow-up queries rounds off the

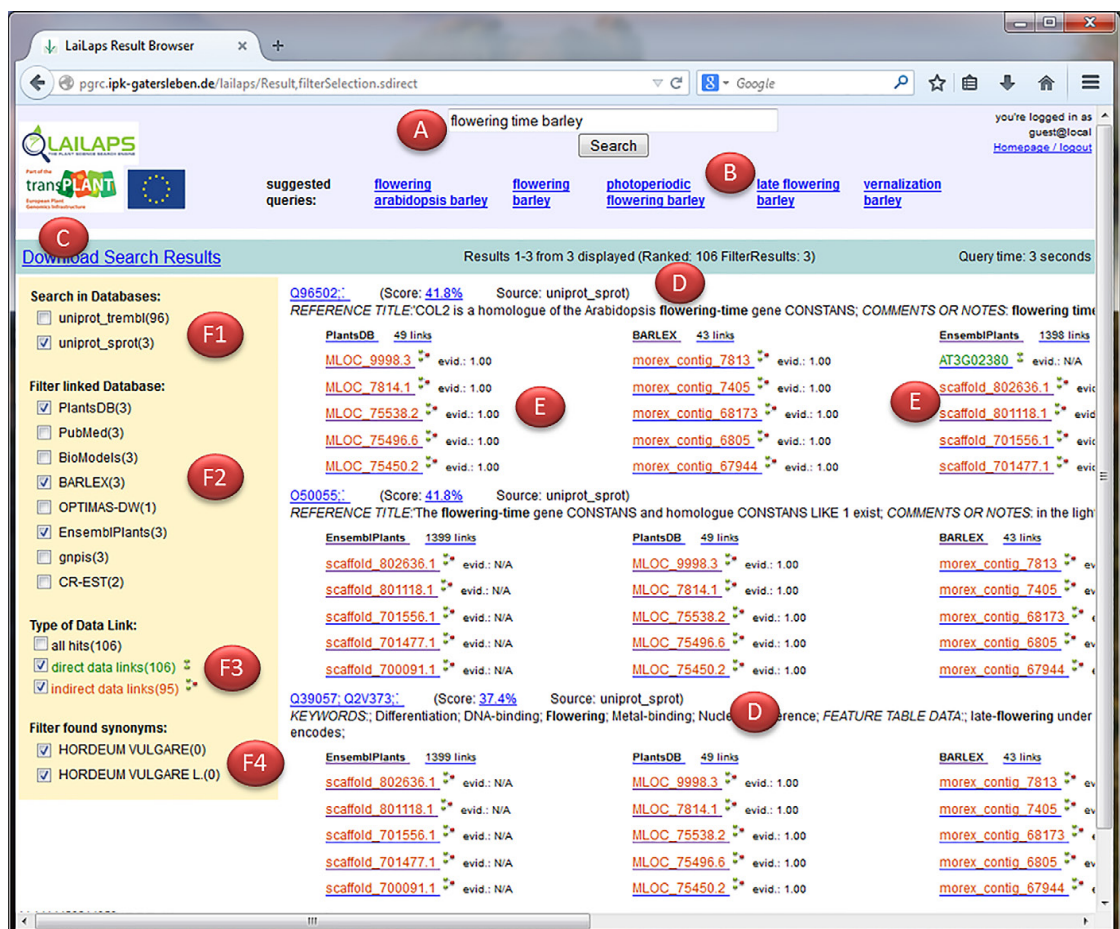


Figure 6. The LAILAPS web interface, illustrating the search and results page. The text box (A) enables an interactive and spelling corrected keyword query submission. After query was executed successfully, a list of semantically related phrases is provided for query refinement (B). The query results can be either downloaded as a Microsoft Excel sheet (C) or interactively explored. For this, all relevant hits are displayed as short excerpts in the result panel (D). Connected to each hit is a list of links to associated genomic data (E). Those links can either refer to genome data directly (green) or reflect an indirect, transitive relationship (red). The left hand filter panel enables to restrict the results by fact databases (F1), linked genome databases (F2), direct or indirect linked gene annotations (F3), or synonyms (F4).

query refinement features. Once the user identifies interesting records, he is guided to the most relevant genomic dataset and is able to rate its quality. In turn, this enables LAILAPS relevance prediction system to improve the relevance prediction network.

### Barley BLAST Server

The IPK hosts a BLAST server (Table 1.13), providing homology searches against the complete barley sequence data sets published in IBSC (2012). This comprises the whole-genome shotgun assemblies of the cultivars Morex, Barke, and Bowman as well as the high- and low-confidence gene sets. The latest POPSEQ anchoring data (Mascher et al., 2013a) was also integrated in addition to the barley exome capture targets (Mascher et al., 2013b).

### BARLEX

The recent progress in sequencing and mapping technology has facilitated the construction of advanced genomics resources in species with large and complex crop genomes like barley. During such genome sequencing projects, the integration of large volumes of diverse information and data from disparate sources is an open issue. Existing genome browsers are not well adapted to this task, as they expect all genomic features to be anchored to a single linearly ordered reference sequence. The IPK provides the barley genome explorer, BARLEX (Colmsee et al., 2015), as a central unified repository for the genomic resources of barley. BARLEX is centered on the genome-wide physical map of barley and links it to an annotated whole-genome shotgun assembly and dense genetic

maps. A web-based interface presents data in tabular and graphical format and associates all information and published sequence data with shotgun assemblies, repeat annotations using KMasker (see Schmutzer et al., 2014), physical contigs, and annotated genes. A novel graph-based visualization strategy was implemented to show overlaps between adjacent BACs based on fingerprint and sequence data. BARLEX is publicly accessible at the website listed in Table 1.14 and is directly connected to the IPK Barley BLAST server as well as the LAILAPS system.

### *e!DAL: Plant Genomics and Phenomics Research Data Repository*

The IPK is hosting a plant genomics and phenomics research data repository, which is based on the e!DAL data sharing and publication infrastructure (Arend et al., 2014). It features the publication of plant research data that is out of scope of existing domain databases, too huge, or less structured. In compliance to international standards, such as DOI, DataCite, and OpenAIRE, plant genomic and phenotypic datasets are published. In a particular focus are studies of plant genetic resources from the system plant from the root to bloom and seed, as well from sequence analysis to systems biology. Examples are genomic datasets of Triticeae (DOIs: 10.5447/IPK/2015/0, 10.5447/IPK/2015/1, and 10.5447/IPK/2015/2).

### *Triticeae Genome Data and Tools at The Genome Analysis Centre*

The TGAC Browser is an open-source genomic browser developed to visualize genome annotations such as genes, variations, and markers for species whose reference sequence may be contiguous or highly fragmented; the IBSC's barley genome and the wheat CSS represent two such fragmented references (Fig. 7A, 7B). Traditional datatypes, such as the reference assembly and gene annotations, reside in an Ensembl schema database. Larger datasets are stored in standard file formats such as SAM, BAM, bigwig, and VCF. The TGAC Browser has an integrated BLAST functionality, essential for identifying and accessing regions of interest in fragmented genomes, and an interface to enable manual annotation of genomic features. Homeologous genes can also be explored through the Aequatus browser (Fig. 7C).

Through collaboration with CerealsDB, the TGAC wheat browser displays the mapped SNP markers from the 90K iSelect and Axiom arrays against the IWGSC chromosome survey sequence contigs. The TGAC Browsers are also available for the barley genome and barley physical map, which display the minimum tile path and BAC-end sequences. All browsers can be accessed at Table 1.15.

To tackle the issue of marker design for polyploid genomes, we have developed PolyMarker, an automated bioinformatics pipeline for SNP assay development that increases the probability of generating homeologue-specific assays for polyploid wheat (Ramirez-Gonzalez et al., 2015). PolyMarker (Fig. 7D) generates a multiple

alignment between the target SNP sequence and the IWGSC chromosome survey sequences (IWGSC, 2014) for each of the three wheat genomes. It then generates a mask with informative positions, which are highlighted with respect to the target genome allowing homeologue-specific primer design. The PolyMarker site (Table 1.16) provides predesigned primers for the iSelect 90K chip and 820K Axiom markers.

For more information and community support for these resources TGAC hosts the Grassroots genomics website (Table 1.17).

## **Discussion**

### **Outlook**

Great progress has been made recently in sequencing, annotating, and analyzing the complex genomes of Triticeae species including bread wheat and barley. The data generated has great potential for applications in plant breeding, experimental plant biology, and comparative genomics. However, to make the best possible use of the large and heterogeneous data sets produced, data archiving, integration, visualization, and access are essential. A variety of platforms provide different types of analysis and presentations, but it can be difficult for users to use these resources in combination. Many of the partners within the transPLANT project are actively involved in past or ongoing Triticeae genome initiatives and the development of resources. A major focus of the project is ensuring interoperability to maximize their collective value.

Critical to this is the development of common standards and formats. transPLANT partners use accepted standards to share and disseminate data wherever possible and are involved in ongoing efforts to standardize plant phenotypic data and metadata (that is, the description of material, experimental conditions, and results [Krajewski et al., 2015]). We have also been working with common data mining interfaces (BioMart [Smedley et al., 2015], InterMine [Smith et al., 2012]), and developing RESTful web services to support integrative programmatic access to data. We have also been developing cloud computing environments to support downstream analyses. This has translated into a number of concrete benefits for end users working with complex Triticeae genome data, including the following:

- Extensive and standardized data exchange and synchronization between partners; all data is served on common reference sequence, and portable annotation tracks can be visualized at different sites
- Data retrieval tool at the transPLANT web hub, indexing multiple types of Triticeae genome data (e.g., ESTs, genes, transcripts, phenotypes, and accessions) from all transPLANT partners
- LAILAPS integrated search engine, linking various Triticeae genomic data from transPLANT partners in the context of phenotypic traits



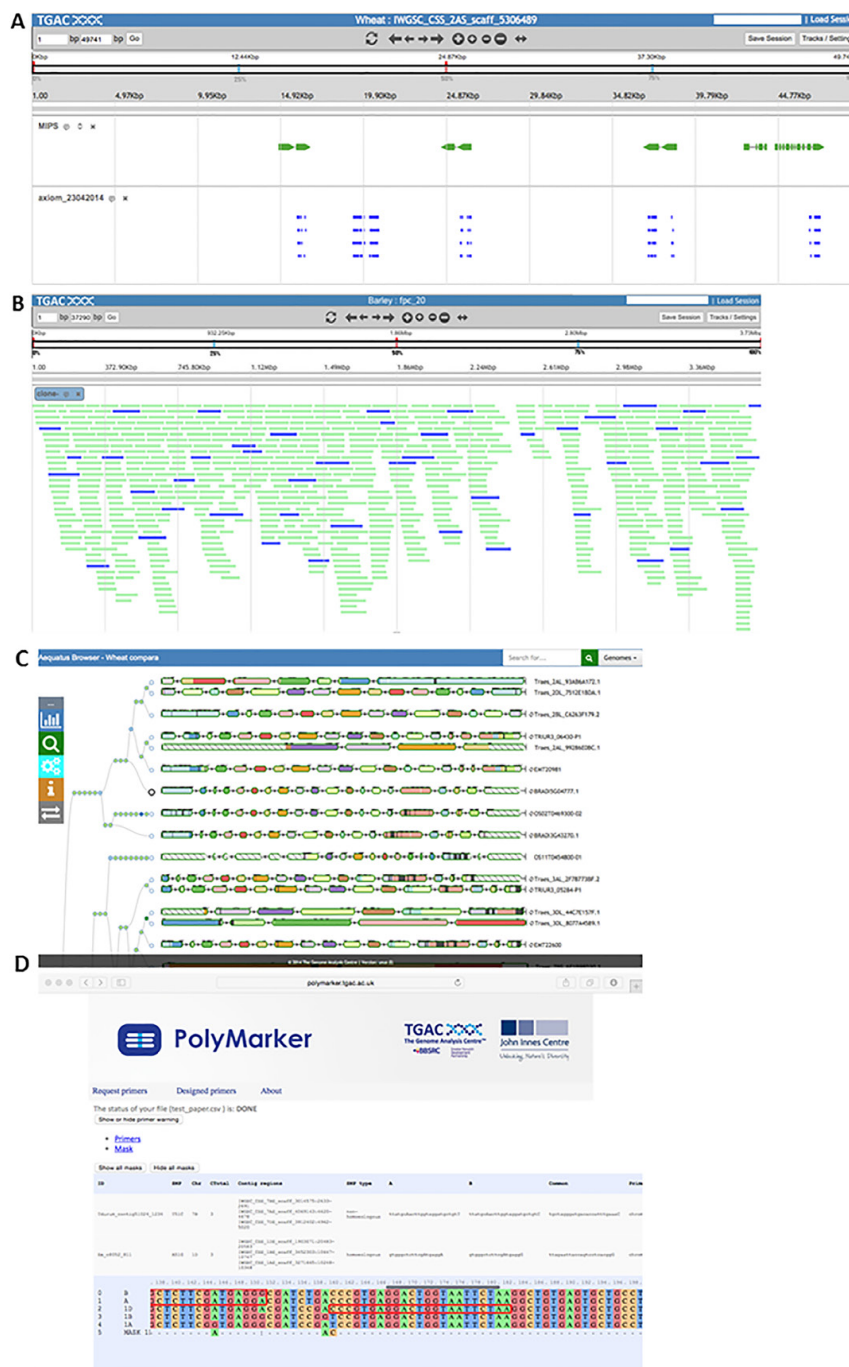


Figure 7. The Genome Analysis Centre (TGAC) wheat genome resources. (A) Wheat TGAC Browser: TGAC browser (Table 1.24) showing *Triticum aestivum* scaffold IWGSC\_2AS\_scaff\_5306489, with SNPs from 820K Axiom array and gene annotations from MIPS/PGSB. (B) TGAC Browser Barley Physical Map: TGAC browser showing the barley physical map for fingerprint (FPC) contig 20 (Table 1.25). MTP BACs are highlighted in blue. (C) Aequatus Browser (Table 1.26) showing *Brachypodium distachyon* gene BRADI5G04777.1 and homologous genes from *Triticum aestivum*, *Aegilops tauschii*, *Oryza sativa*, and *Triticum uratu*. (D) Primers designed by PolyMarker (Table 1.16). The webpage highlights the designed primers for validation that can be downloaded in spreadsheet formats.

- Extensive cross-linking between Triticeae genome resources and tools
- Provision of programmatic access to databases via APIs or web services
- Comprehensive online user training material available for download, covering both resource usage and wider analytical approaches, developed during a series of hands-on workshops (Table 1.18)
- Integrated training videos providing an overview of available resources (Table 1.19)

The result of these efforts is a rich collection of interfaces using common reference data, searchable through a single entry point at the central transPLANT web hub. They also simplify the identification of suitable datasets and databases for research using Triticeae genome data, assist in data acquisition, and provide powerful tools to analyze data in the context of other plant species. The training videos provide example use cases illustrating how users can take advantage of different resources in combination to interrogate the data and perform complex analyses.

With the expected emergence of additional genome sequence data from the Triticeae, the framework for data integration, exchange, and aggregation established within the transPLANT project will help to address the challenges involved with even more distributed data repositories and heterogeneous data types. In this context, a project has started recently with the objective to set up an International Wheat Information System (WheatIS) to support the wheat research community. The main objective is to provide a single-entry web-based system to access the available data resources and bioinformatics tools. The WheatIS project is an international project lead by an Export Working Group of the Wheat Initiative (Table 1.20). The Wheat Initiative is supported by the G20 Agricultural Ministers to coordinate worldwide research efforts in the fields of wheat genetics, genomics, physiology, breeding and agronomy.

The WheatIS project is driven by a network of 21 experts from Australia, Canada, France, Germany, Mexico, United States, and United Kingdom that congregates a group of volunteers willing to participate in the WheatIS project. The WheatIS (Table 1.21) will operate as a hub integrating wheat data produced and submitted to the public repositories by the community, extending the model and technologies established in transPLANT for the coordination of dispersed resources.

### Acknowledgments

All authors and institutions would like to acknowledge funding of the transPLANT project by the European Commission within its 7th Framework Programme, under the thematic area Infrastructures, contract number 283496. EBI acknowledges funding from the United Kingdom Biotechnology and Biological Sciences Research Council grants BB/I008071/1 and BB/I00328X/1, and grant 52930112 from the United States National Science Foundation. The IPK resources Barley Blast Server, BARLEX, and Kmasker eDAL were supported by the Leibniz Association (WGL) in the context of the *Pakt für Forschung und Innovation*/WGL and the German Federal Ministry of Education and Research

(BMBF) in the frame of the projects BARLEX (FKZ 0314000A), and TRITEX (FKZ 0315954A) and DPPN (FKZ 031A053B). URGI likes to acknowledge funding from INRA, french Research National Agency (ANR-09-GENM-025-003) 3BSEQ project, Investment for the Future (ANR-10-BTBR-03, France AgriMer, FSOV) BreedWheat project, European commission within 7th Framework Program TriticeaeGenome (KBBE-212019) and WHEALBI (FP7-613556) projects. TGAC likes to acknowledge funding from the Biotechnology and Biological Sciences Research Council grants BB/L002124/1 and BB/L024144/1. RRG is supported by a Norwich Research Park PhD Studentship and The Genome Analysis Centre Funding and Maintenance Grant. PGSB acknowledges funding from the German Federal Ministry of Education and Research (BMBF) in the frame of the projects BARLEX (FKZ 0314000A), and TRITEX (FKZ 0315954A) as well as Deutsche Forschungsgemeinschaft (DFG) funding to project SFB924 Molecular mechanisms regulating yield and yield stability in plants.

### References

- Arend, D., M. Lange, J. Chen, C. Colmsee, S. Flemming, D. Hecht, et al. 2014. eDAL: A framework to store, share and publish research data. *BMC bioinformatics* 15: 214. doi:10.1186/1471-2105-15-214
- Brenchley, R., M. Spannagl, M. Pfeifer, G.L. Barker, R. D'Amore, A.M. Allen, et al. 2012. Analysis of the bread wheat genome using whole-genome shotgun sequencing. *Nature* 491:705–710. doi:10.1038/nature11650
- Choulet, F., A. Alberti, S. Theil, N. Glover, V. Barbe, J. Daron, et al. 2014. Structural and functional partitioning of bread wheat chromosome 3B. *Science* 345:1249721. doi:10.1126/science.1249721
- Colmsee, C., S. Beier, A. Himmelbach, T. Schmutzer, N. Stein, U. Scholz, et al. 2015. BARLEX: The Barley Draft Genome Explorer. *Mol. Plant* 8:964–966. doi:10.1016/j.molp.2015.03.009
- Deng, W., D.C. Nickle, G.H. Learn, B. Maust, and J.I. Mullins. 2007. ViroBLAST: A stand-alone BLAST web server for flexible queries of multiple databases and user's datasets. *Bioinformatics* 23:2334–2336. doi:10.1093/bioinformatics/btm331
- Esch, M., J. Chen, C. Colmsee, M. Klapperstuck, E. Grafarend-Belau, U. Scholz, et al. 2014. LAILAPS: The plant science search engine. *Plant Cell Physiol.* 56:e8. doi:10.1093/pcp/pcu185
- Fox, S.E., M. Geniza, M. Hanumappa, S. Naithani, C. Sullivan, J. Preece, et al. 2014. De novo transcriptome assembly and analyses of gene expression during photomorphogenesis in diploid wheat *Triticum monococcum*. *PLoS ONE* 9:E96855. doi:10.1371/journal.pone.0096855
- International Barley Genome Sequencing Consortium. 2012. A physical, genetic and functional sequence assembly of the barley genome. *Nature* 491:711–716. doi:10.1038/nature11543
- International Wheat Genome Sequencing Consortium. 2014. A chromosome-based draft sequence of the hexaploid bread wheat (*Triticum aestivum*) genome. *Science* 345:1251788. doi:10.1126/science.1251788
- Jia, J., S. Zhao, X. Kong, Y. Li, G. Zhao, W. He, et al. 2013. *Aegilops tauschii* draft genome sequence reveals a gene repertoire for wheat adaptation. *Nature* 496:91–95. doi:10.1038/nature12028
- Jordan, K.W., S. Wang, Y. Lun, L.J. Gardiner, R. MacLachlan, P. Hucl, et al. 2015. A haplotype map of allohexaploid wheat reveals distinct patterns of selection on homoeologous genomes. *Genome Biol* 16:48. doi:10.1186/s13059-015-0606-4
- Krajewski, P., D. Chen, H. Cwiek, A.D. van Dijk, F. Fiorani, P. Kersey, et al. 2015. Towards recommendations for metadata and data handling in plant phenotyping. *J. Exp. Bot.* 66:5417–5427. doi:10.1093/jxb/erv271
- Ling, H.Q., S. Zhao, D. Liu, J. Wang, H. Sun, C. Zhang, et al. 2013. Draft genome of the wheat A-genome progenitor *Triticum urartu*. *Nature* 496:87–90. doi:10.1038/nature11997
- Luo, M.C., Y.Q. Gu, F.M. You, K.R. Deal, Y. Ma, Y. Hu, et al. 2013. A 4-giga-base physical map unlocks the structure and evolution of the complex genome of *Aegilops tauschii*, the wheat D-genome progenitor. *Proc. Natl. Acad. Sci. USA* 110:7940–7945. doi:10.1073/pnas.1219082110
- Martis, M.M., R. Zhou, G. Haseneyer, T. Schmutzer, J. Vrana, M. Kubalaková, et al. 2013. Reticulate evolution of the rye genome. *Plant Cell* 25:3685–3698. doi:10.1105/tpc.113.114553
- Mascher, M., G.J. Muehlbauer, D.S. Rokhsar, J. Chapman, J. Schmutz, K. Barry, et al. 2013a. Anchoring and ordering NGS contig assemblies

- by population sequencing (POPSEQ). *Plant J.* 76:718–727. doi:10.1111/tpj.12319
- Mascher, M., T.A. Richmond, D.J. Gerhardt, A. Himmelbach, L. Clissold, D. Sampath, et al. 2013b. Barley whole exome capture: A tool for genomic research in the genus *Hordeum* and beyond. *Plant J.* 76:494–505. doi:10.1111/tpj.12294. doi:10.1111/tpj.12294
- Mayer, K.F., M. Martis, P.E. Hedley, H. Simkova, H. Liu, J.A. Morris, et al. 2011. Unlocking the barley genome by chromosomal and comparative genomics. *Plant Cell* 23:1249–1263. doi:10.1105/tpc.110.082537
- Mayer, K.F., S. Taudien, M. Martis, H. Simkova, P. Suchankova, H. Gundlach, et al. 2009. Gene content and virtual gene order of barley chromosome 1H. *Plant Physiol.* 151:496–505. doi:10.1104/pp.109.142612
- Nussbaumer, T., K.G. Kugler, K.C. Bader, S. Sharma, M. Seidel, and K.F. Mayer. 2014. RNASeqExpressionBrowser: A web interface to browse and visualize high-throughput expression data. *Bioinformatics* 30:2519–2520. doi:10.1093/bioinformatics/btu334
- Ramirez-Gonzalez, R.H., C. Uauy, and M. Caccamo. 2015. PolyMarker: A fast polyploid primer design pipeline. *Bioinformatics* 31:2038–2039. doi:10.1093/bioinformatics/btv069
- Schmutzer, T., L. Ma, N. Pousarebani, F. Bull, N. Stein, A. Houben, et al. 2014. Kmasker: A tool for in silico prediction of single-copy FISH probes for the large-genome species *Hordeum vulgare*. *Cytogenet. Genome Res.* 142:66–78. doi:10.1159/000356460
- Smedley, D., S. Haider, S. Durinck, L. Pandini, P. Provero, J. Allen, et al. 2015. The BioMart community portal: An innovative alternative to large, centralized data repositories. *Nucleic Acids Res.* 43:W589–W598. doi:10.1093/nar/gkv350
- Smith, R.N., J. Aleksic, D. Butano, A. Carr, S. Contrino, F. Hu, et al. 2012. InterMine: A flexible data warehouse system for the integration and analysis of heterogeneous biological data. *Bioinformatics* 28:3163–3165. doi:10.1093/bioinformatics/bts577
- Stein, L.D., C. Mungall, S. Shu, M. Caudy, M. Mangone, A. Day, et al. 2002. The generic genome browser: A building block for a model organism system database. *Genome Res.* 12:1599–1610. doi:10.1101/gr.403602
- Steinbach, D., M. Alaux, J. Amselem, N. Choisne, S. Durand, R. Flores, et al. 2013. GnpIS: An information system to integrate genetic and genomic data from plants and fungi. *Database* 2013:Bat058 doi:10.1093/database/bat058
- Wilkinson, M., H. Schoof, R. Ernst, and D. Haase. 2005. BioMOBY successfully integrates distributed heterogeneous bioinformatics Web Services. The PlaNet exemplar case. *Plant Physiol.* 138:5–17. doi:10.1104/pp.104.059170