Methods in Molecular Biology 2512

Springer Protocols

Alejandro Pereira-Santana
Samuel David Gamboa-Tuz
Luis Carlos Rodríguez-Zapata *Editors*

Plant Comparative Genomics

Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor
John M. Walker
School of Life and Medical Sciences
University of Hertfordshire
Hatfield, Hertfordshire, UK

For further volumes:
http://www.springer.com/series/7651

For over 35 years, biological scientists have come to rely on the research protocols and methodologies in the critically acclaimed *Methods in Molecular Biology* series. The series was the first to introduce the step-by-step protocols approach that has become the standard in all biomedical protocol publishing. Each protocol is provided in readily-reproducible step-by-step fashion, opening with an introductory overview, a list of the materials and reagents needed to complete the experiment, and followed by a detailed procedure that is supported with a helpful notes section offering tips and tricks of the trade as well as troubleshooting advice. These hallmark features were introduced by series editor Dr. John Walker and constitute the key ingredient in each and every volume of the *Methods in Molecular Biology* series. Tested and trusted, comprehensive and reliable, all protocols from the series are indexed in PubMed.

Plant Comparative Genomics

Edited by

Alejandro Pereira-Santana

Unidad de Biotecnología Industrial, Centro de Investigación y Asistencia en Tecnología y Diseño del Estado de Jalisco, A.C., Zapopan, Jalisco, Mexico

Samuel David Gamboa-Tuz

Unidad de Biotecnología, Centro de Investigación Científica de Yucatán, Mérida, Yucatán, Mexico

Luis Carlos Rodríguez-Zapata

Unidad de Biotecnología, Centro de Investigación Científica de Yucatán, Mérida, Yucatán, Mexico

Humana Press

Editors
Alejandro Pereira-Santana
Unidad de Biotecnología Industrial
Centro de Investigación y Asistencia
en Tecnología y Diseño del Estado
de Jalisco, A.C.
Zapopan, Jalisco, Mexico

Samuel David Gamboa-Tuz
Unidad de Biotecnología
Centro de Investigación Científica de Yucatán
Mérida, Yucatán, Mexico

Luis Carlos Rodríguez-Zapata
Unidad de Biotecnología
Centro de Investigación Científica de Yucatán
Mérida, Yucatán, Mexico

ISSN 1064-3745 ISSN 1940-6029 (electronic)
Methods in Molecular Biology
ISBN 978-1-0716-2428-9 ISBN 978-1-0716-2429-6 (eBook)
https://doi.org/10.1007/978-1-0716-2429-6

Preface

Similarities, differences, and peculiarities. That is what Comparative Genomics looks for. Today, with highways opened by next-generation sequencing technologies (NGS), enormous amounts of freely available genomic information are generated every year; we could say, every day. Because of that huge amount of genomic information, new tools and strategies are necessary to manage, process, and visualize a lot of information that allows us to find new clues about hidden features in a vast world of words: The "AGCT" world.

This book aims to bring to public light and update some recent methodologies for the task of inspecting the genomic world, extracting valuable information, and presenting it in a "human readable way." This book also serves as a guide to those researchers (advanced undergraduate students, postgraduate students, postdoctoral researchers, early researchers) that want to know how to implement comparative tools to explore their genomic data for their daily scientific work. Much of this book is mainly centered on bioinformatic tools due to their relevance and applications nowadays.

This book does not pretend to be an exhaustive guide covering all comparative genomic areas and a book for well-established researchers with deep knowledge on the field. What about the basic knowledge to approach this book? Basically, have knowledge about general concepts of molecular biology (DNA, RNA, proteins), homology, evolution, and most importantly, get along with the terminal console or at least not have serious disagreements with it.

Zapopan, Jalisco, Mexico — *Alejandro Pereira-Santana*
Mérida, Yucatán, Mexico — *Samuel David Gamboa-Tuz*
Mérida, Yucatán, Mexico — *Luis Carlos Rodríguez-Zapata*

Contents

Contributors

Tania Vanessa Arellano-Fernández • *Laboratorio de Sistemas Genéticos, Langebio, Cinvestav, Mexico; Escuela Nacional de Estudios Superiores, Unidad León, UNAM, León, Mexico*

J. Abraham Avelar-Rivas • *Laboratorio de Sistemas Genéticos, Langebio, Cinvestav, Mexico*

Philipp E. Bayer • *Applied Bioinformatics Group, School of Biological Sciences, The University of Western Australia, Perth, WA, Australia*

Esaú Bojórquez-Velázquez • *Red de Estudios Moleculares Avanzados, Clúster Científico y Tecnológico BioMimic®, Instituto de Ecología A.C. (INECOL), Xalapa, Veracruz, Mexico*

Julien Boutte • *Department of Biology, Hobart and William Smith Colleges, Geneva, NY, USA*

James Breen • *Indigenous Genomics, Telethon Kids Institute, Adelaide, SA, Australia*

Edder D. Bustos-Díaz • *Laboratorio de Evolución de la Diversidad Metabólica, Langebio, Cinvestav, Mexico*

Carlos P. Cantalapiedra • *Estación Experimental de Aula Dei-CSIC, Zaragoza, Spain*

Debarati Chakraborty • *Department of Molecular Biology and Biotechnology, University of Kalyani, Kalyani, India*

Gavin C. Conant • *Bioinformatics Research Center, North Carolina State University, Raleigh, NC, USA; Program in Genetics, North Carolina State University, Raleigh, NC, USA; Department of Biological Sciences, North Carolina State University, Raleigh, NC, USA*

Bruno Contreras-Moreira • *Estación Experimental de Aula Dei-CSIC, Zaragoza, Spain*

Álvaro Rodríguez del Río • *Estación Experimental de Aula Dei-CSIC, Zaragoza, Spain*

David Edwards • *Applied Bioinformatics Group, School of Biological Sciences, The University of Western Australia, Perth, WA, Australia*

Sawsan Elateek • *Genetic Department, Faculty of Agriculture, Ain Shams University, Cairo, Egypt*

José M. Elizalde-Contreras • *Red de Estudios Moleculares Avanzados, Clúster Científico y Tecnológico BioMimic®, Instituto de Ecología A.C. (INECOL), Xalapa, Veracruz, Mexico*

Aarón Espinosa-Jaime • *Escuela Nacional de Estudios Superiores, Unidad León, UNAM, León, Mexico*

Oscar Estrada • *Centre for Anthropobiology and Genomics of Toulouse (CAGT), CNRS UMR 5288, Université Toulouse III - Paul Sabatier, Toulouse, France; Australian Centre for Ancient DNA (ACAD), School of Biological Science, The University of Adelaide, Adelaide, SA, Australia; Grupo de Agrobiotecnología, Instituto de Biología, Universidad de Antioquia, Medellín, Colombia*

Selene L. Fernandez-Valverde • *Unidad de Genómica Avanzada, Langebio, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, Irapuato, Guanajuato, Mexico*

Mark Fishbein • *Department of Plant Biology, Ecology and Evolution, Oklahoma State University, Stillwater, OK, USA*

Samuel David Gamboa-Tuz • *Unidad de Biotecnología, Centro de Investigación Científica de Yucatán, Mérida, Yucatán, Mexico*

Diego Garfias-Gallegos • *Laboratorio de Genómica Ecológica y Evolutiva, Langebio, Cinvestav, Mexico*
Mitchell Gill • *Applied Bioinformatics Group, School of Biological Sciences, The University of Western Australia, Perth, WA, Australia*
Agnieszka A. Golicz • *Department of Plant Breeding, Justus Liebig University Gießen, Gießen, Germany*
Isabelle Guigon • *Univ. Lille, CNRS, Inserm, CHU Lille, Institut Pasteur de Lille, US 41 – UAR 2014 – PLBS – Plateforme bilille, Lille, France*
Yue Hao • *Biodesign Center for Mechanisms of Evolution, Arizona State University, Tempe, AZ, USA*
Sylvain Legrand • *Univ. Lille, CNRS, UMR 8198, Evo-Eco-Paleo, Lille, France*
José Abel Lovaco-Flores • *Escuela Nacional de Estudios Superiores, Unidad León, UNAM, León, Mexico; BetterLab—C3, Irapuato, Mexico*
Jacob I. Marsh • *Applied Bioinformatics Group, School of Biological Sciences, The University of Western Australia, Perth, WA, Australia*
Michael Matschiner • *Department of Palaeontology and Museum, University of Zurich, Zurich, Switzerland; Natural History Museum, University of Oslo, Oslo, Norway*
Victoria Mesa • *3PHM, INSERM, Faculté de Santé, Université Paris Cité, Paris, France*
Andrew D. L. Nelson • *Boyce Thompson Institute, Cornell University, Ithaca, NY, USA*
Anna C. Nelson Dittrich • *Boyce Thompson Institute, Cornell University, Ithaca, NY, USA*
Benjamin J. Nestor • *Applied Bioinformatics Group, School of Biological Sciences, The University of Western Australia, Perth, WA, Australia*
Hans-Wilhelm Nützmann • *Department of Biology and Biochemistry, The Milner Centre for Evolution, University of Bath, Bath, UK*
Katarzyna Oktaba • *Unidad Irapuato, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (Cinvestav), Irapuato, Guanajuato, Mexico*
Alejandro Pereira-Santana • *Unidad de Biotecnología Industrial, Centro de Investigación y Asistencia en Tecnología y Diseño del Estado de Jalisco, A.C., Zapopan, Jalisco, Mexico*
Francisco J. Pérez-de los Santos • *Unidad de Genómica Avanzada, Langebio, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, Irapuato, Guanajuato, Mexico*
América Ramírez-Colmenero • *Unidad de Genómica Avanzada, Langebio, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, Irapuato, Guanajuato, Mexico; Unidad Irapuato, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, Irapuato, Guanajuato, Mexico*
Stephen M. Richards • *Australian Centre for Ancient DNA (ACAD), School of Biological Science, The University of Adelaide, Adelaide, SA, Australia*
Eliel Ruiz-May • *Red de Estudios Moleculares Avanzados, Clúster Científico y Tecnológico BioMimic®, Instituto de Ecología A.C. (INECOL), Xalapa, Veracruz, Mexico*
Rubén Sancho • *Estación Experimental de Aula Dei-CSIC, Zaragoza, Spain; Escuela Polité cnica Superior, Universidad de Zaragoza, Huesca, Spain*
M. Eric Schranz • *Biosystematics Group, Wageningen University and Research, Wageningen, The Netherlands*
Nelly Sélem-Mójica • *BetterLab—C3, Irapuato, Mexico; Centro de Ciencias Matemáticas, UNAM, Morelia, Mexico*

Abdoallah Sharaf • *Institute of Plant Molecular Biology, Biology Centre, Czech Academy of Sciences, České Budějovice, Czech Republic; Genetic Department, Faculty of Agriculture, Ain Shams University, Cairo, Egypt*

Jesús Emiliano Sotelo-Fonseca • *Unidad de Genómica Avanzada, Langebio, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, Irapuato, Guanajuato, Mexico; Unidad Irapuato, Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, Irapuato, Guanajuato, Mexico*

Shannon C. K. Straub • *Department of Biology, Hobart and William Smith Colleges, Geneva, NY, USA*

Cassandria G. Tay Fernandez • *Applied Bioinformatics Group, School of Biological Sciences, The University of Western Australia, Perth, WA, Australia*

Mariana Tiscareño-Andrade • *Unidad Irapuato, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (Cinvestav), Irapuato, Mexico; Grupo de Desarrollo Reproductivo y Apomixis, Unidad de Genómica Avanzada Laboratorio Nacional de Genómica para la Biodiversidad (LANGEBIO), Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (Cinvestav), Irapuato, Mexico*

Hélène Touzet • *Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, Lille, France*

Jean-Philippe Vielle-Calzada • *Grupo de Desarrollo Reproductivo y Apomixis, Unidad de Genómica Avanzada Laboratorio Nacional de Genómica para la Biodiversidad (LANGEBIO), Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (Cinvestav), Irapuato, Mexico*

Pablo Vinuesa • *Centro de Ciencias Genómicas, Universidad Nacional Autónoma de Mé xico, Cuernavaca, Mexico*

Jesús Alejandro Zamora-Briseño • *Red de Estudios Moleculares Avanzados, Clúster Científico y Tecnológico BioMimic®, Instituto de Ecología A.C. (INECOL), Xalapa, Veracruz, Mexico*

Tao Zhao • *State Key Laboratory of Crop Stress Biology for Arid Areas/Shaanxi Key Laboratory of Apple, College of Horticulture, Northwest A&F University, Yangling, China*

Claudia Zirión-Martínez • *Laboratorio de Genómica Ecológica y Evolutiva, Langebio, Cinvestav, Mexico*

Part I

Phylogenetics and Evolution

Chapter 1

Orthology Prediction and Phylogenetic Analysis Methods in Plants

Abdoallah Sharaf and Sawsan Elateek

Abstract

In this chapter, we outline a pipeline for ortholog prediction and phylogenetic analysis in plants. This computational pipeline uses algorithms from different software to enable bioinformatic-beginner biologists to predict orthologs that can be shared with many distinct plant nonmodel and model species and identify gene loss events. Prediction of orthologs allows (1) investigation of the evolutionary relationships of plant genomes, (2) discovery of their origin, function, and (3) the impact of their adaptability to the environment.

We developed a pipeline to fit, not only eukaryote but also prokaryote organisms, with small or large genomes. All results acquired from the orthologs predication will enable phylogenetic tree construction, using gene and species (phylogenomic) phylogeny approaches.

Key words Molecular evolution, Homology search, Orthologs assignment, Domain screening, Maximum likelihood phylogeny, Bayesian inference, Phylogenomic phylogeny, Coalesce phylogeny

1 Introduction

In the era of high-throughput sequencing technologies, the evolutionary relationships of many lineages on the tree of life are being resolved. Thus, our knowledge of the evolution of these lineages has been recently advanced [1–4]. Nevertheless, ortholog identification remains a challenge in phylogenetic construction, as there still no robust automated tools to distinguish all levels of paralogy and more importantly, accurately select orthologs.

Orthologs genes are derived through speciation from a single ancestral sequence. Orthologs retain the same function (protein family), which its homologs can be separated into orthologs. Orthology is the foundation of gene and protein function

Supplementary Information The online version contains supplementary material available at [https://doi.org/10.1007/978-1-0716-2429-6_1].

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_1,

prediction, and can be studied via two approaches: graph (pairwise sequence comparison)-based or tree (phylogenetic analysis)-based approach [5, 6]. Graph (clustering)-based approach employs Bidirectional Best Hits (BBH) method, which is still used by researchers, even though there are reports of uncertainty regarding the identified orthologs [6–8]. Researchers improve this approach by using reciprocal best BLAST hits [8–13]. Alternatively, other researchers prefer using the manual curation of candidate homologs by phylogenetic analysis [14, 15] as well as recent databases and software which use phylogenetic analysis-based protocols to identify orthologs [16, 17].

Previously, phylogenetic analysis was based on several algorithms such as UPGMA (Unweighted Pair Group Method with Arithmetic means) and Neighbor-joining (NJ) algorithms. These algorithms use the distance-based method to justify the distance between each pair of sequences which is usually not precise [6, 18]. This method is neither specific nor accurate as it uses shortcuts to give faster data. On the other hand, character-based methods, such as Maximum Parsimony (MP), Maximum Likelihood (ML), and Bayesian Inference (BI), are justified by alignment information (nucleotides-based) using Heuristic (optimization) algorithm. Compared to the distance-based method, this method is more computationally expensive, but it is advantageous in terms of precision and quality of the resulting phylogenetic tree [6, 18]. Bayesian inference which uses a numerical method such as Markov Chain Monte Carlo (MCMC) has recently gained popularity among the phylogenetic community for several reasons [19]: It gives the possibility to account for phylogenetic uncertainty by using prior information. It incorporates complex models of evolution that limited computational analyses applying Heuristic methods. Different software, including MEGA, that incorporate these methods to construct the phylogenetic tree [20] are easy to use. Unfortunately, they do not generate high-quality outputs and their graphical output of extensive datasets has several limitations.

For all these previous dilemmas, we have created a bioinformatic pipeline described in this chapter, that can be used as a tool to aid researchers in carrying out phylogenetic analyses. It will be demonstrated on identified orthologs sequence data, providing users with a "step-by-step" guide. The pipeline presented here shows the potential to identify and study evolutionary relationships of protein families [21] and protein complex subunits using command-line tools which make this pipeline more suitable for large data analyses. As a proof of concept and to demonstrate the potential benefits of this robust pipeline, we identify orthologs of proteins in different eukaryotic supergroups [37]. An automated version of the orthology prediction part of this pipeline has been deposited at the GitHub (https://github.com/Iva-Mozgova-Lab/PcG_finder).

2 Materials

2.1 Equipment

2.1.1 Hardware

A personal computer with operating system 64-bit Linux and ≥4 GB of RAM, 100 GB of disk space and internet access is required for setting up the analyses. A smaller amount of memory and hard drive (one fifth or less) are recommended for the current example data, while more memory and hard drive capacity are necessary for a larger amount of data.

2.1.2 Software Requirements

All the analyses are performed locally on a Linux workstation; the pipeline was tested on the Ubuntu Linux distribution. Although Linux is the preferred environment, most of the required software tools are also available online through the software official web or the Galaxy server (https://usegalaxy.org/) (except for the AliView).

- Web browser (optional).
- Bash command line.
- Samtools (http://www.htslib.org/).
- DIAMOND (http://www.diamondsearch.org/index.php).
- HMMER (http://hmmer.org/).
- eggnog-mapper (https://github.com/eggnogdb/eggnog-mapper/wiki/eggNOG-mapper-v2).
- MAFFT (https://mafft.cbrc.jp/alignment/software/).
- trimAl (http://trimal.cgenomics.org/trimal).
- AliView (http://www.ormbunkar.se/aliview/).
- IQ-TREE v2.0.3 (http://www.iqtree.org/).
- PhyloBayes v4.1c (http://www.atgc-montpellier.fr/phylobayes/).
- ASTRAL-III v5.15.1 (https://github.com/smirarab/ASTRAL/tree/MP).
- Figtree (http://tree.bio.ed.ac.uk/software/figtree/).

Online Software

- ITOL: Interactive Tree Of Life (https://itol.embl.de/) (optional).

2.1.3 Example Data

Publicly available data are used in this proof-of-concept example. These can be obtained from The Universal Protein Resource (UniProt) database (Table 1). Also, *Arabidopsis thaliana* protein sequences of the chloroplast Ribulose bisphosphate carboxylase large chain (*rbcL*) and Maturase K (*matK*) are used as query sequences for the homology search (Table 2). Using protein sequences is recommended for accurate inference of evolutionary relationships because the amino acid sequences change less rapidly than the corresponding nucleotides [22] (*see* **Note 1**).

Table 1
List of the proteomes used in the example data. Showing the organisms taxonomic and their proteomes' quality information

Proteome ID	Organism	Code	Organism ID	Family	Protein count	BUSCO
UP000054558	*Klebsormidium nitens*	Kn	105231	Klebsormidiaceae	16,251	**C**: 96.9% [S: 96.5%, D: 0.5%] **F**: 1.9%, **M**: 1.2%, **n**: 425
UP000265515	*Chara braunii*	Cb	69332	Characeae	35,576	**C**: 72.2% (S: 65.4% D: 6.8%) **F**: 11.5% **M**: 16.2% **n**: 425
UP000006727	*Physcomitrium patens*	Pp	3218	Funariaceae	30,858	**C**: 97.6% (S: 86.1% D: 11.5%) **F**: 0.2% **M**: 2.1% **n**: 425
UP000077202	*Marchantia polymorpha*	Mp	1480154	Marchantiaceae	17,951	**C**: 91.3% (S: 91.3% D: 0%) **F**: 4% **M**: 4.7% **n**: 425
UP000244005	*Marchantia aquatica*	Ma	3197	Marchantiaceae	21,856	**C**: 96.7% (S: 75.5% D: 21.2%) **F**: 0.7% **M**: 2.6% **n**: 425
UP000001514	*Selaginella moellendorffii*	Sm	88036	Selaginellaceae	33,150	**C**: 89.4% (S: 12.5% D: 76.9%) **F**: 3.5% **M**: 7.1% **n**: 425
UP000006548	*Arabidopsis thaliana*	At	3702	Brassicaceae	39,349	**C**: 99.8% (S: 57.6% D: 42.1%) **F**: 0% **M**: 0.2% **n**: 425

Table 2
Information of the *Arabidopsis thaliana rbcL* and *matK* protein sequences used as queries in the example data

Sequence id	Sequence name	Protein name	Gene name	Length
O03042	RBL_ARATH	Ribulose bisphosphate carboxylase large chain (RuBisCO large subunit) (EC 4.1.1.39)	rbcL	479
P56784	MATK_ARATH	Maturase K	matK	504

2.2 Equipment Setup

2.2.1 Software Installation

- It is assumed that the user will use an Ubuntu-based Linux distribution. The installation was tested on Ubuntu LTS version 18.04. For non-Linux users, we recommend installing the required software using the conda package management system (https://docs.conda.io/en/latest/). The user must create the conda environment after the installation using the following command.

```
conda create -n [user-choice environment name]
```

- In conda package management system, software can be searched using conda search [software name] and install using conda install [software name] (*see* **Note 2**).
- Install Samtools (http://www.htslib.org/) by using the following command in a command-line terminal after activating the conda environment using conda activate [created conda environment name].

```
conda install -c bioconda samtools=1.11
```

- Install DIAMOND (http://www.diamondsearch.org/index.php) by using the following command in a command-line terminal.

```
conda install -c bioconda diamond=2.0.4
```

- Install HMMER: biosequence analysis using profile hidden Markov models (http://hmmer.org/) by using the following command in a command-line terminal:

```
conda install -c bioconda hmmer=3.3.1
```

- Install eggNOG mapper v2: Multiple alignment program for amino acid or nucleotide sequences by downloading the latest version of eggnog-mapper in the GitHub repository https://github.com/eggnogdb/eggnog-mapper/wiki/eggNOG-mapper-v2, using the following command.

```
git clone https://github.com/jhcepas/eggnog-mapper.git
```

- Upon downloading and unzipping, download necessary databases using the following script.

```
download_eggnog_data.py
```

- Install MAFFT: Multiple alignment program for amino acid or nucleotide sequences (https://mafft.cbrc.jp/alignment/

software/) by using the following command in a command-line terminal.

```
conda install -c bioconda mafft=7.471
```

- Install trimAl: A tool for the automated alignment trimming (http://trimal.cgenomics.org/trimal) by using the following command in a command-line terminal.

```
conda install -c bioconda trimal=1.4.1
```

- Install AliView by download the installer from (http://www.ormbunkar.se/aliview/downloads/linux/), Upon downloading and unzipping install AliView using the following commands (most likely you need to "sudo" the installation, if you do not have super user rights to the system see below) (*see* **Note 3**):

```
sudo chmod a+x install.sh
sudo ./install.sh
```

- Install IQ-TREE v2.0.3 (http://www.iqtree.org/) for Maximum-likelihood phylogenetic reconstruction using the following command in a command-line terminal:

```
conda install -c bioconda iqtree=2.0.3
```

- Install PhyloBayes v4.1c (http://www.atgc-montpellier.fr/phylobayes/) for Bayesian phylogenetic reconstruction using the following command in a command-line terminal.
- Download the software source file using the following.

```
wget https://megasun.bch.umontreal.ca/People/lartillot/www/
phylobayes4.1c.tar.gz
```

- Extract the source file in current directory.

```
tar -xf phylobayes4.1c.tar.gz
```

- The software executable file (pb) can be found in the path "phylobayes4.1c/data/".
- Install ASTRAL-III v5.15.1 (https://github.com/smirarab/ASTRAL/tree/MP) for computing coalescence phylogenetic tree using the following command in a command-line terminal.
- Download the software GitHub repository using the following.

```
wget https://github.com/smirarab/ASTRAL/archive/MP.zip
```

- Extract the repository in current directory and install it. Then, you simply use the jar file that is included with the repository. ASTRAL is a java-based application, and required Java 1.6 or later.

```
unzip MP.zip
cd ASTRAL-MP
./make.sh
```

- Install figtree (http://tree.bio.ed.ac.uk/software/figtree/) by using the following commands in a command-line terminal.

```
conda install -c bioconda figtree=1.4.4
```

3 Methods

Our presented pipeline for orthology prediction is divided into three steps while phylogenetic analysis can be done with another three steps (Fig. 1).

3.1 Orthology Prediction

3.1.1 Homology Search

For homology search we will use Jackhmmer. It implements methods using probabilistic models called profile Hidden Markov Models (profile HMMs) with an iterative search [23].

```
jackhmmer --tblout Chara_braunii.jack --cpu 30 -N 10 --noali
AtrbcL.faa Chara_braunii.fasta
```

--tblou	to save results as a table of per-sequence hits.
Chara_braunii.jack	the output (results) file.
--cpu	number of parallel CPU workers to use for multithreads.
-N	set maximum number of iterations.
--noali	don't output alignments, so output is smaller.
AtrbcL.faa	the query sequence.
Chara_braunii.fasta	the searched sequences database (proteome).

The previous command will produce a tab-separated file (Chara_braunii.jack) with the homology search results (Supplementary File S1). You need to apply the same to the rest of the proteomes; all proteomes can be searched once by looping the command over all the proteomes files as follows.

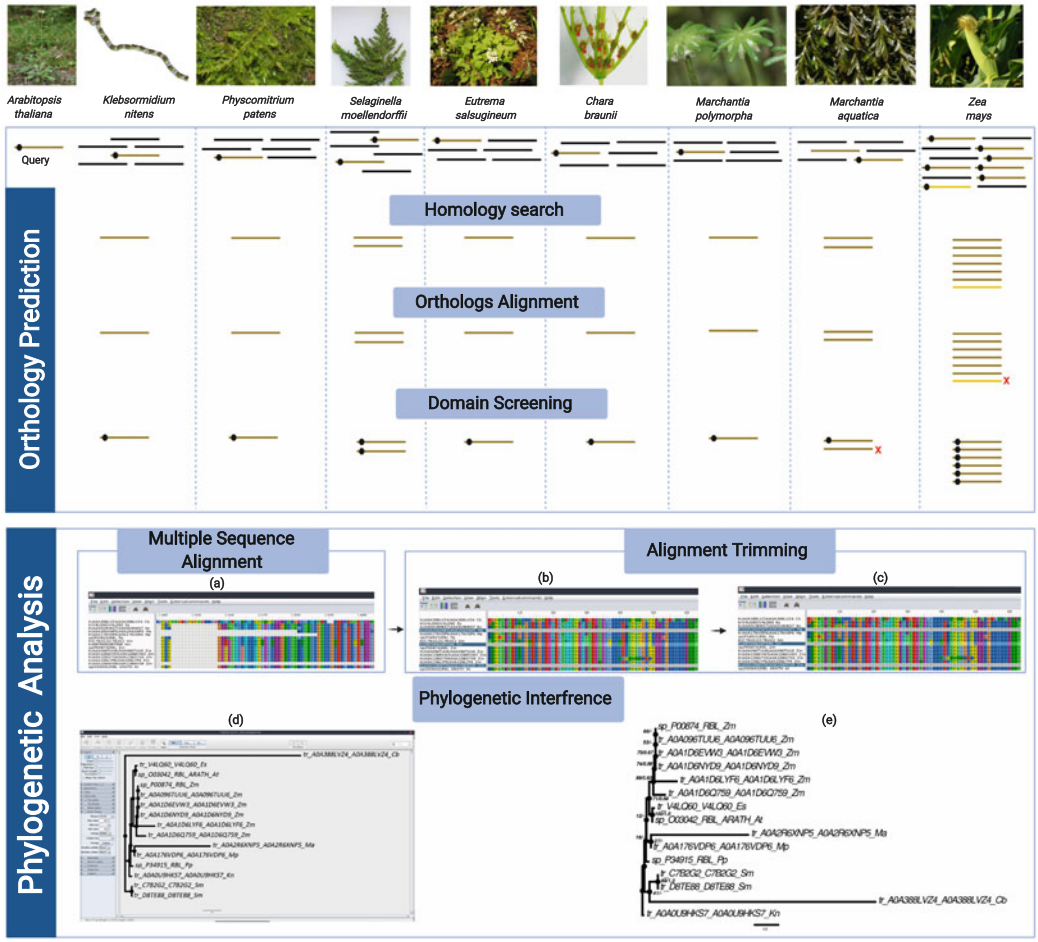


Fig. 1 Schematic representation of the orthology prediction and phylogenetic analysis steps. (**a**) The Multiple Alignments Sequences (MSA) file of the *rbcL* sequences dataset visualized on the AliView software. (**b**) The trimmed MSA file with the terminal gaps and (**c**) after terminal gabs replaced with missing data symbol (?) using AliView software, the edited sequences are highlighted in blue. The *rbcL* phylogenetic tree in the ML topology visualized using figtree software (**d**) without presenting merged support values and (**e**) after presenting the merged support values and rooted on the *Klebsormidium nitens rbcL* sequence and tree's nodes ordered increasingly

```
for i in 'ls *.fasta|sed 's/.fasta//g''; do jackhmmer --tblout
$i.jack --cpu 30 -N 10 --noali AtrbcL.faa $i.fasta;done
```

The corresponding hit(s) sequence(s) can be retrieved by parsing jackhammer's hit(s) using the following command. You need to apply the same to the rest of the proteomes' homology hits.

```
samtools faidx Chara_braunii.fasta -o Chara_braunii.faa \ 'tr|
A0A388LVZ4|A0A388LVZ4_CHABU'
```

-o to save retrieved sequence with the header 'tr|A0A388LVZ4|A0A388LVZ4_CHABU' as sequence file "Chara_braunii.faa" (*see* **Note 4**).

3.1.2 Orthologs Assignment

Due to the computational demanding of the de novo orthologs prediction via the tree-based approach, we will use eggNOG mapper software, which utilizes a tree-based orthologs database (OrthoDB) [24] to validate our homology search hits. This way is computationally cheaper than de novo inference and reduces errors associated with incomplete gene sampling [6].

In our case, the homology search of the *Selaginella moellendorffii* and *Zea mays* proteomes show multiple hits (2 and 7 hits respectively). These proteomes show high percentage of gene duplication according to BUSCO status (Table 1). It will therefore provide a good example to validate the homology search hits using this step.

The COG/KOG of our target protein must be identified first using the eggNOG mapper v2. *Arabidopsis* sequence at the AtrbcL.faa file has been used as a query in our case using the following command.

```
emapper.py -i AtrbcL.faa -o At -m diamond
```

```
-i Input FASTA file containing query sequences.
-o Base name for output files.
-m Search method.
```

The previous command will produce two files with extinctions ".emapper.seed_orthologs" and ".emapper.annotations". Supplementary File S2 illustrates the results in the file with extinction ".emapper.annotations" which contains all annotation of our sequences including the COG/KOG category under the column name "eggNOG OGs". We always choose the hit labeled as "COG@ or KOG@", in our case it's **COG1850**.

- Orthologs can be assigned for the target sequences resulting from the homology search using eggNOG mapper v2 with the following commands.

```
eggnog-mapper/emapper.py -i Zea_mays.faa --output Zm -m diamond
```

As previously mentioned, the same can be applied to the rest of the hits within all proteomes, and all hits can be validated once by looping the command over all the proteome files as follows.

```
for i in 'ls *.faa|sed 's/.faa//g'';do eggnog-mapper/emapper.py \
-i $i.faa --output $i -m diamond;done
```

The annotation results of the *Selaginella moellendorffii* hits show that both hits matched with our target COG category "**COG1850**" (Supplementary File S3) which means that *Selaginella moellendorffii* contains two homologs (paralogs) for the *rbcL* gene. On the other hand, one of the seven *Zea mays* homologs did not match with the target COG category (Supplementary File S4); this hit 'tr|A0A1D6EVW0|A0A1D6EVW0_MAIZE" must be filtered out as it is an artifact or false-positive hit.

3.1.3 Domain Screening

For the last step of orthology prediction, domain architecture of validated sequences will be identified. It is common to use entire or partial gene sequences as template for ortholog identification. However, it has been suggested that domains are suitable factors for orthology and consequently for phylogenetic inference [6, 25, 26].

We recommend that a conserved domain (e.g., a catalytic domain) which is found within our sequence to consider it a true homolog. This step can be ignored in case the information regarding the conserved domain of the gene of interest is missing or the domain architecture is unclear in general. Considering that, some genes can lose some protein domains throughout evolutionary history as a consequence of reductive evolution. Such protein domain change requires genetic recombination events [27], which means that this step is recommended to identify the analogs proteins; analogs are proteins with related folds and function.

As an example, we used *rbcL* protein that contains the catalytic domain Ribulose bisphosphate carboxylase Large chain (*RuBisCO_large*, PF00016) and/or the conserved domain Ribulose bisphosphate carboxylase large chain, N-terminal domain (*RuBisCO_large_N*, PF02788) [28] while the *matK* protein contains the conserved domain MatK/TrnK amino terminal region (*MatK_N*, PF01824) [29].

Hmmscan search algorithm was used to screen the domain architecture (https://www.ebi.ac.uk/Tools/hmmer/search/hmmscan) which is one of the Hidden Markov Model (HMM)-based tools (HMMER, http://hmmer.org/) [23]. Hmmscan is used to search the protein sequences against collections of protein profiles.

- A domain database needs to be downloaded first. The domain database we used is an nonredundant domains database (Pf_Sm) which resulted from merging SMART and Pfam databases [30, 31]. The Pf_Sm database can be downloaded using the following command.

```
wget -v https://www.dropbox.com/s/dhdi8xu64osizn6/Pf_Sm?dl=0
--content-disposition
```

- Prepare the domains database for hmmscan using the following command.

```
hmmpress Pf_Sm
```

- Screen the domain using hmmscan for all the validated sequences from last step which can be performed once with the following command.

```
 for i in 'ls *.faa|sed 's/.faa//g'';do hmmscan --tblout $i.
domain \
--noali --cpu 30 Pf_Sm $i.faa;done
```

--tblout to save results as a table of per-sequence hits.
$i.domain the output (results) file.
--cpu number of parallel CPU workers to use for multi-threads.
--noali don't output alignments, so output is smaller.
Pf_Sm the searched domains database.
$i.faa the query sequence.

The previous command will produce a tab-separated with extinction files (.domain) with the domain search results. Supplementary File S5 illustrated the results in the file "Es.domain" that contains domain screen results of the validated sequence of *Eutrema salsugineum*. The results showed that the *Eutrema salsugineum rbcL* protein sequence contains both (*RuBisCO_large* and *RuBisCO_large_N*) domains. We identified *RuBisCO_large* and/or *RuBisCO_large_N* domains in the rest of the validated sequences of the other organisms except *Chara braunii*. The domain screening of *Chara braunii*'s *rbcL* protein sequence 'tr|A0A388LVZ4|A0A388LVZ4_CHABU' revealed that it does not contain either *RuBisCO_large* or *RuBisCO_large_N* domains. As mentioned above, we can filter this sequence out or keep it which depends on our aim of the analysis. In our case, we will keep it since we are looking for the evolutionary relationships between the studied organisms.

The final *rbcL* identified orthologs including the query sequence file "AtrbcL.faa" needed to be combined in one file "rbcl.fasta", and the same for the *matK* identified orthologs "matk.fasta". This can be done by merging all fasta files together using the following command.

```
cat *.faa >> rbcl.fasta
```

(*see* **Note 5**)

3.2 Phylogenetic Analysis

3.2.1 Multiple Sequence Alignment (MSA)

MAFFT v. 7 software was used, MAFFT is one of the most software used to compute the Multiple Sequence Alignment (MSA), which adopt the progressive approach [32]. Furthermore, L-INS-I method was used as a recommended method for deeper divergences and an accurate alignment of up to ~200 sequences × ~2000 site using one of the following commands.

```
mafft-linsi --thread 30 rbcl.fasta > rbcl.fas
```

OR

```
mafft --thread 30 --maxiterate 1000 --localpair rbcl.fasta >
rbcl.fas
```

- If you are unsure which method is suitable for your sequence datasets, the following command is recommended.

```
mafft --thread 30 --auto rbcl.fasta > rbcl.fas
```

```
--thread    number of parallel CPU workers to use for multi-
            threads.
rbcl.fasta  the input file.
rbcl.fas    the output (MSA) file.
```

The alignments need to be visualized for manual inspection, to detect any sequences with spurious and poorly aligned regions and to remove any sequences with noticeably short alignments. Alignments can be visualized by running AliView software then choose "openFile" option from the File drop-down menu or press Ctrl + O. The Alignment blocks will be produced once you choose the output alignments file "rbcl.fas" (Fig. 1a) (*see* **Note 6**).

3.2.2 Alignments Trimming

It is necessary to infer the phylogeny based on the conserved regions. Homologs proteins can include some unaligned regions which are not inherited, and some other regions that may have evolved so fast that the correct multiple alignments will be difficult to infer. To remove these poorly aligned regions, the software TrimAl will be used with the following command.

```
trimal -in rbcl.fas -out rbcl_trim.phy -phylip -gappyout
```

```
-gappyout Uses an automatic method to decide optimal thresh-
          olds, based on the gap percentage count over the
          whole alignment.
-in       Input file "rbcl.fas".
-out      Output file "rbcl_trim.fas", trimmed alignments.
-phylip   Output file in PHYLIP format.
```

The trimmed alignments need to be visualized as mentioned above, and not only for the manual inspection but also to replace terminal gaps with missing data value (?). If the terminal gaps are not labeled as missing data; it will be interpreted in the final software for phylogenetic analysis as deletions and the sequences containing such gaps may incorrectly look as a highly divergent sequence (Fig. 1b). Terminal gaps can be replaced with missing data value (?) using the option "Replace terminal GAPs into missing char (?)" under the drop-down menu Edit in the AliView software (Fig. 1c). Finally, the edited trimmed alignments can be saved with the option "Save as Phylip" under the File drop-down menu. This will produce an alignment file "rbcl_final.phy" with the final alignment for the phylogeny analysis.

3.2.3 Phylogenetic Inference

For a robust phylogenetic analysis, more sequences standing for a wide range of taxonomic groups with a balanced number of sequences from each group are recommended practices. In this chapter and for an easier presentation, we will use only the predicted homologs from the last step to compute the phylogeny analysis. Moreover, we will present two strategies of computing the phylogeny as follows.

Gene Phylogeny

Maximum Likelihood (ML) and Bayesian Inference (BI) are the most accurate and recommended algorithms for phylogenetic analysis [33]. We chose IQ-TREE and PhyloBayes software for phylogenetic analysis using ML and BI methods respectively. These software identify and automatically remove genes or taxa that show compositional bias from the analysis [6, 34].

Phylogenetic analysis for each gene (*rbcL* and *matK*) can be computed separately using the edited trimmed-alignment file "rbcl_final.fasta" with ML and BI methods respectively with the following commands.

- IQ-TREE command

```
iqtree -s rbcl_final.phy -m TEST -bb 1000 -nt AUTO
```

```
-s  Input file "rbcl_final.phy".
-m  SUBSTITUTION MODEL, "TEST" tells IQ-TREE to perform jMo-
    delTest/ProtTest
    and the remaining analysis using the selected model.
-bb Ultrafast bootstrap (>=1000).
-nt Number of cores, "AUTO" for automatic detection.
```

- PhyloBayes command which is suitable for single gene trees of average size (hundreds of amino acids long), assuming that the phylobayes at the current directory (*see* **Note 7**).

```
 ./phylobayes4.1c/data/pb -d rbcl_final.phylip -wag -cat -dgam
4 -f rbcl_pb.tree

-d   Input file "rbcl_final.fas".
-wag Improved, General Amino-Acid Replacement Matrix.
```

Within the several files that have been generated by each software, the ML and BI trees with the support values in NEWICK format can be found in files named "rbcl_final.phy.treefile" and "rbcl_final.fasta.contree" respectively. Moreover, the support value from both algorithms can be merged into one tree based on one of the algorithms topologies. Our python script "treecombine.py" can be used to perform such merging as following.

- Download the python script.

```
wget https://raw.githubusercontent.com/ObornikLEP/treecom-
bine/master/treecombine.py
```

- Run the script using the following command.

```
 python treecombine.py -l rbcl_final.py -m rbcl_final.phy.
treefile \
 -s rbcl_final.fasta.contree -o rbcl_merged.tree

 -l list of sequences id, alignment file "rbcl_final.py" can be
    used.
 -m master tree input file "rbcl_final.phy.treefile"
 -s secondary tree input file "rbcl_final.fasta.contree"
 -o output tree file "rbcl_merged.tree"
```

This will generate a tree file "rbcl_merged.tree" with the ML topology since the ML tree "rbcl_final.phy.treefile" was used as the master tree while the support values from both trees will be merged; the script will merge the support values, not the topologies. The final tree "rbcl_merged.tree" is in NEWICK format and can be visualized by any supported tree viewer programs. We will use figtree for visualizing our tree but you can use the optional online tool ITOL: Interactive Tree Of Life (https://itol.embl.de/) which can be used as well. Figtree can be used with the following command.

```
figtree rbcl_merged.tree
```

Press "OK" on the alert message to confirm your support values name, it is labeled as "label" by default but you can change it. The tree will appear on the software screen but without support value and rooting point (Fig. 1d).

The tree can be rooted by selecting the out-group sequence or branch then choose "Root on Branch" from the drop-down menu "Tree" or Ctrl + R. In our example the *Klebsormidium nitens* sequence "tr|A0A0U9HKS7|A0A0U9HKS7_Kn" will be used as the rooting point. Then the tree nodes can be ordered according to the length of the nodes by choosing "Increasing Node Order" or Ctrl + U from the drop-down menu "Tree." Finally, the merged support values can be presented by checking the "Node Labels" box from the left side menu then selecting "label" from the drop-down menu "Display." Two support values exist as described above, the first one is the ML analysis while the second one is based on the BI analysis (Fig. 1e). It is preferable to have the out-group sequence(s) which represent evolutionary ancestor(s), in order to determine the direction of ancestral relationships. If the rooting point is unclear within the studied sequences dataset then the unrooted tree layout is recommended.

The Final tree (Fig. 1e) shows a logical evolutionary order of our studied organisms (Table 1) since the higher land plant *Zea mays* is in the crown of the tree and rooted by other flowering plant sequences (*Arabidopsis thaliana* and *Eutrema salsugineum*). Moreover, all six paralogs of *Zea mays* are grouped in one clade "branch" revealing that these paralogs are a result of a gene duplication event. The same applies to *Selaginella moellendorffii* paralogs (Fig. 1e).

Species (Phylogenomic) Phylogeny

Species tree would usefully resolve any ambiguities within evolutionary relationships [6]. There are two approaches to infer a species tree: (1) the aligned genes can be concatenated into a supermatrix, which is analyzed to produce the species tree. A supermatrix can be generated, in the case of genes with one identified homolog in all studied species. (2) a separate gene tree based on each gene alignment that can be inferred and the different trees can then be coalesced to produce the species tree, known as the super-tree approach [6]. A coalescence phylogenetic tree can be generated if multiple homologs (paralogs) were identified in one or more species, which is the case in our example.

Alignment concatenation can be done using various software or online tools such as T-coffee server (Combine) (http://tcoffee.crg.cat/apps/tcoffee/do:combine). While, we will use ASTRAL-III tool (https://github.com/smirarab/ASTRAL/tree/MP) for estimating a super "coalescence" tree given a set of gene trees. ASTRAL is statistically consistent under the multispecies coalescent model [35]. We can compute coalescence phylogenetic tree from the ML trees of the genes "*rbcL* and *matK*" using the following command.

- We need to create first a map file; the species name should be different from the individual names when multiple individuals exist for the same species. This mapping file needs to be in one of the following two formats.

```
species_name [number of individuals] individual_1 individual_2
etc.
```

OR

```
species_name:individual_1,individual_2, etc.
```

(*see* **Note 8**)

- ASTRAL command, considering that the ASTRAL folder (ASTRAL-MP) is in your current directory.

```
java -Xmx3000M -D"java.library.path=./ASTRAL-MP/lib/" \
 -jar ./ASTRAL-MP/astral.5.15.1.jar -i rbcl_matk.trees \
 -o rbcl_matk_coles.tree -C -T 30 -a map.txt
```

-i Input file contains all the maximum likelihood rbcl and matk gene trees (rbcl_matk.trees).
-o The species ASTRAL tree "rbcl_matk_coles.tree"; includes support values, which are again drawn based on the 100 bootstrap replicate trees.
-C Use CPU only and do not use GPUs.
-T Number of threads to use "30".
-a Mapping file "map.txt", one line per species.

By default, ASTRAL performs 100 bootstrap replicates, but the -r option can be used to perform more replicates. ASTRAL will measure branch length in coalescent as well. Moreover, the support values are shown as local posterior probabilities which are revealed as a number between 0 and 1.

The generated species tree can be visualized in the same way as described above. The computed coalescence tree (Fig. 2) shows the same topology as the *rbcL* gene tree (Fig. 1e) but with shorter branching in *Marchantia aquatica* and *Chara braunii*. This example reveals the benefit of species (phylogenomic) phylogeny construction to resolve the Long-Branch Attraction (LBA).

4 Notes

1. BUSCO is a handy tool for assessing genome completeness [36] then we recommend obtaining the BUSCO status of the target genomes or proteomes data as a start (Table 1).

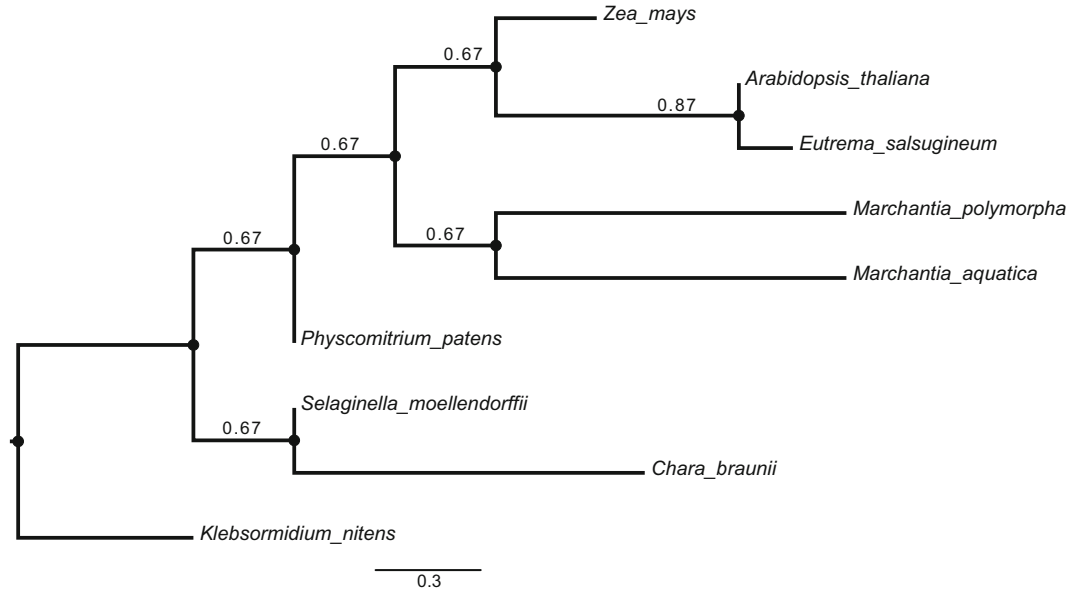


Fig. 2 The coalescence (species) phylogenetic tree inferred from the ML *rbcL* and *matK* gene trees using the ASTRAL-III tool

The information acquired from BUSCO can be used to identify the true gene loss and duplication events.

2. The conda environment can be created and all needed conda-available software and packages at once with the provided "package_list.txt" file (Supplementary File S6) using the following command.

```
conda create -n [user-choice environment name] --file package_list.txt
```

   - The user still needs to install the software which is not available in conda such as AliView.

3. For better performance of AliView software, consider upgrading to Java 8. There is a significant improvement in drawing speed since it uses Linux built in XRender. Install Oracle Java 8 using the following commands.

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt update
sudo apt install oracle-java8-installer
```

4. Samtools software will create a mapping file for each user-provided sequence database with the ".fai" extension, that is, "Chara_braunii.fasta.fai".
5. In this chapter, we tend to identify orthologs of a certain gene in several organisms. All orthologs groups between two or more organisms can be identified using new robust software named OrthoFinder (https://github.com/davidemms/OrthoFinder) [17].
6. Manual inspection of the aligned orthologs is important to detect and filter out wrongly identified contaminants (outlier) which can yield long branches, biased model parameters, or even changes to the tree topology. As a recommended approach to determine such outlier, the compatibility of closest neighbors can be tested with phylogenetic expectations. A fast ML tree based on the untrimmed alignments without any bootstrap should be suitable for this task using the following command.

```
iqtree -s rbcl.fas -m TEST -nt AUTO
```

7. Bayesian inference is a computationally expensive method; the analysis can take days to finish or never finished, especially with highly divergent amino acid sequence alignments. A Bayesian Inference phylogeny using the same example data is provided in the Supplementary File S7. The substitution model was chosen according to the best model identified during the ML tree construction; it can be found in the IQ-TREE log file "rbcl_final.phylip.log" under "Best-fit model."
8. The map file "map.txt" is provided in the Supplementary File S8.

Acknowledgments

This work was supported by the Czech Academy of Sciences, ERC-CZ, grant number [ERC200961901]. The work also supported by the Science and Technology Development Fund (STDF) and the Partnership for Research and Innovation in the Mediterranean Area (PRIMA), GENDIBAR project. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, funded under the programme "Projects of Large Research, Development, and Innovations Infrastructures." The authors thank Dr. Chayma Ben Saoud for improving the graphs, and Dr. Iva Mozgová for proofreading the manuscript.

References

1. Rokas A, Williams BI, King N, Carroll SB (2003) Genome-scale approaches to resolving incongruence in molecular phylogenies. Nature 425:798. https://doi.org/10.1038/nature02053
2. Gee H (2003) Ending incongruence. Nature 425:782
3. Hipp AL, Eaton DAR, Cavender-Bares J et al (2014) A framework phylogeny of the American oak clade based on sequenced RAD data. PLoS One 9:e93975. https://doi.org/10.1371/journal.pone.0093975
4. Widhelm TJ, Grewe F, Huang J-P et al (2019) Multiple historical processes obscure phylogenetic relationships in a taxonomically difficult group (Lobariaceae, Ascomycota). Sci Rep 9:8968. https://doi.org/10.1038/s41598-019-45455-x
5. Wang Y, Coleman-Derr D, Chen G, Gu YQ (2015) OrthoVenn: a web server for genome wide comparison and annotation of orthologous clusters across multiple species. Nucleic Acids Res 43:W78. https://doi.org/10.1093/nar/gkv487
6. Kapli P, Yang Z, Telford MJ (2020) Phylogenetic tree building in the genomic age. Nat Rev Genet 21:428–444. https://doi.org/10.1038/s41576-020-0233-0
7. Dalquen DA, Dessimoz C (2013) Bidirectional best hits miss many orthologs in duplication-rich clades such as plants and animals. Genome Biol Evol 5:1800. https://doi.org/10.1093/gbe/evt132
8. Salomaki ED, Eme L, Brown MW, Kolisko M (2020) Releasing uncurated datasets is essential for reproducible phylogenomics. Nat Ecol Evol 4:1435
9. Rotterová J, Salomaki E, Pánek T et al (2020) Genomics of new ciliate lineages provides insight into the evolution of obligate anaerobiosis. Curr Biol 30:2037. https://doi.org/10.1016/j.cub.2020.03.064
10. Lax G, Eglit Y, Eme L et al (2018) Hemimastigophora is a novel supra-kingdom-level lineage of eukaryotes. Nature 564:410. https://doi.org/10.1038/s41586-018-0708-8
11. Shaver S, Casas-Mollano JA, Cerny RL, Cerutti H (2010) Origin of the polycomb repressive complex 2 and gene silencing by an E (z) homolog in the unicellular alga Chlamydomonas. Epigenetics 5:301–312. https://doi.org/10.4161/epi.5.4.11608
12. Chen DH, Qiu HL, Huang Y et al (2020) Genome-wide identification and expression profiling of SET DOMAIN GROUP family in *Dendrobium catenatum*. BMC Plant Biol 20:1–19. https://doi.org/10.1186/s12870-020-2244-6
13. Burki F, Pawlowski J (2006) Monophyly of rhizaria and multigene phylogeny of unicellular bikonts. Mol Biol Evol 23:1922. https://doi.org/10.1093/molbev/msl055
14. Torruella G, Derelle R, Paps J et al (2012) Phylogenetic relationships within the Opisthokonta based on phylogenomic analyses of conserved single-copy protein domains. Mol Biol Evol 29:531. https://doi.org/10.1093/molbev/msr185
15. Saunders GW, Jackson C, Salomaki ED (2018) Phylogenetic analyses of transcriptome data resolve familial assignments for genera of the red-algal *Acrochaetiales-Palmariales* Complex (Nemaliophycidae). Mol Phylogenet Evol 119:151. https://doi.org/10.1016/j.ympev.2017.11.002
16. Huerta-Cepas J, Szklarczyk D, Heller D et al (2019) EggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. Nucleic Acids Res 47:D309. https://doi.org/10.1093/nar/gky1085
17. Emms DM, Kelly S (2019) OrthoFinder: phylogenetic orthology inference for comparative genomics. Genome Biol 20:238. https://doi.org/10.1186/s13059-019-1832-y
18. Panzetta A (2016) A new similarity measure for phylogenetic trees. Ca' Foscari University
19. Metropolis N, Rosenbluth AW, Rosenbluth MN et al (1953) Equation of state calculations by fast computing machines. J Chem Phys 21:1087. https://doi.org/10.1063/1.1699114
20. Kumar S, Stecher G, Li M et al (2018) MEGA X: molecular evolutionary genetics analysis across computing platforms. Mol Biol Evol 35:1547. https://doi.org/10.1093/molbev/msy096
21. Sharaf G, Jiroutová O (2019) Characterization of aminoacyl-tRNA synthetases in chromerids. Genes (Basel) 10:582. https://doi.org/10.3390/genes10080582
22. Hall BG (2005) Comparison of the accuracies of several phylogenetic methods using protein and DNA sequences. Mol Biol Evol 22:792–802. https://doi.org/10.1093/molbev/msi066
23. Johnson LS, Eddy SR, Portugaly E (2010) Hidden Markov model speed heuristic and iterative HMM search procedure. BMC Bioinformatics 11:431. https://doi.org/10.1186/1471-2105-11-431

24. Kriventseva EV, Kuznetsov D, Tegenfeldt F et al (2019) OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. Nucleic Acids Res 47:D807–D811. https://doi.org/10.1093/nar/gky1053
25. Scornavacca C, Galtier N (2016) Incomplete lineage sorting in mammalian phylogenomics. Syst Biol 66:syw082. https://doi.org/10.1093/sysbio/syw082
26. Sonnhammer ELL, Gabaldon T, Sousa da Silva AW et al (2014) Big data and other challenges in the quest for orthologs. Bioinformatics 30:2993–2998. https://doi.org/10.1093/bioinformatics/btu492
27. Forslund SK, Kaduk M, Sonnhammer ELL (2019) Evolution of protein domain architectures. In: Methods in molecular biology. Humana Press Inc., Totowa, NJ, pp 469–504
28. Taylor TC, Andersson I (1997) The structure of the complex between rubisco and its natural substrate ribulose 1,5-bisphosphate. J Mol Biol 265:432–444. https://doi.org/10.1006/jmbi.1996.0738
29. Mohr G, Perlman PS, Lambowitz AM (1993) Evolutionary relationships among group II intron-encoded proteins and identification of a conserved domain that may be related to maturase function. Nucleic Acids Res 21:4991–4997. https://doi.org/10.1093/nar/21.22.4991
30. El-Gebali S, Mistry J, Bateman A et al (2019) The Pfam protein families database in 2019. Nucleic Acids Res 47:D427. https://doi.org/10.1093/nar/gky995
31. Letunic I, Bork P (2018) 20 years of the SMART protein domain annotation resource. Nucleic Acids Res 46:D493. https://doi.org/10.1093/nar/gkx922
32. Katoh K, Standley DM (2013) MAFFT multiple sequence alignment software version 7: improvements in performance and usability. Mol Biol Evol 30:772–780. https://doi.org/10.1093/molbev/mst010
33. Sardaraz M, Tahir M, Aziz Ikram T, Bajwa H (2012) Applications and algorithms for inference of huge phylogenetic trees: a review. Am J Bioinformatics Res 2:21–26. https://doi.org/10.5923/j.bioinformatics.20120201.04
34. Nesnidal MP, Helmkampf M, Bruchhaus I, Hausdorf B (2010) Compositional heterogeneity and phylogenomic inference of metazoan relationships. Mol Biol Evol 27:2095–2104. https://doi.org/10.1093/molbev/msq097
35. Rabiee M, Sayyari E, Mirarab S (2019) Multi-allele species reconstruction using ASTRAL. Mol Phylogenet Evol 130:286–296. https://doi.org/10.1016/j.ympev.2018.10.033
36. Seppey M, Manni M, Zdobnov EM (2019) BUSCO: assessing genome assembly and annotation completeness. Methods Mol Biol 1962:227–245. https://doi.org/10.1007/978-1-4939-9173-0_14
37. Sharaf A, Vijayanathan M, Oborník M, Mozgová I (2022) Phylogenetic profiling resolves early emergence of PRC2 and illuminates its functional core. Life Science Alliance 5(7) e202101271. https://doi.org/10.26508/lsa.202101271

Chapter 2

Species Tree Inference with SNP Data

Michael Matschiner

Abstract

While the inference of species trees from molecular sequences has become a common type of analysis in studies of species diversification, few programs so far allow for the use of single-nucleotide polymorphisms (SNPs) for the same purpose. In this book chapter, I discuss the use of the Bayesian program SNAPP, which infers the species tree by mathematically integrating over all possible genealogies at each SNP. In particular, I focus on a molecular clock model developed for SNAPP, allowing the inference of divergence times together with the species tree topology and the population size, directly from SNP datasets in variant call format. With the growing availability of SNP datasets for multiple closely related species, this approach is becoming increasingly relevant for the reconstruction of the temporal framework of recent species diversification.

Key words Genomics, Phylogeny, Species tree, SNPs, Divergence times, SNAPP, BEAST

1 Introduction

Genetic data have long been used to infer relationships among individuals, populations, and species. Traditionally, DNA fragments were sequenced for certain markers and aligned to each other, and the resulting multiple sequence alignment was used to deduce relationships based on pairwise distances, parsimony, or the likelihood of the alignment under certain models of sequence evolution. When the same set of taxa have been sequenced for multiple markers, a common approach has been to join—concatenate—the multiple sequence alignments for these markers into a single alignment before the inference. However, investigations motivated by the growing number of multimarker datasets have identified important issues with this approach. Based on simulations, Kubatko and Degnan [1] demonstrated that under certain conditions, concatenation of alignments can lead to the inference of incorrect relationships among species that even receive greater support with increasing size of the dataset. Their conclusion has since been corroborated by several studies [2–4], including a mathematical

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_2,

proof of the statistical inconsistency of concatenation [5]. Moreover, while the inconsistency affects the estimated topology of the species tree only under certain conditions, the estimates for the lengths of the tree's branches are almost certainly affected by concatenation [6, 7]. This is particularly problematic when the species tree is time calibrated and branch lengths are used as a measure of the amount of time that passed between speciation events.

The underlying cause for the inconsistency resulting from concatenation is the fact that the true genealogies of markers may differ from each other and also from the true species tree, due to recombination. To account for this variation among marker genealogies in the estimation of the species tree, the multispecies coalescent (MSC) model has been developed [8] and implemented in a growing number of inference tools [9–14]. These tools fall into two categories where some estimate the marker genealogies jointly with the species tree and others rely on separately estimated marker genealogies as input. Under the assumptions of the MSC model, which include random mating within species, the absence of gene flow after speciation, and the absence of recombination within markers, inference of species trees with these tools is statistically consistent and therefore reliable [14]. Naturally, all of the assumptions of the MSC model may be violated by empirical systems, but as concatenation has been argued to represent nothing else than a particularly unrealistic special case of the MSC model [15], the use of this model may nevertheless improve the accuracy of species tree estimates. Of these assumptions of the MSC model, particularly the last one—the absence of within-marker recombination—has been criticized, and Springer and Gatesy [16] argued that, depending on population sizes, time between speciation events, and recombination rates, within-marker recombination can change the true genealogy as often as every few base pairs. In such cases, inference with the MSC model may be expected to suffer from the same problems as concatenation [16].

One alternative application of the MSC model that is immune to within-marker recombination is the estimation of species trees directly from single-nucleotide polymorphisms (SNPs) instead of marker sequences. With a length of a single base pair, recombination within a SNP is of course impossible. And while individual SNPs do not carry enough information for the estimation of the genealogy at the position of the SNP, this problem can be circumvented in two ways: In the quartet inference approach implemented in the program SVDquartets [17], the taxon set is decomposed into a large number of quartets (combinations of four species), the support for alternative quartet topologies is assessed, and quartet topologies are finally reassembled into the estimated species tree topology. The second possibility to avoid the uninformative genealogies of individual SNPs is implemented in SNAPP [18], which

integrates over all possible genealogies at each SNP mathematically rather than inferring them. This approach is conceptually elegant, but unfortunately comes at the cost of high computational demand, meaning that SNAPP can usually be applied only to comparatively small datasets of tens of species and thousands of SNPs. In contrast, SVDquartets runs quickly enough to be applied to hundreds of species and millions of SNPs. Besides these two methods based on the MSC, tools that model mutation and allelic drift instead of coalescent variation can also be applied to infer species trees from SNP data; these tools include POMO [19] and the recently developed Snapper [20].

Despite its limitation to smaller datasets, SNAPP is highly useful for the inference of species trees from recently diverged groups. As a Bayesian inference tool, SNAPP produces probabilistic node support values that can be interpreted intuitively, and it allows model comparisons by Bayes factors, which enables its use for species delimitation [21]. And as simulations have shown, the inferred species tree can be accurate and precise even when only hundreds of SNPs are used [7]. Finally, with the molecular clock model developed for SNAPP by Stange et al. [7], the program also estimates population sizes and divergence times, allowing the reconstruction of the temporal framework of species diversification.

In the rest of this chapter, I am going to focus on species tree estimation with SNAPP based on the model of Stange et al. [7], assuming that the reader is not only interested in the topology of the species tree but also in the timeline of diversification. I do not cover species tree inference with SVDquartets or species delimitation with SNAPP but would like to point the readers interested in these analysis types to the excellent tutorials that can be found online at www.phylosolutions.com (by Dave Swofford and Laura Kubatko) and www.evomics.org (by Adam Leaché), respectively.

2 Materials

SNAPP is available as an add-on package for BEAST 2 [22, 23]; therefore, both the BEAST 2 suite of programs and this add-on package are required for species tree inference with SNAPP. To use the model of Stange et al. [7] in SNAPP, the ***snapp_prep.rb*** script, written in the Ruby programming language, is additionally required. To apply this script, several input files, including a genotype data matrix, a file assigning individuals to species, a file with age constraints, and possibly a starting tree are needed. Finally, two more programs, Tracer [24] and FigTree are useful for post-processing the output of SNAPP.

2.1 BEAST 2

The BEAST 2 suite of programs includes BEAST itself, BEAUti, LogCombiner, and TreeAnnotator. BEAST employs Markov-chain Monte Carlo (MCMC) to infer the Bayesian posterior distribution of phylogenetic trees and parameter estimates, under models that are specified in input files in XML format [25]. To facilitate the writing of XML files, BEAST is distributed together with BEAUti, a graphical user interface program through which the model settings can be selected and exported in XML format (however, the SNAPP model of Stange et al. cannot be specified through BEAUti; see below). The user interface of BEAUti also represents an easy way to access the BEAST 2 Package Manager, through which add-on packages like SNAPP can be installed. The program LogCombiner can be used to merge posterior distributions from multiple replicate BEAST analyses into a single file, and TreeAnnotator serves to generate summary trees from the posterior tree distribution. The BEAST 2 suite of programs for Mac OS X, Linux, or Windows can be obtained freely from https://www.beast2.org. All BEAST 2 programs are written in the Java programming language and thus Java is required to run them. Therefore, either the Java Development Kit (version 8 or higher) should be installed (e.g., from https://adoptopenjdk.net) or one of the BEAST 2 versions bundled with Java should be selected from the BEAST 2 website (https://www.beast2.org). The version of Java installed can be identified on the command line with java -version; version number 1.8 or higher corresponds to Java Developer Kit version 8 or higher.

2.2 SNAPP

Owing to SNAPP's integration into BEAST 2, its model settings can be defined with BEAUti, its analyses use the MCMC machinery of BEAST, and postprocessing can be performed with the LogCombiner and TreeAnnotator tools distributed with BEAST 2. The model applied in SNAPP, however, is rather different from those of other BEAST analyses, due to its use of SNP markers instead of sequence alignments and the mathematical integration over all possible genealogies at each SNP. The SNAPP add-on package can be installed with the BEAST 2 Package Manager, accessed through BEAUti as shown in Fig. 1.

2.3 *snapp_prep.rb* and *add_theta_to_log.rb*

While the settings for most SNAPP analyses can be defined with BEAUti's graphical user interface, this is not the case for analyses with the molecular clock model of Stange et al. [7]. To implement this model, the XML file for SNAPP needs to be written differently, and one convenient way in which this can be done is the ***snapp_prep.rb*** Ruby script. The script can be obtained from GitHub at https://github.com/mmatschiner/snapp_prep. A second Ruby script, named ***add_theta_to_log.rb***, is useful for postprocessing of SNAPP results and available from the same repository. To run both scripts, the Ruby programming language (version 2 or higher) is

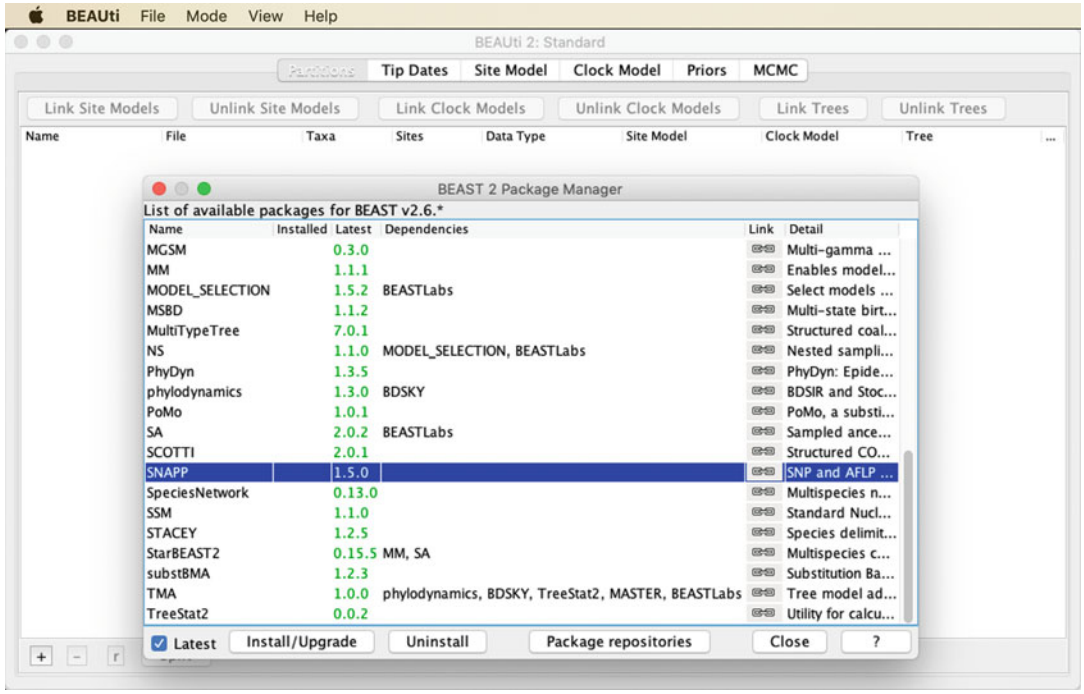


Fig. 1 Screenshot of the BEAUti graphical user interface with the BEAST 2 Package Manager. The Package Manager can be opened by clicking "Manage Packages" in BEAUti's "File" menu. SNAPP can then be installed by selecting the package as in the screenshot and clicking "Install/Upgrade" at the bottom of the Package Manager window

required. The language is included with Mac OS X and Linux operating systems but may first need to be installed on Windows systems. All installation options are described on https://www.ruby-lang.org/en/documentation/installation/. The installed version of Ruby can be identified on the command line with ruby --version.

2.4 Genotype Data Matrix

One of the inputs required by **snapp_prep.rb** to write an XML file for SNAPP is a matrix containing diploid genotype data. This matrix can be provided in Phylip format [26] or in uncompressed variant call format (VCF), the latter of which is probably more convenient for most users as genotyping data is most commonly stored in VCF files. As SNAPP can only handle biallelic SNPs, all indels, multiallelic SNPs, and monomorphic sites should first be removed from the matrix. To further comply with SNAPP's expectation of SNPs that are unlinked [18], it may be advisable to thin the matrix so that no two SNPs are within a short distance of each other on the same chromosome (this can also be done with ***snapp_prep.rb***; see below). Which minimum distance should be chosen may depend on the genome size and the target number of SNPs for the analysis, but minimum distances of at least thousands of base

pairs (bp) may be sensible (that said, the effect of linkage between some SNPs is likely negligible when the matrix includes thousands of SNPs). SNPs with missing data can be used by SNAPP as long as genotypes are known for at least one individual per species; SNPs for which this is not the case will be recognized by the ***snapp_prep.rb*** script and excluded from the produced XML file. Importantly, the genotype matrix should not be filtered by minor allele count or frequency as this filter would introduce bias to the estimated lengths of terminal branches of the species tree. Instead of filters on minor allele count or frequency, filtering for high genotype quality is recommended. The data matrix should not include genotypes for large numbers of individuals per species as these would primarily extend SNAPP's run times without adding much information to the analysis. As a rule of thumb, a total number of 20–30 (diploid) individuals across all species and between 1000 and 10,000 SNPs constitute a suitable dataset size, but if SNAPP's results should turn out to be too uninformative or if the run times are too long, these numbers should be adjusted. Note that SNAPP can provide good species tree estimates even when only a single (diploid) individual is used per species [7].

2.5 Species Table

SNAPP requires information assigning individuals to species, and to write this information into SNAPP's XML file, the ***snapp_prep.rb*** script expects an input file with a two-column table. The file should be in plain text format, the first column should list species IDs, and the second column should list the corresponding IDs of individuals. These individual IDs should exactly match those used in the genotype data matrix. The two columns can be either tab- or space-delimited. The table may include a header row; if it does, the row content should be "Species" in the first column and "Specimen," "Specimens," "Sample," or "Samples" in the second column; these keywords are case-insensitive. An example of a species table, taken from a study by Barth et al. [27], is shown in Table 1.

2.6 Age Constraints

In sequence-based analyses of divergence times, phylogenies are usually time calibrated either by specifying an estimate of the mutation rate or by placing age constraints on one or more divergence events in the tree. In SNP-based species tree inference with SNAPP, however, mutation rates applying to the dataset can usually not be estimated a priori because the SNP data are subject to ascertainment bias as only variable sites are included [7]. Thus, the better approach for time calibration of SNP-based species trees is to specify age constraints for divergences within the tree. The information for these constraints may come from the fossil record or from previous phylogenetic studies, but either way, some age information must be available for at least one divergence, otherwise the molecular clock model of Stange et al. [7] cannot be used. If the user should not be aware of published age estimates for the group

Table 1
Example of a species table assigning individuals to species, taken from Barth et al. [25]

Species	Specimen
mar	BOU15023
mar	BOU15010
mar	SAW16055
mar	BOU15014
mar	SAW16054
meg	VAG12056
meg	SAW17B10
meg	VAG12041
meg	BOU15027
meg	BOU15030
obs	VAG12061
obs	SAW16038
obs	SAW16042
obs	SAW16041
obs	SAW16032
bic	JAV11007
bic	JAV11015
bic	JAV11016
bic	JAV11022
mos	REU03026

of study, it may be worth checking whether some of the recently published large-scale time-calibrated trees [28–32] contain taxa from the study group, which could allow the transfer of age information. If there really is no published age estimate for any divergence event within the study group, a possible solution could be to extend the dataset by adding a closely related species for which a divergence time estimate is available. If this is also not feasible, perhaps because samples of outgroups are not available or too distantly related to map to the same reference, a last option could be to also generate mitochondrial sequence data for some of the species in the dataset and estimate their divergence times based on an assumed mitochondrial substitution rate. This would need to be done a priori in a separate phylogenetic analysis, for example with BEAST 2, and the uncertainty in the assumed substitution rate should be accounted for.

Once a divergence event is identified for which external age information is available, this age information needs to be expressed in the form of a prior probability distribution (simply called "prior" hereafter). The molecular clock model for SNAPP allows the same types of priors that are used by BEAST 2 more generally, including uniform, normal, lognormal, exponential, and gamma distributions. Each of these distributions are defined by a set of parameters, such as the lower and upper boundaries in the case of the uniform distribution or the mean and the standard deviation in the case of the normal distribution. In addition, "offsets" can be used to shift the entire distribution without modifying its shape. A good introduction to the various priors available in BEAST 2 and SNAPP is given in Drummond and Bouckaert [23]. For age constraints based on previous studies, the most suitable prior types are usually normal or lognormal distributions. If, for example, a previous study had found that the group for which SNP data are analyzed began to diverge around 10 million years ago (Ma) with a 95% confidence interval spanning from 9 to 11 Ma, a normal distribution with a mean of 10 and a standard deviation adjusted so that 95% of the probability mass lie between 9 and 11 would be a suitable prior for the age of the root of the SNP-based species tree. If, however, the previously reported confidence interval would be skewed with respect to the mean estimate, which is often the case for age estimates, a lognormal distribution could provide a better fit. For example, when the 95% confidence interval ranges from 9 to 13 Ma and the mean age estimate is 10 Ma, a normal distribution would not be able to accommodate the asymmetry of the estimate, but a lognormal distribution (e.g., with an offset of 8.5, a mean of 1.5, and a standard deviation of 0.55) could approximate it. Identifying the distribution parameter combination that best fits published age estimates may require some trial-and-error testing, aiming for a distribution that approximates both the mean and the confidence interval of the published estimate well. BEAUti's prior preview panel (in the "Priors" tab) may be of help for this testing, but some example data must first be loaded into BEAUti to be able to set an age constraint in this panel (Fig. 2).

With the divergence event identified and the type of prior and the distribution parameters selected, an age constraints input file for ***snapp_prep.rb*** can be written. Based on this file, the script can then translate the constraint to XML format and include it in the input file for SNAPP. The format of the age constraints file for ***snapp_prep.rb*** is relatively simple: For each constraint, a single line with three tab- or space-delimited elements is required (*see* **Note 1**). The first of these three elements specifies the type of the prior (normal, lognormal, uniform, or "CladeAge"; the latter type is described in Matschiner et al. [33]), followed by comma-separated parameter values in parentheses. For normal and a lognormal distributions, the parameters offset, mean (in real space in case of

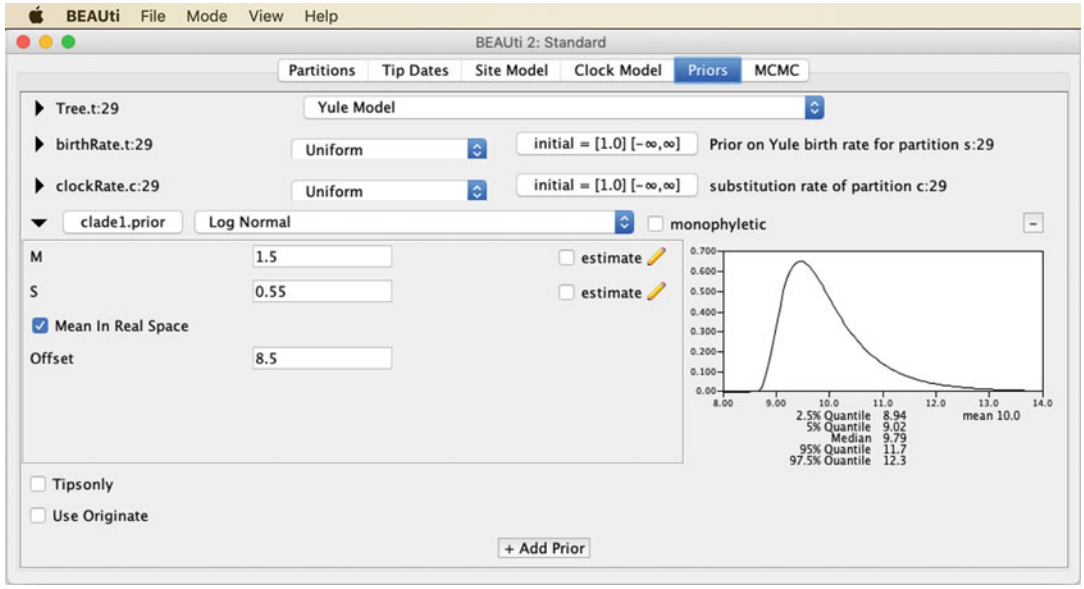


Fig. 2 Screenshot showing BEAUti's prior preview panel. With the chosen parameters, the lognormal prior has a mean of 10 (the sum of the offset, 8.5, and the distribution mean, 1.5) and 95% of the prior probability fall within the range from 8.94 to 12.3, the 2.5% and 97.5% quantiles

lognormal distributions), and standard deviation are expected, while for uniform distributions, only the lower and upper boundaries need to be specified (*see* **Notes 2** and **3**). The second element of the line should be either "crown" or "stem," depending on whether the age constraint should apply to the most recent common ancestor of the selected group ("crown") or the divergence of the group from its sister lineage ("stem"). The species IDs for members of the group should be specified separated by commas as the third element of the line; these should correspond to species IDs used in the species table. The following examples all show valid age constraints:

normal(0,10,0.5) crown speciesA,speciesB,speciesC

(a normally distributed constraint on the age of the most recent common ancestor of three species with a mean of 10 and a standard deviation of 0.5),

lognormal(8.5,1.5,0.55) stem speciesA,speciesB,speciesC

(a lognormally distributed constraint on the divergence time of three species from their sister lineage with an offset of 8.5, a mean of 1.5, and a standard deviation of 0.55),

uniform(10,15) crown speciesA,speciesB,speciesC

(a uniform constraint on the age of the most recent common ancestor of three species with a lower boundary of 10 and an upper boundary of 15).

In Barth et al. [27], a lognormal distribution was used to constrain the age of the most recent common ancestor of five species with an offset of 0, a mean of 13.76, and a standard deviation of 0.1, according to an earlier study based on mitochondrial sequences [34]:

lognormal(0,13.76,0.1) crown mar,meg,obs,bic,mos

Further information on constraint specification can be found in file example.con.txt, which is part of the snapp_prep GitHub repository (https://github.com/mmatschiner/snapp_prep).

2.7 Starting Tree

To initiate the MCMC chain, BEAST requires a starting tree. Usually, BEAST attempts to generate this starting tree itself; however, particularly when multiple age constraints are used, BEAST may not be able to produce a starting tree that is compatible with all constraints. If this is the case, BEAST immediately stops with an error message that includes the line "Fatal exception: Could not find a proper state to initialize" and also the line "P(prior) =--Infinity (was -Infinity)". When this problem occurs, it can be fixed by providing a starting tree in Newick format (http://evolution.genetics.washington.edu/phylip/newicktree.html) that is compatible with all specified constraints. To be readable by ***snapp_prep.rb***, the starting tree should be written to a file that contains only a single line and only the tree in Newick format on this line (*see* **Notes 4** and **5**). Note that besides the requirement that the tree should be compatible with all constraints, the topology and branch lengths chosen for the starting tree should not have any effect on the outcome of the SNAPP analysis and can therefore be chosen arbitrarily. For the dataset used by Barth et al. [27], a suitable starting tree would be the following.

((((mar:3,meg:3):3,obs:6):3,bic:9):3,mos:12);

To verify whether the starting tree is written as intended, the program FigTree (see below) can be used to visualize it.

2.8 Tracer

Tracer [24] is a very convenient and easy-to-use graphical user interface program for the assessment of MCMC stationarity and convergence. The program is available for Mac OS X, Linux, or Windows operating systems from GitHub at https://github.com/beast-dev/tracer/releases. While the instructions in this book chapter assume that Tracer is used to assess stationarity and convergence, it is worth pointing out the coda R package [35] as a useful alternative that also implements many of the functions available in Tracer.

2.9 FigTree

FigTree is a versatile graphical user interface program for the visualization of phylogenetic trees in Newick format. The program is available for Mac OS X, Linux, or Windows systems from GitHub at https://github.com/rambaut/figtree/releases.

3 Methods

3.1 Model

To reduce the computational demand of SNAPP, the method for divergence time estimation developed by Stange et al. [7] implements a model that is even more simplistic than the one used in standard SNAPP analyses. As in other SNAPP analyses, the Yule model [36] of lineage diversification is used, meaning that speciation events are assumed to occur with a constant rate per lineage and extinction is assumed to be absent. Also in common with other SNAPP analyses is the model assumption of a constant mutation rate that is identical in all lineages. While both assumptions are clearly violated by most or all empirical systems, it may be argued that at least within a system undergoing rapid diversification, the effects of extinction and rate variation may be small enough to be ignored. Going beyond the simplicity of standard SNAPP models is the assumption made in the model of Stange et al. [7] that all species have exactly the same population size. Even for recently diverged lineages, this assumption is rather unrealistic [12, 37], but as ancestral population sizes are inherently difficult to estimate and SNAPP analyses would otherwise hardly be possible for datasets of more than ten species, the assumption may nevertheless often be justified. Further reduction of model complexity is achieved by linking the forward and reverse mutation rates, which is not the case in standard SNAPP analyses.

Besides these model simplifications, the method of Stange et al. differs from standard SNAPP analyses also in the choice of priors. To both of the two parameters speciation rate (λ) and clock rate (μ), a scale-independent one-over-x prior is applied. The advantage of this is that the prior works equally well with young or old groups of species and no group-specific adjustments from the user are required. This is also the case for the prior on the population size parameter Θ, for which a very wide and therefore essentially uninformative uniform distribution is used. The model developed by Stange et al., including the above-described priors, is automatically selected when the XML file for SNAPP is written with the ***snapp_prep.rb*** script. For most users of divergence time estimation with SNAPP, no further modifications to the XML file will be necessary.

3.2 Generating the XML File with snapp_prep.rb

The minimum input required by ***snapp_prep.rb*** are three files: the one with the genotype data matrix, the file with the species assignment table, and the file with age constraints. If these are named matrix.vcf, species.txt, and constraints.txt, an XML file can be generated with ***snapp_prep.rb*** using the command:

```
ruby snapp_prep.rb -v matrix.vcf -t species.txt -c con-
straints.txt
```

This command would use all biallelic SNPs with sufficiently complete data, it would specify the default run length of 500,000 MCMC iterations, it would write an XML file with the default name snapp.xml, and it would set the output files of the SNAPP analysis to be named snapp.log and snapp.trees. A different number of MCMC iterations could be specified with the -l option (e.g., -l 100000), and smaller numbers of iterations might be advisable in initial analyses to explore the run time per iteration and how fast the MCMC chain approaches stationarity. The name of the XML file could be changed with the -x option, and different names for SNAPP's output files could be set with the -o option. A file with a starting tree could additionally be provided with the -s option, which may be helpful when BEAST is unable to generate a suitable starting tree itself. An overview of all available options can be displayed with the command:

```
ruby snapp_prep.rb -h
```

Some of the further options may be useful:

The relative weight of topology operators, and with it the frequency at which SNAPP attempts to change the tree topology during MCMC, can be changed with the -w option. The default for this option is 1; with values smaller or larger than 1, SNAPP will attempt to change the topology less frequently or more frequently, respectively, than other parameters. This option may be particularly useful when the user would like to fully fix the tree topology to the topology of a starting tree, which can be done by setting the relative weight to zero with -w 0.

To gain better control of the computational demand of the SNAPP analysis, a maximum number of SNPs can be specified with the -m option. When this option is used, the specified number of SNPs will be randomly selected from all those that are suitable for SNAPP. Similarly, a minimum distance between SNPs can be set with the -q option to reduce the potential effect of linkage among sites.

The effects of these two options are identical to those achieved by reducing and thinning the input VCF file a priori, but it may be more convenient to apply these filters with ***snapp_prep.rb*** because other tools cannot easily discriminate between SNPs suitable for SNAPP (e.g., those that have data for at least one individual per species) and those that will need to be excluded anyway.

While rates of mutations are well known to vary depending on the types of nucleotides that are exchanged [38], SNAPP does not model rate variation. One practical way to account at least partially for varying rates among nucleotide pairs is to reduce the genotype matrix to only transitions or only transversions, given that most rate variation is usually partitioned between these classes rather than within them [39]. This reduction can be done with the -i option to

include only transitions or with the -r option to include only transversions. When unsure which of the two classes of mutations to use, two separate XML files could be produced and SNAPP analyses could be performed separately with both files, allowing an assessment of the robustness of the results to these data subsets.

3.3 MCMC with BEAST

To perform SNAPP analyses with the XML file written with *snapp_prep.rb*, this file needs to be provided as input to BEAST. This can be done either using the command-line version of BEAST or its graphical user interface. If the XML file is named snapp.xml and BEAST is located in /Applications/BEAST/, the command-line version can be used to start MCMC with the command:

```
/Applications/BEAST/bin/beast snapp.xml
```

As SNAPP analyses can be parallelized very efficiently if multiple processors are available, the use of threading is recommended. On the command line, the use of multiple threads can be specified with the -threads option. For example, four threads can be used with the command:

```
/Applications/BEAST/bin/beast -threads 4 snapp.xml
```

The graphical user interface of BEAST can be launched by double-clicking on the program icon, which should open two windows as shown in Fig. 3. The input file can then be loaded by clicking on "Choose File ...", the number of threads can be selected from the drop-down menu next to "Thread pool size," and the MCMC chain can be started by clicking "Run."

During MCMC, BEAST's screen output shows values in eight columns that represent the current MCMC iteration, the posterior probability for this iteration, the cumulative effective sample size (ESS; see below) for the posterior probability, and the likelihood, the prior probability, the tree height (the age of the root of the tree), and the clock rate for this iteration. The last column at first only shows "--" but this is replaced after a certain number of iterations with an estimate of the required run time for one million iterations, as shown in Fig. 4.

It is worth following the screen output for some time. If the ESS value for the posterior increases above 200, the MCMC chain may have reached stationarity and a further extension to the chain may not be required. To verify stationarity, the output file with the ".log" filename extension should be inspected as described in the next section. If, on the other hand, the ESS value remains very low for a long time, stationarity may be difficult to reach and a restart of the analysis with a smaller dataset, or the use of a larger number of threads, should be considered. It is not uncommon for SNAPP analyses to require hours or days to finish, and in some cases the

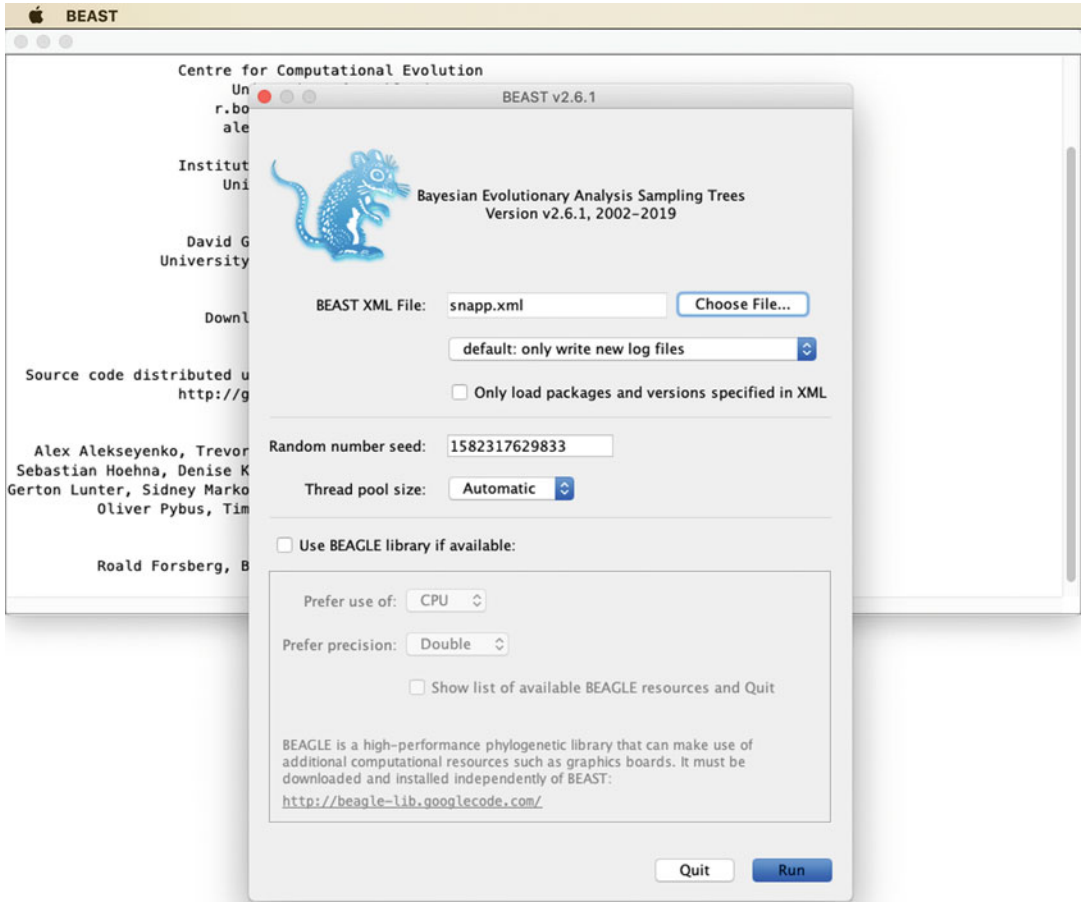


Fig. 3 Screenshot showing the file opening dialog of BEAST's graphical user interface

analysis may take weeks. Ultimately, whether or not the completion of a SNAPP analysis with a given dataset is feasible may depend on the patience of the user and the long-term access to computational resources with multiple processors.

3.4 Assessing Stationarity and Convergence with Tracer

During MCMC, BEAST writes two output files with the ".log" and ".trees" filename extensions. At the end of MCMC, these files should each contain output describing the state of the MCMC chain for 2000 iterations sampled at regular intervals. The output is divided so that all model parameters except the tree and the population size parameter Θ are written to the file with the ".log" extension while the tree, including branch lengths, and Θ are written in annotated Newick format to the file with the ".trees" extension. To assess MCMC chain stationarity, and thus whether or not the analysis should be extended, the file with the ".log" extension should be inspected with the program Tracer.

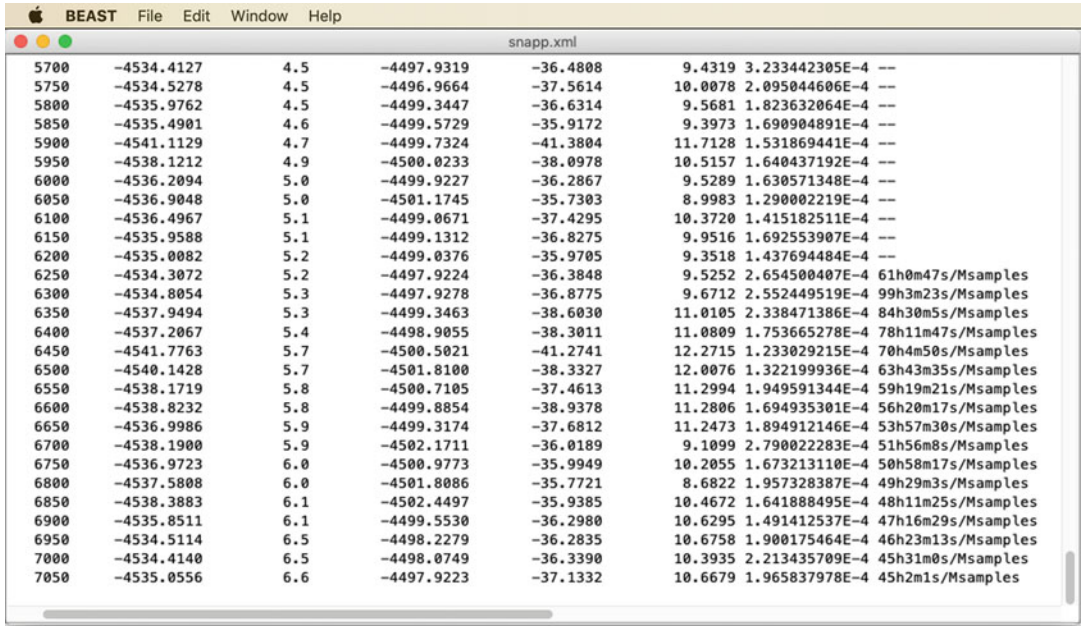


Fig. 4 Screenshot showing the SNAPP screen output in BEAST's graphical user interface

The many ways in which Tracer can be used to analyze MCMC results are described well in its publication [24] and in Drummond and Bouckaert [23]. In brief, Tracer is used to assess whether or not the MCMC chain has run long enough to allow conclusions, to adjust the length of the part of the MCMC chain that is considered as burn-in, and to extract parameter estimates and their confidence intervals. To determine that the chain was sufficiently long, stationarity and convergence must have been reached, indicating that the MCMC chain has sampled from the true posterior distribution. The first of these two criteria—stationarity—can be assumed when trends are no longer recognizable in trace plots of the sampled posterior probability, the likelihood, the prior probability, and all parameter values. One such trace plot, showing samples of the posterior probability, is illustrated in Fig. 5. Perhaps the most important measures of MCMC stationarity are the ESS values that are listed for posterior and prior probabilities, the likelihood, and parameter estimates in the bottom left panel of the Tracer window. These quantify the number of effectively independent samples drawn from the posterior distribution and thus account for autocorrelation in estimates sampled throughout the MCMC chain. As a rule of thumb, all ESS values should be greater than 200 before the MCMC chain can be considered stationary, but even larger values are preferable as they allow better estimates of confidence intervals [23]. To point out problematic estimates, Tracer marks ESS values smaller than 200 in red (ESS < 100) or yellow ($100 \leq \text{ESS} < 200$).

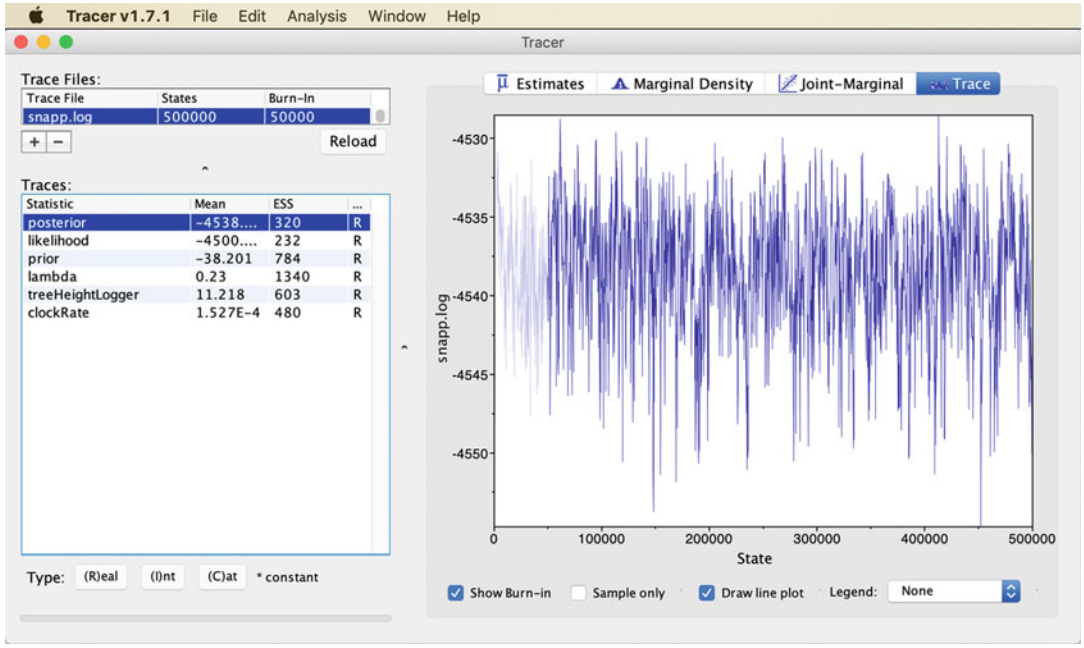


Fig. 5 Screenshot showing the Tracer window with the trace plot of the posterior probability. Trace plots can be displayed by selecting a statistic in the lower left panel and clicking the "Trace" button at the top right of the window

Even when the MCMC chain appears stationary based on visual inspection of trace plots and the ESS values, it may nevertheless not sample from the true posterior distribution. This is possible when the posterior probability surface has multiple peaks and the MCMC chain only explored a peak that is not the highest peak overall. While the probability surface may rarely be complex enough for this to happen with the simple models used in SNAPP, it is important to exclude this possibility. A good way to do so is to run multiple replicate analyses with the same XML file and verify that the MCMC chains in these analyses converge, meaning that they all arrive at roughly the same estimates even though they had different starting points (which is the case unless the same random number seed is reused).

To verify MCMC chain convergence, the files with ".log" extensions resulting from the multiple replicate analyses can be loaded jointly into Tracer. Below the names of these files in the top left panel of the Tracer window, an entry named "Combined" should then appear. When this entry is selected, trace plots will display the combined MCMC chain from the multiple files (excluding the burn-in parts of the individual chains), and all ESS values will be recalculated for the combined chain. If one or more of the run replicates did not converge, this will be obvious from marked steps in the trace plots and substantial decreases of ESS values.

If MCMC chains have reached stationarity and convergence, it may be worth optimizing the percentage of the chain that is considered as burn-in and thus excluded from the calculation of parameter estimates. With the default settings, the number of burn-in samples is specified as 50,000 in the column titled "Burn-in" in the top left panel of the Tracer window (Fig. 5), corresponding to 10% of the default chain length of 500,000 iterations. The length of the burn-in could be increased to, for example, 20%, by clicking on "50000" and writing "100000" instead. Adjusting the burn-in length is advisable if larger burn-in percentages improve the ESS values and the visual appearance of stationarity in trace plots.

If multiple MCMC replicates were performed (and all have converged), downstream analyses can be simplified by combining the result files from these replicates. This can be done separately for the files with the ".log" extension and for the files with the ".trees" extension, using the LogCombiner tool from the BEAST 2 suite of programs. The graphical user interface of LogCombiner can be used intuitively to load multiple input files, specify burn-in percentages for each of these, and set the name of the combined output file.

3.5 Obtaining Parameter Estimates with Tracer

Besides the posterior probability, the likelihood, and the prior probability, Tracer shows only three parameters in the lower left panel (if the XML was prepared with ***snapp_prep.rb***): "lambda," "treeHeightLogger," and "clockRate." Of these, "lambda" refers to the speciation rate (λ), "treeHeightLogger" refers to the age of the most recent common ancestor in the tree (thus, it is the sum of multiple branch lengths rather than a parameter itself), and "clockRate" refers to the rate of the molecular clock (μ). As discussed in Stange et al., the clock rate is subject to ascertainment bias when the dataset includes only SNPs and should not be directly interpreted as the mutation rate. However, as also shown in Stange et al., the clock rate estimate, together with the estimate for the population size parameter Θ, can serve to accurately estimate the effective population size N_e, given that $\Theta = 4N_e\mu g$ (with g being the generation time).

To add an estimate of N_e to a new file with ".log" extension that can be read by Tracer, the Ruby script ***add_theta_to_log.rb***, from the same GitHub repository as ***snapp_prep.rb***, can be used. This script reads the sampled clock rates from the result file with the ".log" ending and the sampled Θ values from the file with the ".trees" ending, calculates N_e from these values and a user-specified generation time, and writes a new file with the ".log" extension that is identical to the first except that it also contains samples for Θ and N_e. For example, with the result files snapp.log and snapp.trees and a generation time of 3 years, the script could be run with the command:

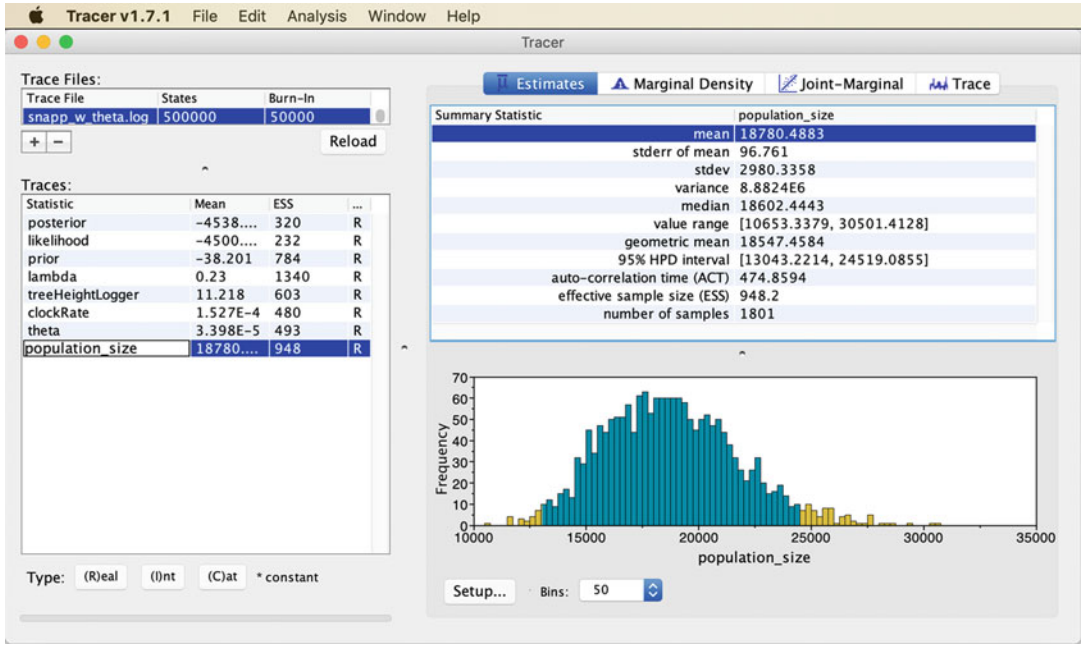


Fig. 6 Screenshot showing the Tracer window with summary statistics for the estimates of the effective population size (N_e)

```
ruby add_theta_to_log.rb -l snapp.log -t snapp.trees -g 3
```

This command would write an output file with the default name snapp_w_theta.log; other names could be specified with the -o option. Opening the output file of ***add_theta_to_log.rb*** in Tracer should show that two entries named "theta" (Θ) and "population_size" (N_e) have been added to the list of parameters in the lower left panel of the window (Fig. 6). Both of these parameters should also be checked for stationarity, but as the estimates for Θ are subject to the same ascertainment bias as those for the clock rate (μ), only the estimates for the population size should be interpreted. Selecting "population_size" in the parameter list in Tracer and clicking the "Estimates" button at the top center of the window should show summary statistics for this parameter, including the mean estimate, the standard deviation, and the 95% highest posterior density (HPD) interval, which in Bayesian analyses serves as the confidence interval (Fig. 6).

3.6 Generating a Summary Tree with TreeAnnotator

Opening the output file of SNAPP with the ".trees" extension in FigTree allows the user to view all the trees sampled during MCMC one by one. Alternatively, all sampled trees could be displayed simultaneously with DensiTree [40], another tool that is included in the BEAST 2 suite of programs. Often, however, a single tree summarizing the information from all sampled trees is required.

Such summary trees can be generated with TreeAnnotator, which identifies the most credible tree topology and representative clade ages based on criteria selected by the user [41]. For the tree topology, either "maximum clade credibility tree" or "maximum sum of clade credibilities" can be chosen; these two options select the tree topology for which either the product or the sum, respectively, of all node support values is highest. The clade ages, on the other hand, can be set either to the mean or the median of each clade's age in all posterior trees that contain this clade. Alternatively, "Common Ancestor heights" can be chosen, which calculates clade ages from all posterior trees, not just those that contain the clade [41]. For most species trees generated with SNAPP, these options should have rather little effect, given that SNAPP trees are usually well-supported and not overly species-rich. Perhaps the most commonly used setting is to produce a maximum clade credibility tree with mean node heights, which should work well for all SNAPP trees. Besides these options, the burn-in percentage should be specified (unless the burn-in part of the MCMC has already been removed, e.g., with LogCombiner), and input and output file names must be given. It is convenient to name the output file exactly like the input tree file, except that the ".trees" file extension is replaced with ".tre".

3.7 Visualizing the Summary Tree in FigTree

After opening the summary tree in FigTree, the program has various options to customize the tree's visualization. These options are accessible from the menu on the left of the FigTree window, within several panels that can be opened by clicking on the triangles and activated by checking the boxes next to these. Generally useful are the following options.

- Uncheck "Scale Bar" but check "Scale Axis," open the panel for "Scale Axis," uncheck "Show grid," and check "Reverse Axis." This adds a time scale in units of millions of years before present.
- Check and open the "Node Labels" panel, then set the drop-down menu next to "Display" to "posterior." This shows the support values for each node in the form of Bayesian posterior probabilities (BPP).
- Check and open the "Node Bars" panel, then set the drop-down menu next to "Display" to "height_95%_HPD". This adds blue bars to each node indicating the confidence interval for its age.

After the tree visualization has been adjusted as described above, a publication-ready figure of the species tree can be exported in PDF format via FigTree's "File" menu.

4 Notes

Issues encountered by users of ***snapp_prep.rb*** are often related to the preparation of the age constraints or species table input files. The following points should be considered if any issues arise in the preparation of these files:

1. If an error message or the resulting estimate of the species tree indicate that the constraints file may not have been read properly by ***snapp_prep.rb***, it may be worth checking that no spaces are included at the very beginning of the line defining the constraint. The same issue can result when the old Mac OS 9 format for line endings is accidentally used in the constraints file.
2. When using normal prior distributions for age constraints, the offset is redundant with the mean; thus, one of the two distribution parameters can always be set to zero.
3. Normal prior distributions may in some cases not work as well as lognormal distributions, because their tails are not bounded in either direction and therefore they do assign a certain prior probability to an age of zero (and even to negative ages). Through the interaction with the priors on the speciation rate and the population size, this may cause the MCMC chain to move toward a tree age of zero. When this issue is encountered, it can be fixed by replacing the normal prior distribution with a similarly shaped lognormal distribution.
4. The individual IDs used in the species table should match those used in the genotype data matrix exactly. No individual IDs should be used only in one of the two files but not the other. Similarly, the species IDs used in the species table should exactly match those in the starting tree if a starting tree is provided. Finally, the IDs used in the definition of age constraints should be species IDs, not individual IDs.
5. If a starting tree is provided, this tree should not contain nodes with only one descendant, and no branch should be included above the root of the tree. One way to test for unintended nodes and branches is to open the tree in FigTree and activate the "Node Shapes" panel, which marks all nodes with circles or other symbols.

If any other issues should arise, I recommend that questions related to SNAPP are posted on the BEAST user group (https://groups.google.com/forum/#!forum/beast-users) while questions related to ***snapp_prep.rb*** should be directed to me by email.

Acknowledgments

I thank Julie Lee-Yaw, Amanda Haponski, Livia Loureiro, Sue Sherman-Broyles, Bohao Fang, Yayan Kusuma, Daniel Poveda-Martínez, Xiaoxi Yang, Cecilia Fiorini, Kristen Finch, Armel Donkpegan, Marta Liber, Jie Gao, and Julia Canitz for testing the ***snapp_prep.rb*** script. Funding was provided by the Research Council of Norway (FRIPRO 275869).

References

1. Kubatko LS, Degnan JH (2007) Inconsistency of phylogenetic estimates from concatenated data under coalescence. Syst Biol 56:17–24
2. Leaché AD, Rannala B (2011) The accuracy of species tree estimation under simulation: a comparison of methods. Syst Biol 60:126–137
3. Liu L, Edwards SV (2009) Phylogenetic analysis in the anomaly zone. Syst Biol 58:452–460
4. Degnan JH, DeGiorgio M, Bryant D, Rosenberg NA (2009) Properties of consensus methods for inferring species trees from gene trees. Syst Biol 58:35–54
5. Roch S, Steel M (2014) Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. Theor Popul Biol 100:56–62
6. Ogilvie HA, Bouckaert RR, Drummond AJ (2017) StarBEAST2 brings faster species tree inference and accurate estimates of substitution rates. Mol Biol Evol 34:2101–2114
7. Stange M, Sánchez-Villagra MR, Salzburger W, Matschiner M (2018) Bayesian divergence-time estimation with genome-wide SNP data of sea catfishes (Ariidae) supports Miocene closure of the Panamanian Isthmus. Syst Biol 67:681–699
8. Maddison WP (1997) Gene trees in species trees. Syst Biol 46:523–536
9. Liu L (2008) BEST: Bayesian estimation of species trees under the coalescent model. Bioinformatics 24:2542–2543
10. Edwards SV (2009) Is a new and general theory of molecular systematics emerging? Evolution 63:1–19
11. Kubatko LS, Carstens BC, Knowles LL (2009) STEM: species tree estimation using maximum likelihood for gene trees under coalescence. Bioinformatics 25:971–973
12. Heled J, Drummond AJ (2010) Bayesian inference of species trees from multilocus data. Mol Biol Evol 27:570–580
13. Yang Z (2015) The BPP program for species tree estimation and species delimitation. Curr Zool 61:854–865
14. Zhang C, Rabiee M, Sayyari E, Mirarab S (2018) ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. BMC Bioinformatics 19:153
15. Edwards SV, Xi Z, Janke A et al (2016) Implementing and testing the multispecies coalescent model: a valuable paradigm for phylogenomics. Mol Phylogenet Evol 94: 447–462
16. Springer MS, Gatesy J (2016) The gene tree delusion. Mol Phylogenet Evol 94:1–33
17. Chifman J, Kubatko LS (2014) Quartet inference from SNP data under the coalescent model. Bioinformatics 30:3317–3324
18. Bryant D, Bouckaert RR, Felsenstein J, Rosenberg NA, RoyChoudhury A (2012) Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis. Mol Biol Evol 29:1917–1932
19. De Maio N, Schrempf D, Kosiol C (2015) PoMo: an allele frequency-based approach for species tree estimation. Syst Biol 64:1018–1031
20. Stoltz M, Bauemer B, Bouckaert R et al (2021) Bayesian inference of species trees using diffusion models. Syst Biol 70:145–161
21. Leaché AD, Fujita MK, Minin VN, Bouckaert RR (2014) Species delimitation using genome-wide SNP data. Syst Biol 63:534–542
22. Bouckaert RR, Vaughan TG, Barido-Sottani J et al (2019) BEAST 2.5: an advanced software platform for Bayesian evolutionary analysis. PLoS Comput Biol 15:e1006650
23. Drummond AJ, Bouckaert RR (2015) Bayesian evolutionary analysis with BEAST 2. Cambridge University Press, Cambridge
24. Rambaut A, Drummond AJ, Xie D, Baele G, Suchard MA (2018) Posterior summarization

in Bayesian phylogenetics using Tracer 1.7. Syst Biol 67:901–904

25. Drummond AJ, Nicholls GK, Rodrigo AG, Solomon W (2002) Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. Genetics 161:1307–1320
26. Felsenstein J (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol 17:368–376
27. Barth JMI, Gubili C, Matschiner M et al (2020) Stable species boundaries despite ten million years of hybridization in tropical eels. Nat Commun 11:1433
28. Kumar S, Stecher G, Suleski M, Hedges SB (2017) TimeTree: a resource for timelines, timetrees, and divergence times. Mol Biol Evol 34:1812–1819
29. Fernández R, Kallal RJ, Dimitrov D et al (2018) Phylogenomics, diversification dynamics, and comparative transcriptomics across the spider Tree of Life. Curr Biol 28:1489–1497
30. Rabosky DL, Chang J, Title PO et al (2018) An inverse latitudinal gradient in speciation rate for marine fishes. Nature 559:392–395
31. Upham NS, Esselstyn JA, Jetz W (2019) Inferring the mammal tree: species-level sets of phylogenies for questions in ecology, evolution, and conservation. PLoS Biol 17:e3000494
32. Janssens S, Couvreur TLP, Mertens A et al (2020) A large-scale species level dated angiosperm phylogeny for evolutionary and ecological analyses. Biodiv Data J 8:e39677
33. Matschiner M, Musilova Z, Barth JMI et al (2017) Bayesian phylogenetic estimation of clade ages supports trans-Atlantic dispersal of cichlid fishes. Syst Biol 66:3–22
34. Jacobsen MW, Pujolar JM, Gilbert MTP et al (2014) Speciation and demographic history of Atlantic eels (*Anguilla anguilla* and *A. rostrata*) revealed by mitogenome sequencing. Heredity 113:432–442
35. Plummer M, Best N, Cowles K, Vines K (2006) CODA: convergence diagnosis and output analysis for MCMC. R News 6:7–11
36. Yule GU (1925) A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F.R.S. Phil Trans R Soc Lond B 213:21–87
37. Genner MJ, Turner GF (2014) Timing of population expansions within the Lake Malawi haplochromine cichlid fish radiation. Hydrobiologia 748:121–132
38. Yang Z (1994) Estimating the pattern of nucleotide substitution. J Mol Evol 39:105–111
39. Brown WM, Prager EM, Wang A, Wilson AC (1982) Mitochondrial DNA sequences of primates: tempo and mode of evolution. J Mol Evol 18:225–239
40. Bouckaert RR (2010) DensiTree: making sense of sets of phylogenetic trees. Bioinformatics 26: 1372–1373
41. Heled J, Bouckaert RR (2013) Looking for trees in the forest: summary tree from posterior samples. BMC Evol Biol 13:211

Chapter 3

High-Throughput Evolutionary Comparative Analysis of Long Intergenic Noncoding RNAs in Multiple Organisms

Anna C. Nelson Dittrich and Andrew D. L. Nelson

Abstract

Comparative genomic and transcriptomic analyses can help prioritize and facilitate the functional analysis of long noncoding RNAs (lncRNAs). Evolinc-II is a bioinformatic pipeline that automates comparative analyses, searching for sequence and structural conservation for thousands of lncRNAs at once. In addition, Evolinc-II takes a phylogenetic approach to infer key evolutionary events that may have occurred during the emergence of each query lncRNA. Here, we describe how to use command line or GUI (CyVerse's Discovery Environment) versions of Evolinc-II to identify lncRNA homologs and prioritize them for functional analysis.

Key words Comparative genomics, Long intergenic noncoding RNAs, RNA evolution, Genome annotation, Functional genomics, RNA structure, Bioinformatics, Docker, CyVerse

1 Introduction

Long noncoding RNAs (lncRNAs) are a large but poorly characterized class of transcripts. Their (on average) low-level transcription and noncoding nature meant that they were ignored or remained hidden during initial genome annotation efforts. However, advances in genome-wide transcriptome analyses, alongside high throughput data analysis pipelines, have facilitated the discovery of thousands of putative lncRNAs in every major eukaryotic lineage. Long-read RNA-seq has further improved lncRNA identification by removing transcript assembly artifacts common with second generation short read sequencing. Thus, with identification trivialized, lncRNA functional annotation now serves as the outstanding challenge in the field of lncRNA biology.

Several strategies have been proposed to assign putative function and prioritize lncRNA in vivo analyses. Given their known roles as transcriptional regulators and molecular decoys, emphasis has been placed on bioinformatic guilt-by-association techniques

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_3,

and predictions of lncRNA/DNA/RNA/protein interactions [1–7]. Given sufficient sequencing resolution or structural models, these data can narrow down the list of putative lncRNA regulatory (or physical) interactors.

In vertebrate, yeast, and to a lesser degree, plant systems, comparative genomic and transcriptomic analyses have also been utilized to assess lncRNA conservation [8–11]. Conservation of synteny (location within a genome), sequence, structure, or context-dependent transcription are interpreted as indicators of conservation of function. A deeply conserved lncRNA is a stronger candidate for functional analysis than a poorly conserved lncRNA, particularly as conservation (synteny, sequence, or structural) can help guide functional hypotheses. While conservation is a useful metric, few pipelines exist to assess lncRNA conservation in high throughput. To rectify this shortcoming, the Evolinc computational suite was developed [12]. Evolinc is composed of two pipelines, one for lncRNA identification and one for lncRNA evolutionary analyses (Evolinc-I and -II, respectively). Here we describe, in detail, how to use Evolinc-II in the command line or within CyVerse's Discovery Environment. We also describe improvements to Evolinc-II that were not in the original release, including speed enhancements and additional optional analyses.

2 Materials

2.1 Software and Test/Sample Data

1. The Docker container for running Evolinc-II on the command line can be found at https://hub.docker.com/r/evolinc/evolinc-ii and pulled using the Docker command: "docker pull evolinc/evolinc-ii".
2. The Evolinc-II app can be found within CyVerse's Discovery Environment [13] by searching for Evolinc-II in the Apps menu and clicking on the latest version (v2.0).
3. Example test data can be retrieved from the following locations for command line or Discovery Environment analyses, respectively.
   https://github.com/Evolinc/Evolinc-II/releases/download/v1.0/sample_data.zip
   or
   iplant/home/shared/iplantcollaborative/example_data/Evolinc.sample.data/Evolinc-II.
4. Evolinc-II minimally requires the following input information.
   (a) Query_lincRNAs.fasta: A FASTA file of query lincRNA sequences (*see* **Note 1**).

Table 1
Example species list with four-letter abbreviation codes

Scientific name	Evolinc-II code
Arabidopsis thaliana	Atha
Arabidopsis lyrata	Alyr
Capsella rubella	Crub
Brassica rapa	Brap
Schrenkiella parvula	Spar
Eutrema salsugineum	Esal
Aethionema arabicum	Aara

Table 2
The Evolinc-II startup list. The startup list denotes the order in which the comparisons will be performed. This is a tab-delimited file. All of these files should be saved in the input folder. (I) Name of the genome/transcriptome files used in the analysis. (II) Name of the query lincRNA file. (III) Four-letter code for the query species. (IV) The species to be scanned against, which includes the query species. (V) The names of the genome annotation files (KG = known genes) for each species. (VI) The name of the known lincRNA (KL) files, where available. Different combinations of files can be added to the startup list. For instance, if a KG file is not present for a particular species but a KL is, then leave column "v" blank. Do not place the KL file in the KG column

I	II	III	IV	V	VI
Atha_genome.fasta	query_lincRNA.fasta	Atha	Atha	Atha_KG.gff	
Alyr_genome.fasta	query_lincRNA.fasta	Atha	Alyr	Alyr_KG.gff	Alyr_KL.fasta
Crub_genome.fasta	query_lincRNA.fasta	Atha	Crub	Crub_KG.gff	Crub_KL.fasta
Brap_genome.fasta	query_lincRNA.fasta	Atha	Brap	Brap_KG.gff	
Spar_genome.fasta	query_lincRNA.fasta	Atha	Spar	Spar_KG.gff	Spar_KL.fasta
Esal_genome.fasta	query_lincRNA.fasta	Atha	Esal	Esal_KG.gff	
Aara_genome.fasta	query_lincRNA.fasta	Atha	Aara		

(b) Genome.fasta: A FASTA file for the genome of each species to be queried, including the genome of the species from which the query lincRNAs originate.

(c) Species_list.txt: A single column text file with all species listed in order of phylogenetic relatedness to the query species (*see* Table 1).

(d) Startup_list.txt: A tab-delimited text file containing file names that delineate the order of operations for Evolinc-II. Evolinc-II uses this file to iterate through all of its analyses (*see* Table 2).

5. **Additional** input information include the following.
   (a) Genome_annotation.gff/gff3/gtf: Annotation files corresponding to the genomes to be queried listing the known genes in their respective species. These file names should be incorporated into the Startup_list file in the fourth column (*see* Table 2 and **Note 2**).
   (b) Known_lincRNAs.fasta: A FASTA file of any lincRNAs that might be known for a particular subject species but not currently incorporated into the genome annotation file (e.g., identified by the user with Evolinc-I). The names of these files should be incorporated into the Comparison_list file in the fifth column.
   (c) Species_tree.txt: A species tree (in Newick format) describing the relationship of all the species to be queried.

2.2 Computational Requirements

2.2.1 Requirements to Run Evolinc-II on the Command Line

1. Evolinc-II is prepackaged in a Docker container (https://docs.docker.com/) for ease of use. As such, Evolinc-II is compatible with a PC or a server (with super-user privileges) where Docker is installed, or on an HPC where Singularity (https://singularity.lbl.gov/) is installed. This also means that Evolinc-II can be run under any operating system (Windows/Mac/Linux). No software dependencies other than Docker/Singularity are required.
2. Storage requirements: In general, the majority of space requirements for Evolinc-II are associated with the query and subject genomes. An additional 1 GB will be more than sufficient for analyses.
3. CPU requirements: The basic homolog identification step of Evolinc-II is relatively fast with minimal CPU requirements (e.g., homologs for 5000 lncRNAs can be identified in <10 genomes with four CPUs in <30 min). However, the optional structural and phylogenetic assessments greatly benefit from additional CPU support (8+) and will take longer to run (~8 h for the sample dataset with 8 cores).

2.2.2 Requirements to Run Evolinc-II in CyVerse's Discovery Environment

The Evolinc-II app is currently integrated in CyVerse's Discovery Environment and is free to use by researchers. If you do not have access to a high performance computing cluster we highly recommend using Evolinc-II within the DE. Limited computing resources can lead to long Evolinc-II run times, especially when searching through large genomes and with the optional steps added. Not only is the CyVerse DE free to researchers, but many of the genomes are already integrated (and almost all others are available through CoGe; genomevolution.org) [14, 15], thereby alleviating the data storage hurdle common with these types of analyses.

A free CyVerse user account is required to run Evolinc-II in the Discovery Environment. More information on setting up a Discovery Environment user account can be found at https://user.cyverse.org/.

In addition, an up-to-date Java-enabled web browser and an Internet connection are required.

3 Methods

3.1 General Overview of Evolinc-II

Regardless of the method by which the user launches a Evolinc-II job, the pipeline follows the same key steps (Fig. 1, panels a–d). A set of lincRNAs (1–1000 s) are used as query in searches against the genomes (or, if genomes are unavailable, transcriptomes in FASTA format) of all species of interest. Putative sequence homologs are then used as query in a reciprocal search of the query genome to help remove spurious returns (*see* **Note 3**). Families of sequence homologs are built and then annotated based on any additional genome annotation or known lncRNA files that the user provides. Overlap with known genes or known lncRNAs is used as a proxy for evidence of transcription of the homologous locus in target species. Following family building, a standard set of results are generated irrespective of whether the optional steps were included. These results are described in Subheading 3.4.1, but in general allow the user to infer lncRNA origins, identify gene IDs associated with their sequence homologs, and determine the degree to which their set of lncRNAs are conserved across their tested species (Fig. 1, panels e–h). The optional structural and evolutionary analyses will occur if selected and if sufficient taxa ($n \geq 4$) are available in a given family of sequence homologs (Fig. 1, panels e, f; Figs. 2 and 3). The lncRNA structural analysis step (Fig. 2) infers RNA structure from multiple sequence alignments (MSAs) using the LocARNA package [16, 17]. The evolutionary analysis step builds phylogenies using RAxML [18] and infers duplication and loss histories within each lncRNA family using Notung (Fig. 3) [19]. The results from each of these steps are described in Subheading 3.4.2.

3.2 Launching Evolinc-II on the Command Line

1. To prepare the input data, place all of the files to be used in the analyses in a single directory with read/write access.
2. Create unique four letter Genus/species identifiers for each species (*see* Table 1 for the abbreviations related to the sample data).
3. Create a single column text file with all of the unique species identifiers, with the query species listed first and the following species listed in order of the relationship to the query species. The right column of Table 1, without the header, depicts how the species list file should look. A phylogenetically correct list is

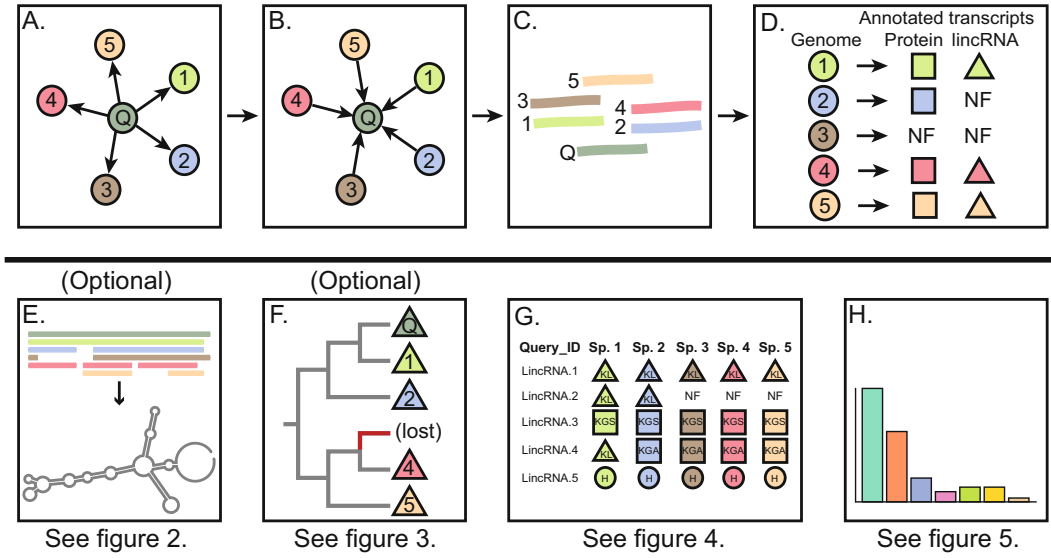


Fig. 1 The Evolinc-II workflow. (**a**) The initial homology search. Using 1–1000 s of user provided lncRNAs as query (Q), Evolinc-II scans the genomes (or transcriptomes) of target species to identify sequence homologs with significant conservation and coverage. (**b**) Reciprocal homolog search. Putative sequence homologs identified in (**a**) are used as query against the original query species. (**c**) Sequences that return the original query lncRNA as the top hit are then extracted and clustered with other top hits into families of lncRNA sequence homologs. Note that this can include the sequence homolog with the highest sequence similarity as well as additional, secondary homologs that still meet similarity thresholds (potential paralogs). (**d**) Using the provided genome annotation and known lncRNA files (both optional), Evolinc-II will assign protein-coding or lncRNA IDs to corresponding sequence homologs. In the absence of prior annotations (NF), sequence homologs will simply be labeled as homologs. (**e**) If selected by the user, Evolinc-II will infer the most likely structure based on a multiple sequence alignment of all primary sequence homologs. (**f**) Also, optionally, Evolinc-II will infer phylogenies and then use these phylogenies to assess the evolutionary history of each lncRNA family. (**g**, **h**). Finally, Evolinc-II generates a set of results to aid the user in identifying deeply conserved or potentially functionally interesting lncRNAs. Information about sequence homologs, including IDs of corresponding known genes (mRNAs/lncRNAs), which are located in the summary table in the results, can be used to infer lncRNA origins (**g**). In addition, plots depicting homolog recovery and frequency with which lncRNA families harbor sequence homologs from particular species are generated

not absolutely necessary if some species relationships are not known.

4. Simplify the names of the files and add unique species identifiers to each genome, genome annotation, and known lincRNA file (e.g., A_thaliana.Araport11_genome.fa becomes "Atha_genome.fa").
5. Create a "Startup_list.txt" file that delineates the comparisons Evolinc-II will perform. *See* Table 2 for an example and **Note 4** for additional information. If wishing to include the phylogenetic/structural step, generate a species tree (in Newick format). An example can be found in the example data.

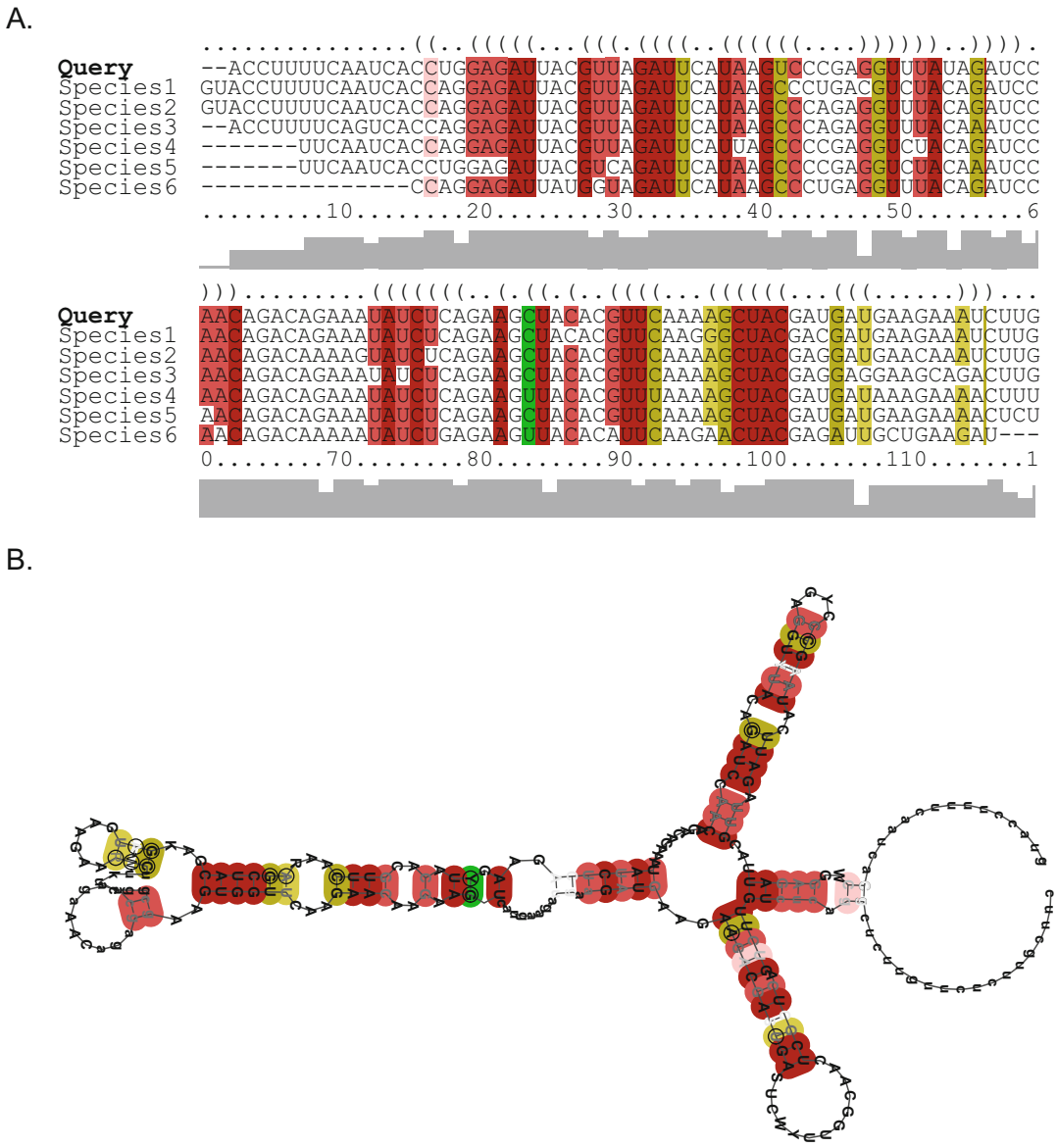


Fig. 2 LncRNA structural analysis. The optional structural analysis step infers the minimum free energy (MFE) structure from a multiple sequence alignment (MSA) of all primary homologs within a lncRNA homolog family. Files are organized in the "Structures_From_MSA" folder and grouped by the ID of the query lncRNA. Shown here are the "aln.pdf" and "alirna.pdf" files. (**a**) Representative alignment with dot-bracket structural notation. Coloring is according to the Vienna package conservation coloring scheme. Dark red indicates consistent and conserved base pairing across the alignment. Pale colors indicate that base pairing is not possible in certain taxa within the alignment. Yellow, green, and blue shading indicates base pairing is retained although sequence has varied, suggesting covariation across the alignment. (**b**) Graphical output of the MFE structure generated from the alignment. Sequence with conserved structural annotation is denoted by a black circle around a specific nucleotide. Conservation of base pairing is denoted by the same coloring scheme in (**a**)

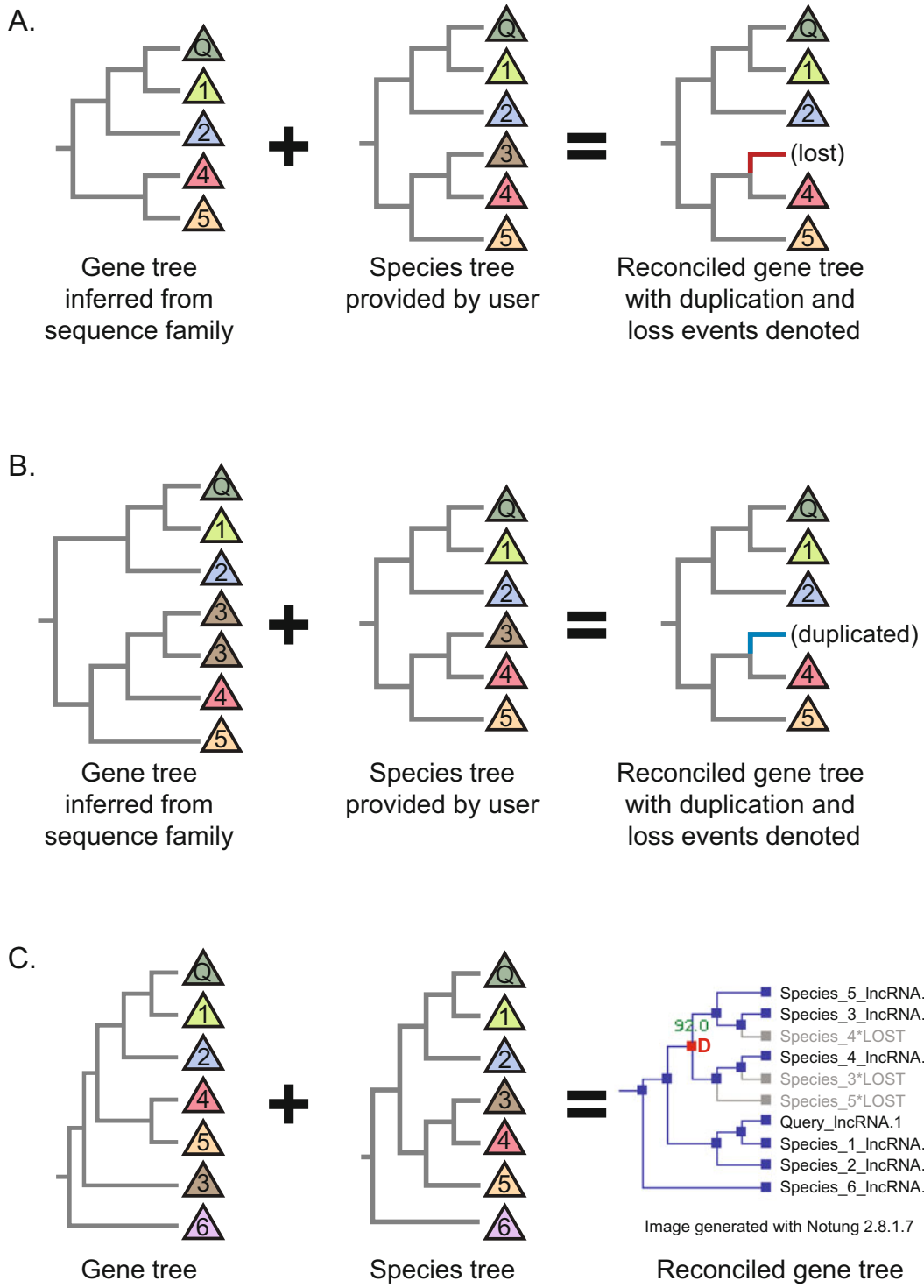


Fig. 3 Phylogenetic analysis. MSAs are used to infer sequence relationships using RAxML. These phylogenies are then reconciled to the known gene tree to determine duplication and loss events that may have occurred within each lncRNA family. (**a**, **b**) Examples of scenarios in which a duplication or loss event is inferred. (**c**) An example from a query Arabidopsis lncRNA where the homolog from Species 3 (brown triangle) fell sister to the Q-2,4-5 lineage instead of sister to Species 3. Notung used the species tree to reconcile three loss events and

6. Pull the latest Evolinc-II container from Docker using the Docker commands.

```
###
Docker pull evolinc/evolinc-II:latest
###
```

7. Then launch the Evolinc-II Docker container. An example is shown below for the sample data. A list of the different flags and what they refer to are shown in Table 3. With the exception of the "-t" flag, all others are required.
8. Running Evolinc-II on an HPC without sudo privileges requires Singularity. Singularity syntax is similar (but is highly dependent on the version of Singularity running on your HPC; please *see* https://singularity.lbl.gov/ for more details):
9. To identify deeply conserved lincRNAs without the phylogenetic analysis (fast), use the following Docker commands.

```
###
Docker run --rm -v $(pwd):/working-dir -w /working-dir evo-
linc/evolinc-ii:latest -b path/to/input/files/comparison_-
list.txt -l path/to/input/files/query_lincRNAs.fasta -q Atha
-i path/to/input/files -s path/to/input/files/species_list.
txt -o path/to/output -v 1e-20 -n 8
###
```

10. To add in the phylogenetic and structural analysis (slow), the following is used.

```
###
Docker run --rm -v $(pwd):/working-dir -w /working-dir evo-
linc/evolinc-ii:latest -b path/to/input/files/comparison_-
list.txt -l path/to/input/files/query_lincRNAs.fasta -q Atha
-i path/to/input/files -s path/to/input/files/species_list.
txt -o path/to/output -v 1e-20 -n 8 -t path/to/input/files/
species_tree.txt
###
```

Fig. 3 (continued) one duplication event (red "D"). A confidence value is assigned to the duplicated node (green text). Caution should be used in interpreting results, as phylogenetic inferences are highly influenced by the quality of the alignment. In this scenario, a whole genome duplication event is known to have occurred specifically within Species 3. A testable hypothesis is that only one of the two homologs was retained but displayed higher than expected levels of sequence variation

Table 3
Flags for Evolinc-II on the command line

Flag	Description
-b	Path to startup list
-l	Path to query lincRNA file
-q	Four letter format of the query species (e.g., Atha = *A. thaliana*)
-i	Input folder where all input files are stored
-s	Species list (*see* Table 1)
-v	Stringency value (will be updated in the near future, see GitHub repo for more info)
-n	Number of threads
-t	Path to species tree in newick format
-o	Path to output folder
-h	Usage

11. A discussion of the results produced is shown in Subheading 3.4 below. When running in the command line, Evolinc-II is quite verbose and will provide constant progress updates. These updates are useful for troubleshooting should an analysis fail. Please report issues to our GitHub issues page so that the team can address them (https://github.com/Evolinc/Evolinc-II/issues). More information on common mistakes can be found in **Notes 1–4**.

3.3 Launching Evolinc-II in CyVerse's Discovery Environment

1. Open the DE Apps window and search for Evolinc-II.
2. Click on the latest version of Evolinc-II (currently v2.0) to open the analysis panel.
3. In the Analysis Name:Evolinc-II panel: first change the analysis name (optional; but do not use spaces if changing). Next, enter any comments about the analysis (notes to self, etc.). Finally, in the "Select output folder" click "Browse" and navigate to a folder where Evolinc-II will deposit final results (optional). Default is to save results to the "analyses" folder.
4. Note that it is possible at this point to immediately launch a test of Evolinc-II. All mandatory fields are populated with example data to perform a comparative analysis with 100 sample lincRNAs from *Arabidopsis thaliana*. If running a test with the example data, provide the analysis a new name and click "Launch Analysis" in the bottom right.
5. To examine these example data and their format, navigate to: /iplant/home/shared/iplantcollaborative/example_data/Evolinc.sample.data/Evolinc-II.

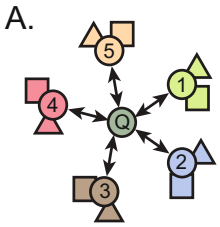
6. To start an analysis with new lncRNAs, click on the "Mandatory arguments for comparative analysis" tab. Place all necessary files in their appropriate field. Files/folders can be dragged and dropped into the appropriate field or navigated to using the "Browse" button.
7. To identify deeply conserved lncRNA sequence homologs, click "Launch Analysis" after filling out the mandatory arguments.
8. To perform the full analysis, click on the "Phylogenetic and structural analysis" tab. To initiate these analyses Evolinc-II requires a species tree in Newick format. Drag and drop this file into the appropriate field or browse to the appropriate location. An example species tree for the sample data can be found in the Evolinc.sample.data/Evolinc-II folder. Then click "Launch Analysis."
9. After launching the analysis, three notifications will appear at the bell icon in the top right of the DE browser stating that the analysis has launched, is running, and is completed. Job status can also be monitored by clicking on the "Analyses" icon.
10. When the analysis is completed, click on the job ID from the "Analyses" panel to navigate directly to the output folder. A discussion of the results follows below.

3.4 Interpreting Results

After performing the comparative analyses to identify sequence homologs across a user-defined group of species (*see* Fig. 1 for more details), Evolinc-II will return a variety of folders and files to help the user in examining the degree to which their lncRNAs are conserved and to aid in further analysis. What follows is an overview of the results and how they might be useful.

3.4.1 Results from a Standard Analysis

1. **Families of lincRNA sequence homologs:** The fundamental product of Evolinc-II's comparative analysis is a set of sequences, from each species for which search criteria are met, that share significant homology with a query lincRNA. A query-centric list of these families can be found in the file "final_summary_table.csv". These families of lincRNA sequence homologs are also saved as individual FASTA files (named by the query lincRNA) in the Orthologs/lincRNA_families directory. The visualization of the frequency of recovery, as well as species composition of each lincRNA family, can be found in the barplot and upset plots in the output directory. Finally, the genomic coordinates displaying the regions of sequence conservation can be found, in BED format for genome browser import (e.g., CoGe), in the file "All_orthologs_for_viewing.bed" in the output directory.
2. **/Orthologs/lincRNA_families:** This directory contains fasta files with the sequence homologs for each query lincRNA (a lincRNA sequence family). The FASTA headers for each



B.

Query_ID	Species 1	Species 2	Species 3	Species 4	Species 5	Inference of lincRNA origins/conservation
LincRNA.1	KL	KL	KL	KL	KL	Ancient origins, lincRNA in all tested species
LincRNA.2	KL	KL	NF	NF	NF	Recently derived or lineage-specific lincRNA
LincRNA.3	KGS	KGS	KGS	KGS	KGS	Recently derived lincRNA (possible pseudogene)
LincRNA.4	KL	KGA	KGA	KGA	KGA	Potential antisense lincRNA of ancient origins
LincRNA.5	H	H	H	H	H	Ancient origins, transcription status unknown outside query

Fig. 4 Inferring lncRNA origins using the final summary table. (**a**) Reciprocal searches are performed to identify the best candidate sequence homolog for each lncRNA. These sequence homologs are compared against genome annotation or known lncRNA files to determine if they correspond to a known gene. This information is then reported in the final summary table which is located in the Evolinc-II output folder. (**b**) A pictorial representation of the final summary table. Each lncRNA used as query will be listed in alphabetical order in the first column. Each subsequent column will correspond to a tested species in the order listed in the "Species_-list" file. Where applicable, sequence homologs from each species will have additional labels appended to their ID. KL = Known lncRNA and implies that the sequence homolog showed strong similarity to a lncRNA in the user-provided lncRNA file for that species. KGS = Known gene sense, indicating that the sequence homolog overlapped with a known gene on the sense strand in that species. Depending on the level of annotation, this annotation could suggest overlap with a protein-coding gene (low-level annotation in that species) or lncRNA, TE, etc. (high-level annotation). To assist the user in determining the molecule type of the corresponding gene, the gene ID (or lncRNA ID) is also appended. KGA = Known gene antisense, indicating that the sequence homolog overlapped with a known gene on the antisense strand. NF = Not found, indicating that no sequence homolog passed the reciprocal search criteria. H = Homolog, indicating that a sequence homolog was identified but that no additional information was obtained. Note: while KGS/KL annotations are strong indicators of transcription in a species, the lack of annotation assigned to a sequence homolog does not confirm lack of expression

sequence homolog contain the full information for that locus for the species from which it is derived. These files can be downloaded and used to develop more sensitive MSAs. In addition, consensus sequences derived from these sets of homologs can allow the user to search for homologs in more distant relatives.

3. **Final_summary_table.csv:** This table lists out the sequence homologs for each query lincRNA, and provides details about the sequence homolog in each subject species (e.g., whether the sequence homolog corresponds to a known mRNA or lncRNA, the ID of that gene, and the strandedness of the overlapping locus). Corresponding gene IDs are also attached to each FASTA header in the appropriate lincRNA sequence homolog FASTA file. These data can be used to infer lncRNA origins and depth of conservation (*see* Fig. 4).

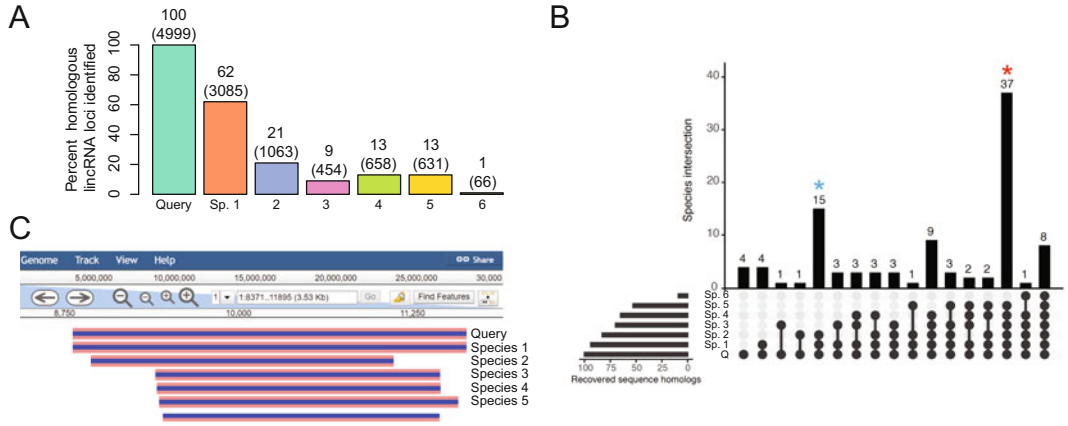


Fig. 5 Additional results files always generated by Evolinc-II. (**a**) Bar plot depicting percent recovered loci in each species. (**b**) Upset plot depicting compositional frequency of recovered lncRNA families. (**c**) Query-centric BED file for quickly depicting conserved regions of all lncRNA families at once

4. **LincRNA_barplot.pdf/png:** This plot, which is stored directly within the output folder, describes the number of sequence homologs that were identified in each species as a percent of the number of query sequences. An example is shown in Fig. 5. The PNG and PDF version are identical. The PNG is viewable within the Discovery Environment, whereas the PDF is editable for publication purposes. This plot is useful for inferring large scale trends in lncRNA evolution, such as the rapid loss of Arabidopsis lncRNA homologs seen in the Brassica lineage [10].
5. **LincRNA_upset_plot.pdf/png:** This plot utilizes the UpSetR package [20] to visualize the compositional frequency of the recovered families of sequence homologs. For instance, using the test data supplied with Evolinc-II, 15 lincRNA families contain sequences from three species: *A. thaliana* (the query), *A. lyrata*, and *C. rubella* (blue asterisk; Fig. 5). In contrast, 37 lincRNA families contain a representative from six species (red asterisk; Fig. 5), and eight lincRNA families contain a representative from each queried species.
6. **All_orthologs_for_viewing.bed:** This file contains the query-centric genomic coordinates corresponding to each sequence homolog for all query lincRNAs (i.e., many lincRNA families all at once).

3.4.2 Results from the Phylogenetic and Structural Analyses

When selected, Evolinc-II will also (1) use a phylogenetic approach to infer the evolutionary history of each lincRNA based on the degree to which it is conserved across the chosen taxa, and (2) use a multiple sequence alignment (MSA) approach to infer the most likely structure for each lincRNA (*see* **Note 4**). The files generated that are specific to this optional step are kept in separate folders in the output directory and are comprised of:

1. Reconciled_trees/*.png: These reconciled trees are meant to depict the most parsimonious tree, inferred by RAxML for each lincRNA sequence family with sufficient taxa (>3). Alignments are generated using MAFFT [21], and then passed on to RAxML to infer phylogenies, and then reconciled with NOTUNG [19]. Trees are reconciled with the user provided species tree to infer duplication and loss events (represented at nodes with a "D" or grayed out text at individual leaves). Trees are named based on the query lincRNA ID.
2. Orthologs/lincRNA_families/RAxML_families: This directory contains the RAxML inferred phylogenies for each lincRNA sequence family in Newick format, with branch support values. Files are named based on the query lincRNA ID.
3. Orthologs/lincRNAs/Final_results: This directory contains the MAFFT alignment files used by RAxML and can be useful for other downstream applications (e.g., covariation analysis).
4. Structures_from_MSA/*/results/alirna.pdf: These files are structures inferred by locARNA's analysis of multiple sequence alignments of each lincRNA sequence family. An example is shown in Fig. 2. Structures are only predicted for lincRNA sequence families with enough taxa to generate a MSA as MSA-derived structural predictions are in general more precise than alignments form single sequences.
5. Structures_from_MSA/*/results/aln.pdf: These files depict the alignment used by locARNA to make the structural prediction.

4 Notes

While Evolinc is still a work in progress (i.e., there are still bugs we are fixing), there are a few things that the user can do to make things run smoothly. In this section we will attempt to describe the most common causes of a failed analysis (or factors you should think about to get the most information out of your analyses). Please visit the Evolinc-II github repository (https://github.com/Evolinc/Evolinc-II) to ask questions or raise issues that are not addressed here.

1. Extended lincRNA IDs with spaces: Another common issue comes from having long lincRNA IDs with spaces, for example "LincRNA 1". Evolinc-II checks for "messy" gene IDs and attempts to clean them (command line users might notice the following statement: "Query lincRNA names are messy, attempting to clean them up. Replacing spaces and underscores with periods"). However, special symbols in the lincRNA headers might still be problematic. Consider simplifying lincRNA IDs to only include numbers, letters, and periods.
2. Concordance between genome sequence and genome annotations (FASTA and GTF/GFF): One common issue for anyone working with genome/transcriptome assembly is lack of concordance between genomes and their corresponding annotation files. Typically this is seen in differences in the way in which different files report the chromosome ID. For instance, one file might designate Chromosome 1 as "Chr1", whereas another file might simply denote it as "1". Generally this can be avoided by obtaining the genome and annotation files from the same source (ENSEMBL, Phytozome, etc.).
3. To filter or not to filter lincRNAs against rFAM: We have found that a number of lincRNAs (the amount varies by species and manner in which RNA-Seq was performed) contain high sequence similarity to (i.e., contain) known RNAs such as small nucleolar, microRNAs, and spliceosomal RNAs. The lincRNA molecules that contain these sequence motifs may indeed be lincRNAs; retaining them in your dataset prior to running Evolinc-II may skew your overall conservation rates, as these sequences tend to be more conserved. Evolinc-I naturally removes spliceosomal (U1-U6) RNAs and scans for and removes ribosomal RNAs. Depending on the researcher's interests, it may be appropriate to further filter their lincRNA catalog against rFAM [22].
4. Picking the right phylogenetic depth for comparative genomic analyses: LincRNA loci appear to be conserved to varying degrees in different eukaryotic lineages. Most of what we know about lincRNA conservation comes from vertebrates and, to a lesser degree, plants. What we have seen in plants is that comparative genomic analyses of plant lincRNAs are most informative if restricted to the family level. However, informative comparisons can be made at much deeper nodes when starting with mammalian lincRNAs. Thus, we suggest picking a few genomes to start your analysis: a couple of close relatives and then a few that are much more diverged. Run that, and then start filling the analysis in with more species.

Acknowledgments

We would like to thank Upendra K. Devisetty for continual advice on the Evolinc project. This work was supported by NSF-IOS grant # 1758532/2021753 to A.D.L.N.

References

1. Guttman M, Rinn JL (2012) Modular regulatory principles of large non-coding RNAs. Nature 482:339–346
2. Golicz AA, Bhalla PL, Singh MB (2018) lncRNAs in Plant and Animal Sexual Reproduction. Trends Plant Sci 23:195–205
3. Golicz AA, Singh MB, Bhalla PL (2018) The Long Intergenic Noncoding RNA (LincRNA) Landscape of the Soybean Genome. Plant Physiol 176:2133–2147
4. Foley SW, Gosai SJ, Wang D et al (2017) A global view of RNA-protein interactions identifies post-transcriptional regulators of root hair cell fate. Dev Cell 41:204–220.e5
5. Gosai SJ, Foley SW, Wang D et al (2015) Global analysis of the RNA-protein interaction and RNA secondary structure landscapes of the Arabidopsis nucleus. Mol Cell 57:376–388
6. Di C, Yuan J, Wu Y et al (2014) Characterization of stress-responsive lncRNAs in Arabidopsis thaliana by integrating expression, epigenetic and structural features. Plant J 80: 848–861
7. Charon C, Moreno AB, Bardou F et al (2010) Non-protein-coding RNAs and their interacting RNA-binding proteins in the plant cell nucleus. Mol Plant 3:729–739
8. Hezroni H, Perry RBT, Meir Z et al (2017) A subset of conserved mammalian long non-coding RNAs are fossils of ancestral protein-coding genes. Genome Biol 18:162
9. Darbellay F, Necsulea A (2020) Comparative transcriptomics analyses across species, organs, and developmental stages reveal functionally constrained lncRNAs. Mol Biol Evol 37: 240–259
10. Nelson ADL, Forsythe ES, Devisetty UK et al (2016) A genomic analysis of factors driving lincRNA diversification: lessons from plants. G3 6:2881–2891
11. Ruiz-Orera J, Mar Albà M (2019) Conserved regions in long non-coding RNAs contain abundant translation and protein–RNA interaction signatures. NAR Genom Bioinformatics 1:e2–e2
12. Nelson ADL, Devisetty UK, Palos K et al (2017) Evolinc: a tool for the identification and evolutionary comparison of long intergenic non-coding RNAs. Front Genet 8:52
13. Merchant N, Lyons E, Goff S et al (2016) The iPlant collaborative: cyberinfrastructure for enabling data to discovery for the life sciences. PLoS Biol 14:e1002342
14. Nelson ADL, Haug-Baltzell AK, Davey S et al (2018) EPIC-CoGe: managing and analyzing genomic data. Bioinformatics 34:2651–2653
15. Tang H, Lyons E (2012) Unleashing the genome of *Brassica rapa*. Front Plant Sci 3:172
16. Will S, Joshi T, Hofacker IL et al (2012) LocARNA-P: accurate boundary prediction and improved detection of structural RNAs. RNA 18:900–914
17. Raden M, Ali SM, Alkhnbashi OS et al (2018) Freiburg RNA tools: a central online resource for RNA-focused research and teaching. Nucleic Acids Res 46:W25–W29
18. Stamatakis A (2015) Using RAxML to infer phylogenies. Curr Protoc Bioinformatics 6: 14.1–14.14. https://doi.org/10.1002/0471250953.bi0614s51
19. Darby CA, Stolzer M, Ropp PJ et al (2017) Xenolog classification. Bioinformatics 33: 640–649
20. Conway JR, Lex A, Gehlenborg N (2017) UpSetR: an R package for the visualization of intersecting sets and their properties. Bioinformatics 33:2938–2940
21. Nakamura T, Yamada KD, Tomii K et al (2018) Parallelization of MAFFT for large-scale multiple sequence alignments. Bioinformatics 34: 2490–2492
22. Kalvari I, Argasinska J, Quinones-Olvera N et al (2018) Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. Nucleic Acids Res 46:D335–D342

Chapter 4

NGS-Indel Coder v2.0: A Streamlined Pipeline to Code Indel Characters in Phylogenomic Data

Julien Boutte, Mark Fishbein, and Shannon C. K. Straub

Abstract

Hypothesized evolutionary insertions and deletions in nucleic acid sequences (indels) contain significant phylogenetic information and can be integrated in phylogenomic analyses. However, assemblies of short reads obtained from next-generation sequencing (NGS) technologies can contain errors that result in falsely inferred indels that need to be detected and omitted to avoid inclusion in phylogenetic analysis. Here, we detail the commands that comprise a new version of the NGS-Indel Coder pipeline, which was developed to validate indels using assembly read depth.

Key words Target capture sequencing, SNP-detection, Character coding, Gaps, Insertion-deletion, Phylogeny, NGS read depth, IQ-TREE

1 Introduction

It is broadly accepted that indels (gaps in multiple sequence alignments inferred to represent insertion/deletion evolutionary events) contain significant phylogenetic information, and coding them may be helpful in resolving phylogenies [1–8]. However, in large phylogenomic data sets constructed from next generation sequencing (NGS) short read data, validation of correctly inferred indels has remained difficult because they may result from a problem during the heuristic multiple sequence alignment process or from errors during short read assembly processes. Several tools are available to validate inferred single nucleotide polymorphisms (SNPs) [8–12] and large intraspecific indels [13, 14]. More recently, Boutte et al. [7] developed a new pipeline, NGS-Indel Coder, which validates interspecific indel regions of any length, including indels greater than the read length. NGS-Indel Coder is applied to multilocus sequence alignments, such as those produced by targeted sequencing approaches, and uses sequencing read depth information to identify chimeric assemblies (resulting in falsely inferred indels),

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_4,

mainly caused by low or inadequate sequencing depth or misassembly, generally due to repetitive sequences [15]. For each alignment, NGS-Indel Coder identifies indels $\geq i$ bp ($i \in n$) (indels $<i$ bp are automatically not coded as characters for phylogenetic analysis), for all sites at which at least two samples contain sequences (i.e., insertions unique to a single sample are omitted). For each putative indel region (j) retained, a mean read depth (MRD) value is calculated for each sample. A MRD value corresponds to the mean read depth across the gap plus the flanking 10 bp before and after (MRD-2; for an insertion) or to the mean read depth flanking 10 bp before and after the gap (MRD-3; for a deletion). To reduce the effects of sequencing variation between samples, the program identifies for each indel region the sample pair with the most similar read depth (X_i, Υ_j), and compares MRD-2 (for individuals with sequences in the region) and MRD-3 (for individuals with gaps) values using Eq. 1.

$$\begin{aligned}\left(X_j = \max\left(\text{MRD-3}_k, \quad \forall \quad k \in j\right) \wedge \Upsilon_j = \text{abs}(\min\left(\left(\text{MRD-2}_k - \text{MRD-3}_k\right)\right),\right.\\ \forall \quad k \in j) \vee \left(X_j = \text{abs}(\min\left(\left(\text{MRD-3}_k - \text{MRD-2}_k\right)\right),\right.\\ \left.\forall \quad k \in j) \wedge \Upsilon_j = \max\left(\text{MRD-2}_k, \quad \forall \quad k \in j\right)\right) \qquad (1)\end{aligned}$$

Then, the pipeline retains the indel regions (j) only if $X_j \geq T_i$ and $\Upsilon_j \geq T_i$, where T_i ($i \in n$) corresponds to a predetermined MRD threshold (*see* [7] for details; Fig. 1). The predetermined MRD threshold T_i is dependent on the data set, and it is necessary to identify an appropriate T_i prior to coding indels for phylogenomic analyses [7]. The NGS-Indel Coder pipeline offers the option to exclude alignment partitions that consist of introns and exons shorter than a specified length. Another option allows the annotation of intron and exon boundaries to support the downstream application of mixed substitution models across partitions (*see* Subheading 3). NGS-Indel Coder generates output files suitable for input to IQ-TREE [16, 17]. This pipeline was successfully applied to improve phylogenomic resolution in milkweeds (*Asclepias*) [7]. In this chapter, step-by-step command lines of the new version of the NGS-Indel Coder pipeline [7] (available at https://github.com/juboutte/NGS-Indel_Coder_v2.0) are detailed.

2 Materials

2.1 System Requirements

NGS-Indel Coder requires Python 2.7.12, files created by 2MATRIX [18], developed in PERL, as well as a BLASTN output file (to annotate exon and intron boundaries). The 2matrix-master folder (https://github.com/nrsalinas/2matrix.git) must be copied and pasted into the NGS-Indel Coder folder (available at https://github.com/juboutte/NGS-Indel_Coder_v2.0).

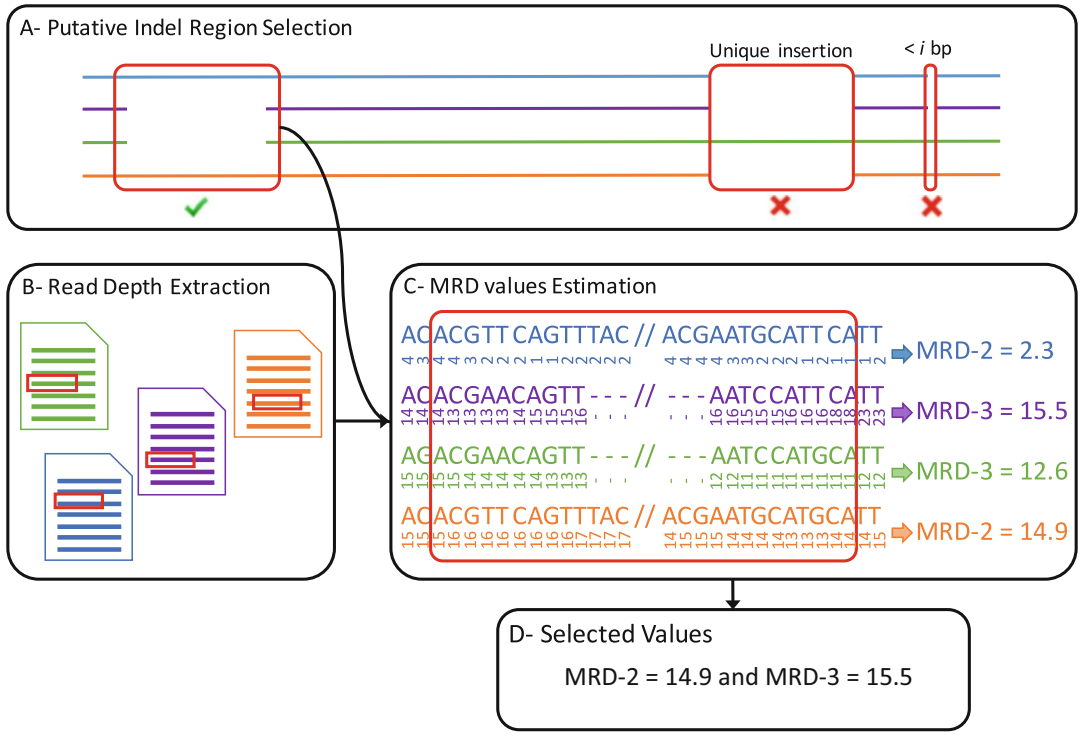


Fig. 1 Theoretical representation of indel selection and MRD estimation. (**a**) For each alignment, the pipeline identifies putative indel regions ($\geq i$ bp ($i \in n$) and at least two samples with sequences). (**b**) For each indel region selected, read depth information is extracted from text files and (**c**) the mean read depth is estimated for each sample. Then, the sample pair with the most similar read depth is identified using Eq. 1. Finally, the pipeline includes or excludes the indel region if the MRD-2 and MRD-3 retained values are higher than the selected threshold. In this example, the indel region was selected if the threshold was $\leq$14

2.2 NGS-Indel Coder Options

NGS-Indel Coder v.2.0 includes six custom python scripts (vs. 11 scripts in the original version [7]). For the new version, several scripts were gathered together for ease of use. All pipeline options are presented in Table 1. NGS-Indel Coder can include (Subheading 3.1) or exclude (Subheading 3.2) alignment partitions that are smaller than a specified length. Using BLASTN results, it is also possible to annotate intron and exon boundaries (Subheading 3.1.3 or 3.2.4) to support downstream application of mixed substitution models across partitions of an alignment. A decision tree is presented in Fig. 2.

2.3 Input Files Requirements

NGS-Indel Coder requires an aligned multisequence FASTA formatted file including reference sequences (e.g., HybPiper supercontig sequences [19]) and a read depth file for each sequence. Boutte et al. [7] used BWA v.0.7.12 [20], Samtools v.1.3.1 [9], and Picard v.1.131 (http://broadinstitute.github.io/picard/) tools to create read depth files. The command lines are presented below in

Table 1
List of NGS-Indel Coder python scripts and options

Script	Options
1-parsing_Samtools_depth-files.py	- -
2-Indel_validation_deletion.py	-f input FASTA file -r read depth files path -t threshold for Mean Read Depth (MRD). Indels with low MRD are not coded -d no/yes, delete temporary files -l minimum length for indels. Indels with low length are not coded
3-IQTREE_binary_matrices_creation.py	-y input PHYLIP (.phy) file -p input RAxML-style (.part) file -s no/yes, delete temporary files (for Subheading 3.2.3)
4-IQTREE_DNA_matrices_nexus_files_creation.py	-y input PHYLIP (.phy) file -p input RAxML-style (.part) file -f folder name for IQ-tree analysis -i input indel PHYLIP (_indel.phy) file -d no/yes, delete temporary files
5-identification_boundaries_nexus_files.py	-b input blastn file -f input FASTA file -n input NEXUS file -d no/yes, delete temporary files
6-identification_boundaries_small_partitions_deletion.py	-b input blastn file -f input FASTA file -t minimum size to include a partition. Partitions with low length are not used -o output FASTA file -d no/yes, delete temporary files

monospaced font. Example_species_A.fasta contains HybPiper supercontig sequences of species A. Example_species_A_1P.fastq and Example_species_A_2P.fastq correspond to cleaned (e.g., quality trimmed) paired read files. Example files (including raw paired read files and a HybPiper FASTA file) are available in NGS-Indel_Coder_v2.0/ Materials. It is possible to use single read files and reference files not generated by HybPiper (e.g. BWA, Samtools, or Picard manuals).

```
$ bwa index Example_species_A.fasta
```

```
$ bwa mem Example_species_A.fasta Example_species_A_1P.fastq
Example_species_A_2P.fastq > Example_species_A.sam
```

```
$ java -jar /opt/picard-tools/1.131/picard.jar NormalizeFasta
```

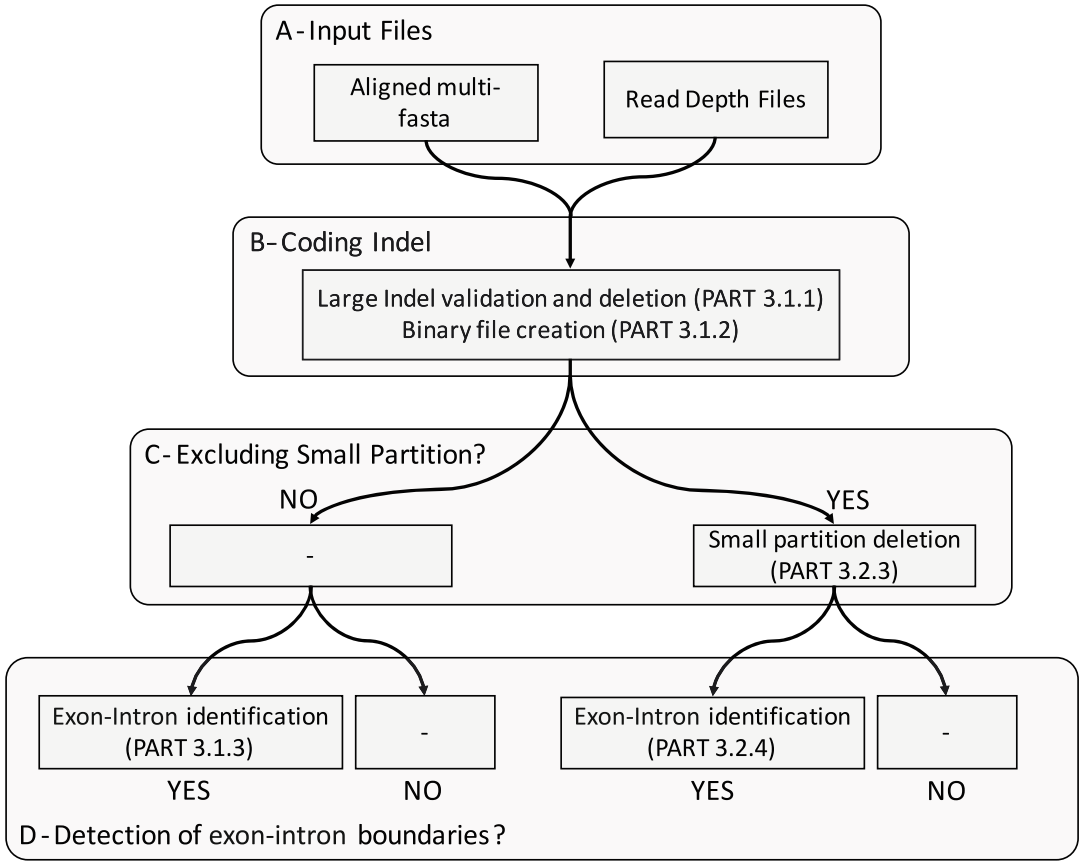


Fig. 2 NGS-Indel Coder decision tree

```
\
      I= Example_species_A.fasta \
      O= Example_species_A_cleaned.fasta

$ samtools faidx Example_species_A_cleaned.fasta

$ java -jar /opt/picard-tools/1.131/picard.jar CreateSequen-
ceDictionary R= Example_species_A_cleaned.fasta O= Example_-
species_A_cleaned.dict

$ java -jar /opt/picard-tools/1.131/picard.jar SortSam \
    VALIDATION_STRINGENCY=LENIENT \
    I= Example_species_A.sam \
    O= Example_species_A.bam \
    SORT_ORDER=coordinate

$ java -jar /opt/picard-tools/1.131/picard.jar AddOrReplaceR-
eadGroups \
      I= Example_species_A.bam \
```

```
      O= Example_species_A_cleaned.bam \
      RGID=4 \
      RGLB=lib1 \
      RGPL=illumina \
      RGPU=unit1 \
      RGSM=20

$ java -jar /opt/picard-tools/1.131/picard.jar BuildBamIndex \
      I= Example_species_A_cleaned.bam

$ samtools depth -a Example_species_A_cleaned.bam > Example_-
species_A_infos.txt
```

The Example_species_A_infos.txt (e.g., Asclepiasaffstandleyi1321-10025.txt available in NGS-Indel_Coder_v2.0/Example/depth_files/) file contains all read depth information for the sequences of species A. Users can parse this file using the 1-parsing_Samtools_depth-files.py program.

```
$ cd Desktop/NGS-Indel_Coder_v2.0/

$ python Scripts/1-parsing_Samtools_depth-files.py Example_-
species_A_infos.txt.
```

3 Methods

Command lines are in monospaced font. All files created using example command lines (including temporary files) are available in NGS-Indel_Coder_v2.0/Example/All_example_files/.

3.1 Running NGS-Indel Coder Including Small Alignment Partitions (Shorter Than Specified Threshold)

Input files:

An aligned FASTA file (Example/fasta_files/Example.fasta)

A folder path (containing all read depth files; Example/depth_files/)

Output file:

A temporary FASTA file (Example/Example_temp.fasta)

3.1.1 Large Indel Validation and Deletion

Sequence names used in the FASTA file must correspond to read depth files names (e.g. Asclepiasaffstandleyi1321-10025 and Asclepiasaffstandleyi1321-10025.txt). To create indel binary files, NGS-Indel Coder generates a temporary FASTA file. This FASTA file is only used to create indel binary files and not for phylogenomic analyses.

```
$ cd Desktop/NGS-Indel_Coder_v2.0/

$ python Scripts/2-Indel_validation_deletion.py -f Example/
fasta_files/Example.fasta -r Example/depth_files/ -t 20 -d
yes -l 2

$ mv Example_temp.fasta Example/
```

3.1.2 Binary File Creation

When the temporary FASTA file is generated (Subheading 3.1.1), 2MATRIX [18] is used to generate temporary binary matrices that include coded indel characters. Then, NGS-Indel Coder generates final NEXUS and PHYLIP files. Users may optionally make a directory called MyFolder_T20 prior to proceeding with the example commands below.

Input file:

The temporary FASTA file (Subheading 3.1.1; Example_temp.fasta).
Output files:
Three output files corresponding to IQ-TREE input files (.nex and .phy files extensions; Example_indel.phy, Example2_dna.phy, and Example2.nex).

```
$ perl 2matrix-master/2matrix.pl -i Example/Example_temp.fasta
-n Example -o p

$ python Scripts/3-IQTREE_binary_matrices_creation.py -y Ex-
ample.phy -p Example.part

$ perl 2matrix-master/2matrix.pl -i Example/fasta_files/Exam-
ple.fasta -n Example2 -o p

$ python Scripts/4-IQTREE_DNA_matrices_nexus_files_creation.
py -y Example2.phy -p Example2.part -i Example_indel.phy -f
MyFolder_T20/ -d yes

$ mv Example_indel.phy Example2_dna.phy Example2.nex Example/
```

MyFolder_T20/ corresponds to the folder that will contain Example2_dna.phy and Example_indel.phy, which the user will use to run IQ-TREE. An IQ-TREE command line example is available (*see* **Note 1**).

3.1.3 Exon/Intron Identification

NGS-Indel Coder offers the option to annotate intron and exon boundaries using a custom approach (and BLASTN results) to provide input files for running a partitioned IQ-TREE analysis. This part of NGS-Indel Coder creates a new NEXUS file (Example_partition.nex) that will replace the previous NEXUS file (Example2.nex; Subheading 3.1.2).

Input files

The NEXUS file created in Subheading 3.1.2.
An aligned FASTA file (Example/fasta_files/Example.fasta)
Output BLASTN result (res_blastn.txt; *see* **Note 2**)
Output file
A NEXUS (.nex) output file (Example_partition.nex).

```
$ python Scripts/5-identification_boundaries_nexus_files.py
-b Example/res_blastn.txt -f Example/fasta_files/Example.fas-
ta -n Example/Example2.nex -d yes

$ mv Example_partition.nex Example/
```

Example_partition.nex replaces Example2.nex. It is one of the three final files used as input by IQ-TREE (Example_partition.nex, Example2_dna.phy and Example_indel.phy). An IQ-TREE command line example is available (*see* **Note 3**).

3.2 Running NGS-Indel Coder Omitting Small Alignment Partitions (Shorter Than a Specified Threshold)

3.2.1 Large Indel Validation and Deletion

Run Subheading 3.1.1

3.2.2 Binary File Creation

Run Subheading 3.1.2

3.2.3 Small Alignment Partition Deletion

NGS-Indel Coder supports the deletion of alignment partitions $\leq x$ bp (100 bp by default).

Input files:

An aligned FASTA file (Example/fasta_files/Example.fasta)
Output BLASTN result (res_blastn.txt; *see* **Note 2**)
Output files:
An aligned FASTA file cleaned (small partitions deleted; Example_-delete_partitions.fasta). Three output files corresponding to

IQ-TREE input files (.nex and .phy files extensions; Example_delete_partitions_indel.phy, Example_delete_partitions_dna.phy, Example_delete_partitions.nex).

```
$ python Scripts/6-identification_boundaries_small_partitions_deletion.py -b Example/res_blastn.txt -f Example/fasta_files/Example.fasta -t 100 -o Example_delete_partitions.fasta -d yes

$ mv Example_delete_partitions.fasta Example/

$ perl 2matrix-master/2matrix.pl -i Example/Example_temp.fasta -n Example_delete_partitions -o p

$ python Scripts/3-IQTREE_binary_matrices_creation.py -y Example_delete_partitions.phy -p Example_delete_partitions.part -s yes

$ perl 2matrix-master/2matrix.pl -i Example/Example_delete_partitions.fasta -n Example_delete_partitions -o p

$ python Scripts/4-IQTREE_DNA_matrices_nexus_files_creation.py -y Example_delete_partitions.phy -p Example_delete_partitions.part -i Example_delete_partitions_indel.phy -f MyFolder_T20/ -d no

$ mv Example_delete_partitions_dna.phy Example_delete_partitions_indel.phy Example_delete_partitions.nex Example/
```

MyFolder_T20/ corresponds to the folder that will contain Example_delete_partitions_dna.phy and Example_delete_partitions_indel.phy which the user will use to run IQ-TREE. An IQ-TREE command line example is available (*see* **Note 4**).

3.2.4 Exon–Intron Identification

NGS-Indel Coder offers the option to annotate intron and exon boundaries using a custom approach (and BLASTN results) to provide input files for running a partitioned IQ-TREE analysis. This part of NGS-Indel Coder creates a new NEXUS file (Example_delete_partitions_partition.nex) that will replace the previous NEXUS file (Example_delete_partitions.nex; Subheading 3.2.3).

Input files

The NEXUS file created in Subheading 3.2.3.
An aligned FASTA file (Example/fasta_files/Example_delete_partitions.fasta)
Output BLASTN result (res_blastn.txt; *see* **Note 2**)
Output file

A NEXUS (.nex) output file (Example_delete_partitions_partition.nex).

```
$ python Scripts/5-identification_boundaries_nexus_files.py -b Example/res_blastn.txt -f Example/Example_delete_partitions.fasta -n Example/Example_delete_partitions.nex -d yes

$ mv Example_delete_partitions_partition.nex Example/
```

Example_delete_partitions_partition.nex replaces Example_delete_partitions.nex. It is one of the three final files used by the IQ-TREE software (Example_delete_partitions_partition.nex, Example_delete_partitions_dna.phy, and Example_delete_partitions_indel.phy). An IQ-TREE command line example is available (*see* **Note 5**).

4 Notes

1. IQ-TREE command line example.

```
$ iqtree -nt 1 -bb 10000 -spp Example2.nex -m MFP+MERGE -AICc
```

   Example2_dna.phy and Example_indel.phy are in the folder: MyFolder_T20

```
Iqtree_analyses/
  Example2.nex
  MyFolder_T20/
    Example2_dna.phy
    Example_indel.phy
```

2. BLAST command line example (database: the transcript sequences used to create nuclear gene target capture probes, query: an aligned FASTA file).

```
$ makeblastdb -in Example/Example_transcripts.fasta -out transcript_db -dbtype nucl
$ blastn -query Example/fasta_files/Example.fasta -db transcript_db -outfmt 7 -out res_blastn.txt
```

   Example_transcripts.fasta is not available in the example folder but res_blastn.txt is available.

3. IQ-TREE command line example.

```
$ iqtree -nt 1 -bb 10000 -spp Example_partition.nex -m MFP+MERGE -AICc
```

   Example2_dna.phy and Example_indel.phy are in the folder: MyFolder_T20

```
Iqtree_analyses/
```

Example_partition.nex
MyFolder_T20/
Example2_dna.phy
Example_indel.phy

4. IQ-TREE command line example.

```
$ iqtree -nt 1 -bb 10000 -spp Example_delete_partitions.nex
-m MFP+MERGE -AICc
```

Example_delete_partitions_dna.phy and Example_delete_partitions_indel.phy are in the folder: MyFolder_T20
Iqtree_analyses/
Example_delete_partitions.nex
MyFolder_T20/
Example_delete_partitions_dna.phy
Example_delete_partitions_indel.phy

5. IQ-TREE command line example.

```
$ iqtree -nt 1 -bb 10000 -spp Example_delete_partitions_par-
tition.nex -m MFP+MERGE -AICc
```

Example_delete_partitions_dna.phy and Example_delete_partitions_indel.phy are in the folder: MyFolder_T20
Iqtree_analyses/
Example_delete_partitions_partition.nex
MyFolder_T20/
Example_delete_partitions_dna.phy
Example_delete_partitions_indel.phy

Acknowledgments

This work was supported by NSF DEB awards 1457510/1457473 to MF and SCKS.

References

1. Giribet G, Wheeler WC (1999) On gaps. Mol Phylogenet Evol 13:132–143. https://doi.org/10.1006/mpev.1999.0643
2. Simmons MP, Ochoterena H (2000) Gaps as characters in sequence-based phylogenetic analyses. Syst Biol 49:13
3. Redelings BD, Suchard MA (2007) Incorporating indel information into phylogeny estimation for rapidly emerging pathogens. BMC Evol Biol 7:40. https://doi.org/10.1186/1471-2148-7-40
4. Belinky F, Cohen O, Huchon D (2010) Large-scale parsimony analysis of metazoan indels in protein-coding genes. Mol Biol Evol 27:441–451. https://doi.org/10.1093/molbev/msp263
5. Paśko Ł, Ericson PGP, Elzanowski A (2011) Phylogenetic utility and evolution of indels: a study in neognathous birds. Mol Phylogenet

Evol 61:760–771. https://doi.org/10.1016/j.ympev.2011.07.021

6. Warnow T (2012) Standard maximum likelihood analyses of alignments with gaps can be statistically inconsistent. PLoS Curr 4: RRN1308. https://doi.org/10.1371/currents.RRN1308
7. Boutte J, Fishbein M, Liston A, Straub SCK (2019) NGS-Indel Codel: a pipeline to code indel characters in phylogenomic data with an example of its application in milkweeds (Asclepias). Mol Phylogenet Evol 139:106534. https://doi.org/10.1016/j.ympev.2019.106534
8. Houde P, Braun EL, Narula N, Minjares U, Mirarab S (2019) Phylogenetic signal of indels and the neoavian radiation. Diversity 11(7): 108. https://doi.org/10.3390/d11070108
9. Li H, Handsaker B, Wysoker A et al (2009) The sequence alignment/map format and SAMtools. Bioinformatics 25:2078–2079. https://doi.org/10.1093/bioinformatics/btp352
10. Li R, Li Y, Fang X et al (2009) SNP detection for massively parallel whole-genome resequencing. Genome Res 19:1124–1132. https://doi.org/10.1101/gr.088013.108
11. Xu F, Wang W, Wang P et al (2012) A fast and accurate SNP detection algorithm for next-generation sequencing data. Nat Commun 3: 1258. https://doi.org/10.1038/ncomms2256
12. McKenna A, Hanna M, Banks E et al (2010) The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res 20: 1297–1303. https://doi.org/10.1101/gr.107524.110
13. Boutte J, Aliaga B, Lima O et al (2016) Haplotype detection from next-generation sequencing in high-ploidy-level species: 45S rDNA gene copies in the hexaploid spartina maritima. G3 6:29–40. https://doi.org/10.1534/g3.115.023242
14. Boutte J, Ferreira de Carvalho J, Rousseau-Gueutin M et al (2016) Reference transcriptomes and detection of duplicated copies in hexaploid and allododecaploid spartina species (Poaceae). Genome Biol Evol 8:3030–3044. https://doi.org/10.1093/gbe/evw209
15. Muggli MD, Puglisi SJ, Ronen R, Boucher C (2015) Misassembly detection using paired-end sequence reads and optical mapping data. Bioinformatics 31:i80–i88. https://doi.org/10.1093/bioinformatics/btv262
16. Nguyen LT, Schmidt HA, von Haeseler A, Minh BQ (2015) IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Mol Biol Evol 32: 268–274. https://doi.org/10.1093/molbev/msu300
17. Chernomor O, von Haeseler A, Minh BQ (2016) Terrace aware data structure for phylogenomic inference from supermatrices. Syst Biol 65:997–1008. https://doi.org/10.1093/sysbio/syw037
18. Salinas NR, Little DP (2014) 2matrix: a utility for indel coding and phylogenetic matrix concatenation. Appl Plant Sci 2:1300083. https://doi.org/10.3732/apps.1300083
19. Johnson MG, Gardner EM, Liu Y et al (2016) HybPiper: extracting coding sequence and introns for phylogenetics from high-throughput sequencing reads using target enrichment. Appl Plant Sci 4:1600016. https://doi.org/10.3732/apps.1600016
20. Li H, Durbin R (2009) Fast short read alignment with Burrows-Wheeler transform. Bioinformatics 25(14):1754–1760. https://doi.org/10.1093/bioinformatics/btp324

Chapter 5

An SGSGeneloss-Based Method for Constructing a Gene Presence–Absence Table Using Mosdepth

Cassandria G. Tay Fernandez, Jacob I. Marsh, Benjamin J. Nestor, Mitchell Gill, Agnieszka A. Golicz, Philipp E. Bayer, and David Edwards

Abstract

Presence–absence variants (PAV) are genomic regions present in some individuals of a species, but not others. PAVs have been shown to contribute to genomic diversity, especially in bacteria and plants. These structural variations have been linked to traits and can be used to track a species' evolutionary history. PAVs are usually called by aligning short read sequence data from one or more individuals to a reference genome or pangenome assembly, and then comparing coverage. Regions where reads do not align define absence in that individual, and the regions are classified as PAVs. The method below details how to align sequence reads to a reference and how to use the sequencing-coverage calculator Mosdepth to identify PAVs and construct a PAV table for use in downstream comparative genome analysis.

Key words Single-nucleotide polymorphisms, Gene loss, Presence–absence variants, SGSGeneLoss

1 Introduction

Presence–absence variants (PAV) are an important type of genomic variation where regions of the genome are found in some but not all individuals of a species. PAVs contribute significantly to genetic diversity. They have been linked to local adaptations of wild populations [1, 2] and important agronomic traits of crops and crop relatives [3–6]. Differences in gene content within a species, copy number variation, and chromosomal rearrangement have also been attributed to PAVs [7]. Identifying and characterizing PAVs is useful to track the loss or gain of genes that are associated with phenotypic traits, such as abiotic and biotic stress responses [8, 9], identifying genetic divergence, studying evolution histories and determining gene content.

PAVs have been studied in a variety of organisms including bacteria [10], animals [11, 12], including insects [13] and humans [14]. PAVs have been extensively studied in plants [3], particularly

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_5,

crops such as *Brassicas* [2, 5, 15] soybean (*Glycine max*) [16], tomato (*Solanum lycopersicum*) [17], pigeon pea [18], sesame [19], clover (*Trifolium subterraneum*) [20], wheat [21], banana [22], and rice (*Oryza sativa*) [23] and have been used as molecular markers to examine variation in *Arabidopsis thaliana* accessions [24]. Their importance in plants has been demonstrated by being used to locate the geographical origin of crop species and chart their domestication by comparison with wild relatives, revealing important genetic events in the evolutionary history of the species [25]. Understanding gene PAVs can support applications for genetic improvement in plant breeding, identify valuable sources of genetic diversity in wild relatives and aid in reintroducing lost diversity into modern crops [26–28].

Single-reference genomes are limited in that they only represent the gene content of the individual sequenced for that reference. In contrast, a pangenome is a compilation of genes for all members of a given species and can be used to compare the genomes of individuals, capturing genetic diversity in a way single references cannot [29]. A pangenome is comprised of the core genome, which encompasses the genes and genomic regions present in all individuals of a species, and the variable genome, which encompasses all variants found in a given species including PAVs that are not found in all individuals [30, 31]. Using a pangenome reference rather than the reference from an individual of a species improves short-read mapping accuracy, resulting in higher quality variant calling and increased efficiency of SNP calling [4, 32, 33].

The method for PAV calling described in this chapter is based on the freely available Java program SGSGeneLoss which calculates position-wise coverage across the whole genome and within exons of each individual gene [34]. This method is conservative in calling gene absence to avoid calling allelic differences as different genes. Using standard bioinformatics programs, this tutorial is designed to rapidly generate a PAV table using a reference genome or pangenome for a given species while still being thorough and accurate.

2 Materials

2.1 Input Reads

To identify genic PAVs, a reference genome or a pangenome in FASTA format needs to be available (*see* **Note 1**). Additionally, individual sequencing files need to be mapped to this reference genome and compressed into BAM file format (*see* **Note 2**).

This guide will use individual_1_run_1.bam and individual_1_run_2.bam to represent two different sequencing runs for a single individual. These have already been aligned to the hypothetical reference, reference_pan.fasta. A GFF file with the reference genome annotation also needs to be available with the gene, mRNA, and exon fields, which will be reference_pan_annotation.gff in this tutorial.

2.2 Hardware

This protocol is written for bash with the Linux distribution, Ubuntu 18.4 or higher. Large datasets are appropriate for processing through high performance computing (HPC) to expedite the analysis.

2.3 Software

This workflow uses the publicly available tools samtools [35], mosdepth [36], bedtools [37], and BEDOPS [38].

Samtools is needed to merge different sequencing runs from the same individuals and sort the alignments. Mosdepth calculates the depth of coverage per-base and bedtools is used to find the intersection between exons and coverage. These packages are downloadable through bioconda [39] which is a bioinformatics-specific software source for the conda package manager. Bioconda requires the installation of miniconda: https://docs.conda.io/en/latest/miniconda.html.

3 Methods

This method is summarised in Fig. 1. The input sequencing data must be merged which can be done using samtools merge. This produces the output indi_1_merged.bam which is a single ordered output file with the inputs from both files. These alignments are then sorted using samtools sort which organises the alignments by their leftmost coordinates to allow samtools index to index correctly. The sorted file is output to the file indicated by the -o flag.

```
$ samtools merge indi_1_merged.bam individual_1_run_1.bam
individual_1_run_2.bam
$ samtools sort -o indi_1_merged_sorted.bam indi_1_-
merged.bam
```

Samtools index allows for fast and random access of a file, allowing programs like samtools view to rapidly process data. The -c flag forces the file into CSI formatted file which can index long contigs. Most bam-based workflows require files to be indexed and indexing should be run after every step. The following command produces indi_1_merged_sorted.bam.csi.

```
$ samtools index -c indi_1_merged_sorted.bam
```

The exon features are then extracted from the annotation GFF3 using awk. The -F flag sets the input field separator to tab-delimited (written as "\t"). $3 indicates the third column of the GFF3 file and == refers the matching value we are looking for

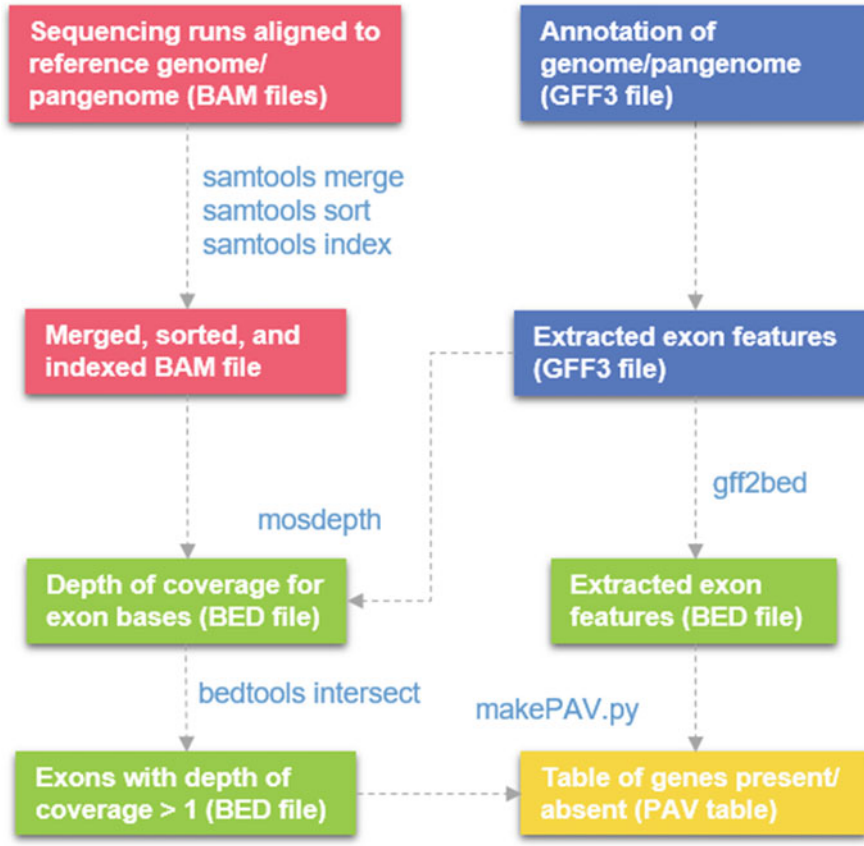


Fig. 1 A flow diagram of the method to create a table of present–absent genes (PAV table) for a given species. Programs used are written on the arrows outside the boxes

("exon"). {print $0} means that if an exon is found in the third column, the entire line is printed (*see* **Note 3**).

```
$ awk -F "\t" '$3 == exon {print $0}'  reference_pan_an-
notation.gff  > reference_pan_annotation_exons_only.gff
```

The resulting GFF file reference_pan_annotation_exons_only.gff is used to calculate the per-exon base coverage using mosdepth. Mosdepth calculates the depth of coverage for the given sequences at every base. The -b flag restricts the search to the exons provided by reference_pan_annotation_exons_only.gff (*see* **Note 4**). This narrows the scope of the analysis and makes searching through the genome a lot faster. indi_1_merged_sorted is the prefix for the name of the output files, eg: indi_1_merged_sorted.mosdepth.per-base.bed.gz. Mosdepth produces text files with the coverages distances: indi_1_merged_sorted.mosdepth.global.dist.txt,

indi_1_merged_sorted.mosdepth.summary.txt, indi_1_merged_-sorted.per-base.bed.gz.

```
$ mosdepth -b reference_pan_annotation_exons_only.gff
indi_1_merged_sorted indi_1_merged_sorted.bam
```

Bedtools intersect can be used on this output to find the intersection of the exons and the coverage. The -wao filter prints both the inputs as well as the number of base pairs that overlap between the two. Features that do not overlap for the first query are reported with a NULL B feature.

The grep command retrieves lines only with the word "exon" in it. The subsequent awk command states that if the value in the fourth column ($4) is equal or greater than two, print into indi_1_merged_sorted.per-base.bed_vs_exons.bed. This is done in order to find base pair positions with a depth greater than 1, which would indicate the presence or absence of a gene.

```
$ bedtools intersect -a indi_1_merged_sorted.mosdepth.
per-base.bed.gz -b reference_pan_annotation_exons_only.
gff -wao | grep "exon" | awk '{if ($4 >=2) print' >
indi_1_merged_sorted.per-base.bed_vs_exons.bed
```

The reference annotation file needs to be converted into BED format. This can be done using the script gff2bed which is part of BEDOPS.

```
$ gff2bed < reference_pan_annotation_exons_only.gff >
reference_pan_exons.bed
```

This custom script, makePAV.py, is a Python script that can be downloaded from https://github.com/AppliedBioinformatics/PAV_table_scripts/blob/main/makePAV.py. This script determines which genes are present or absent based on the per-base coverage reported by mosdepth. The input required for the script is the annotation file in BED format and the filtered mosdepth file (*see* **Note 4**).

```
$ python makePAV.py reference_pan_exons.bed
indi_1_merged_sorted_mosdepth.per-base.bed_vs_exons.bed
>   indi_1_merged_sorted_mosdepth.per-base.bed_vs_exons.
tsv
```

Each of the produced files will have two lines each: a header and the PAV for an individual. To format the table and include proper headers, take the first line from the mosdepth.per-base.bed_vs_exons.tsv file and create a new file with just the header. Then, add the last line from the .tsv file to the new PAV_table.csv. The output should be a properly functioning PAV table. The above can be run in parallel for many individuals, resulting in one TSV file with each header line and a line with the gene PAV. The following is a merge command.

```
$ head -n 1 indi_1_merged_sorted_mosdepth.per-base.bed_-
vs_exons.tsv > header
for i in *tsv ; do tail -n 1 $i ; done >> Header
mv Header PAV_table.csv
```

The resulting file, PAV_table.csv, is now ready for further use and can be used to answer relevant biological questions that the study is attempting to address.

4 Notes

1. There are many reference genomes freely accessible online for a variety of species. The human reference genome can be found on NCBI: https://www.ncbi.nlm.nih.gov/genome/guide/human/ and many plant reference genomes can be found on Ensembl plants: http://plants.ensembl.org/index.html
2. Mapping individual genomes in FASTA format to a reference genome or pangenome in FASTA can be done using bowtie2 end-to-end [14] to produce a SAM alignment file that can be compressed into BAM format using samtools. A bowtie2 index must first be created using bowtie2-build, as below. Bowtie2 can then be used to align the R1 and R2 of individual 1 to the index (flag -x) and output a SAM file named with the -S flag. Samtools view is used to compress the resultant SAM file into a BAM file.

```
$ bowtie2-build <reference-genome-or-pangenome>
<bt2-idx-name>

$ bowtie2 -x <bt2-idx-name> -1 R1_individual_1.
fastq.gz -2 R2_individual_1.fastq.gz -S indivi-
dual_1_run_1.sam
$ samtools view -S -b individual_1_run_1.sam   >
individual_1_run_1.bam
```

3. If an annotated GFF3 file has no exon information, a CDS annotation file can be used instead. The following command searches the no_exon_reference_pan_annotation.gff file for CDS annotations in the third column, before printing matching lines into a separate file. This annotation file with only CDS references is labeled reference_pan_annotation_cds_only.gff

```
$ awk -F "\t" '$3 == CDS {print $0}'  no_exon_refer-
ence_pan_annotation.gff     >  reference_pan_annota-
tion_cds_only.gff
```

4. PAV analysis can take a long time to run, especially on large datasets such as whole plant genomes/pangenomes. In this tutorial, we have restricted the search to the exome, although it may also be appropriate to search in a specific chromosome or other regions of interest that can be provided using a GFF3 or BED file.

References

1. Kreplak J et al (2019) A reference genome for pea provides insight into legume genome evolution. Nat Genet 51(9):1411–1422
2. Dolatabadian A et al (2020) Characterization of disease resistance genes in the Brassica napus pangenome reveals significant structural variation. Plant Biotechnol J 18(4):969–982
3. Saxena RK, Edwards D, Varshney RK (2014) Structural variations in plant genomes. Brief Funct Genom 13(4):296–307
4. Bayer PE et al (2020) Plant pan-genomes are the new reference. Nat Plants 6(8):914–920
5. Bayer PE et al (2019) Variation in abundance of predicted resistance genes in the Brassica oleracea pangenome. Plant Biotechnol J 17(4):789–800
6. Bayer PE et al (2021) The application of pangenomics and machine learning in genomic selection. Plant Genome 14:e20112
7. Brunner S et al (2005) Evolution of DNA sequence nonhomologies among maize inbreds. Plant Cell 17(2):343–360
8. Sieber A-N et al (2016) Copy number variation of CBF-A14 at the Fr-A2 locus determines frost tolerance in winter durum wheat. Theor Appl Genet 129(6):1087–1097
9. Saxena KB (2008) Genetic improvement of pigeon pea — a review. Trop Plant Biol 1(2): 159–178
10. Arrach N et al (2008) Salmonella serovar identification using PCR-based detection of gene presence and absence. J Clin Microbiol 46(8): 2581
11. Gerdol M et al (2019) Massive gene presence/absence variation in the mussel genome as an adaptive strategy: first evidence of a pan-genome in Metazoa. bioRxiv:781377
12. Golicz AA et al (2020) Pangenomics comes of age: from bacteria to plant and animal applications. Trends Genet 36(2):132–145
13. Kern AD, Begun DJ (2008) Recurrent deletion and gene presence/absence polymorphism: telomere dynamics dominate evolution at the tip of 3L in Drosophila melanogaster and D. simulans. Genetics 179(2):1021–1027
14. Iafrate AJ et al (2004) Detection of large-scale variation in the human genome. Nat Genet 36(9):949–951
15. Hurgobin B et al (2018) Homoeologous exchange is a major cause of gene presence/absence variation in the amphidiploid Brassica napus. Plant Biotechnol J 16(7):1265–1274
16. Li YH et al (2014) De novo assembly of soybean wild relatives for pan-genome analysis of diversity and agronomic traits. Nat Biotechnol 32(10):1045–1052
17. Alonge M et al (2020) Major impacts of widespread structural variation on gene expression and crop improvement in tomato. Cell 182(1): 145–161.e23

18. Zhao J et al (2020) Trait associations in the pangenome of pigeon pea (Cajanus cajan). Plant Biotechnol J 18(9):1946–1954
19. Yu J et al (2019) Insight into the evolution and functional characteristics of the pan-genome assembly from sesame landraces and modern cultivars. Plant Biotechnol J 17(5):881–892
20. Yuan Y et al (2018) Large-scale structural variation detection in subterranean clover subtypes using optical mapping. Front Plant Sci 9:971
21. Montenegro JD et al (2017) The pangenome of hexaploid bread wheat. Plant J 90(5): 1007–1013
22. Rijzaani H et al (2021) The pangenome of banana highlights differences between genera and genome. Plant Genome:e20100
23. Hu Z et al (2018) Novel sequences, structural variations and gene presence variations of Asian cultivated rice. Sci Data 5(1):180079
24. Salathia N et al (2007) TECHNICAL ADVANCE: Indel arrays: an affordable alternative for genotyping. Plant J 51(4):727–737
25. Kim MY et al (2010) Whole-genome sequencing and intensive analysis of the undomesticated soybean (Glycine soja Sieb. and Zucc.) genome. Proc Natl Acad Sci U S A 107(51): 22032–22037
26. Zhang Y et al (2018) Applications and potential of genome editing in crop improvement. Genome Biol 19(1):210
27. Gao L et al (2019) The tomato pan-genome uncovers new genes and a rare allele regulating fruit flavor. Nat Genet 51(6):1044–1051
28. Khan AW et al (2020) Super-pangenome by integrating the wild side of a species for accelerated crop improvement. Trends Plant Sci 25(2):148–158
29. Danilevicz MF et al (2020) Plant pangenomics: approaches, applications and advancements. Curr Opin Plant Biol 54:18–25
30. Tettelin H et al (2005) Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial "pan-genome". Proc Natl Acad Sci U S A 102(39):13950
31. Ruperao P et al (2021) Sorghum pan-genome explores the functional utility to accelerate the genetic gain. bioRxiv:2021.02.02.429137
32. Golicz AA et al (2016) The pangenome of an agronomically important crop plant Brassica oleracea. Nat Commun 7(1):13390
33. Hurgobin B, Edwards D (2017) SNP Discovery using a pangenome: has the single reference approach become obsolete? Biology 6(1):21
34. Golicz AA et al (2015) Gene loss in the fungal canola pathogen Leptosphaeria maculans. Funct Integr Genom 15(2):189–196
35. Li H et al (2009) The sequence alignment/map (SAM) format and SAMtools. Bioinformatics 25(16):2078–2079
36. Pedersen B, Quinlan A (2018) Mosdepth: quick coverage calculation for genomes and exomes. Bioinformatics 34(5):867–868
37. Quinlan A, Hall I (2010) BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics 26:841–842
38. Neph S et al (2012) BEDOPS: high-performance genomic feature operations. Bioinformatics 28(14):1919–1920
39. Grüning B et al (2018) Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat Methods 15(7):475–476

Chapter 6

POInT: A Tool for Modeling Ancient Polyploidies Using Multiple Polyploid Genomes

Yue Hao and Gavin C. Conant

Abstract

Ancient polyploidy events are widely distributed across the evolutionary history of eukaryotes. Here, we describe a likelihood-based tool, POInT (the *P*olyploidy *O*rthology *In*ference *T*ool), for modeling ancient whole genome duplications and triplications, assigning homoeologous genes to subgenomes and inferring gene losses across different parental subgenomes after polyploidy.

Key words Ancient polyploidy, Subgenomes, Gene loss, Evolution, Orthology

1 Introduction

About 14 years ago, we described a model-based approach to understanding the resolution of polyploidy events through duplicate gene loss [1, 2]. This tool, POInT (the *P*olyploidy *O*rthology *In*ference *T*ool), uses synteny data to statistically associate adjacent genes in each genome, allowing for the combination of loss information from multiple genes and genomes to "phase" regions of each polyploid genome relative to the others, identifying orthologous and paralogous chromosome regions. This phasing is performed probabilistically using a hidden Markov model (HMM; *see* [3]) that resembles the Lander-Green approach for constructing linkage maps from ordered genetic markers and a pedigree [4]. If we define each of the homoeologous genes created by the polyploidy as a "pillar" (*see* Fig. 1), we can see that, at each such pillar, POInT calculates the probability of the observed gene presence–absence data conditional upon each of the 2^n possible orthology relationships and also conditional upon a phylogenetic tree and a model for gene loss (The approach is conceptually identical for hexaploidies and octoploidies, but there are more possible orthology relationships to be tested). The HMM transition probability θ_j corresponds to the probability that orthology changes between

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_6,

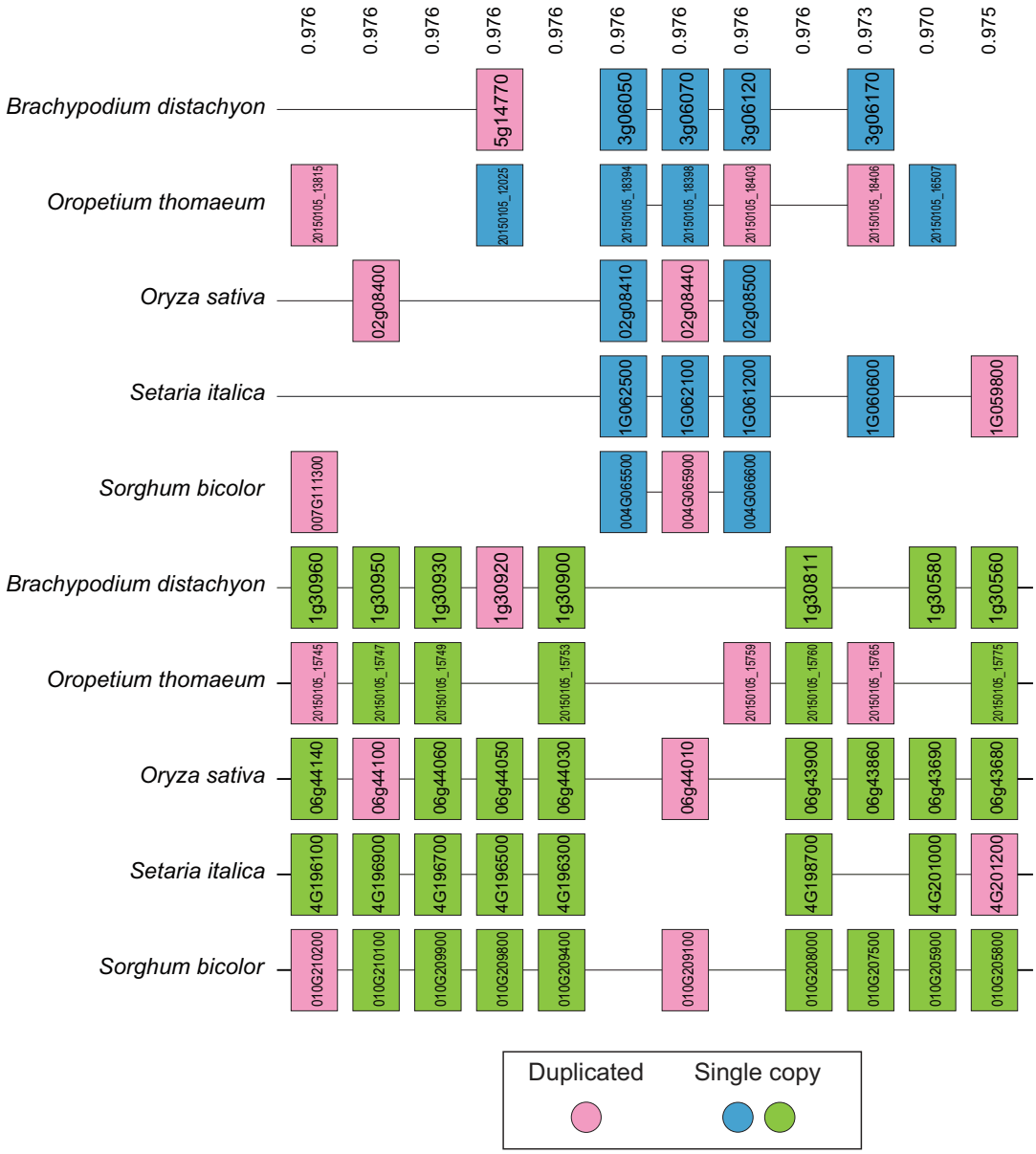


Fig. 1 Double conserved synteny blocks shared by *Brachypodium distachyon*, *Oropetium thomaeum*, *Oryza sativa*, *Setaria italica*, and *Sorghum bicolor* after the grass ρ whole genome duplication [5]. Pink genes are retained duplicates after the WGD, blue and green genes returned to single-copy and are from two different parental subgenomes. Gene synteny is indicated by the horizontal lines, and a gap represents gene loss after polyploidy. Posterior probabilities are shown on top of each pillar

syntenic neighbors at pillars $j-1$ and j, with the θ parameter is best thought of as an "error" term accounting for situations where the orthology assignments at the beginning of a synteny block differ from those at the end. This framework is how we are able to allow the presence–absence data to inform orthology relationships at

neighboring pillars. Pillars that are separated by synteny breaks are independent in their orthology relationships (i.e., $\theta_j = 1/2$). The model parameters and phylogenetic branch lengths are then fit to the pillar data using maximum likelihood and standard numerical optimization [6].

The power of this modeling framework is considerable. It produces probabilistic estimates of the orthology relationships between all of the homoeologous genes in the genome analyzed. These estimates can be used for problems such as identifying valid phylogenetic markers from polyploid genomes. We ourselves have used them as sequence-independent markers of locus history for the detection and analysis of gene conversions [7–9], as well as to link gene loss and preservation patterns after polyploidy to molecular functions [5, 10, 11] and to explore the role a genome ploidy increase played in the formation of a clade of parasitic nematodes [12]. POInT can also be used as a simulation engine: we have used it to confirm that polyploidies are indeed shared events between a number of genomes [2], test hypotheses about biases in gene loss [11] and to infer the *type* of polyploidy event present in certain genomes [12].

Our original POInT analyses were predicated on sets of homoeologous genes and an inferred ancestral genome order that the Wolfe lab developed for the yeast WGD as part of the Yeast Gene Order Browser (YGOB, http://ygob.ucd.ie) project [13, 14]. Recently, we have expanded POInT and developed a new software pipeline that allows us to examine arbitrary polyploidy events, so long as we have at least two genome sequences from species sharing the event as well as an outgroup genome lacking it [5]. This pipeline operates following a clear analogy to the process of creating datasets for phylogenetic analyses. In particular, we have an "alignment" step followed by the phylogenetic modeling step just sketched. We will generally describe running this pipeline under the assumption of a genome *duplication* (tetraploidy) for simplicity, using terms such as "duplicated regions." However, our approach is fully general, and we have also successfully applied it to hexaploidies and octoploidies.

The full suite of POInT tools is freely available from either our website (http://conantlab.org/POInT/POInT.html) or GitHub (https://github.com/gconant0/POInT). The majority of POInT is written in standard C++ (with the exception of one perl script) and has modest dependencies: the lapack linear algebra libraries for the likelihood computation [15], the GNU plotutils package (optional and used for producing visualizations), a random number generator (https://people.sc.fsu.edu/~jburkardt/f77_src/ranlib/ranlib.html), the OpenMP shared memory parallel library [16] and the Bioperl package, which is used only for inferring the initial set of gene orders in the various genomes [17]. The open-source code for the random number generator and the required lapack subroutines

are included in the distribution for convenience: on systems with lapack and blas preinstalled, the installed versions are used in preference to the copies in the distribution.

2 POInT Dataset Assembly/Synteny Block Inference

This "alignment" step seeks to assemble blocks of N-fold conserved synteny (NCS) produced by the various types of polyploidy (e.g., $N = 2$ for a tetraploidy, $N = 3$ for a hexaploidy and $N = 4$ for octoploidy). These blocks represent the products of the polyploidy and contain both surviving duplicated loci as well as regions where one or more of the homoeologs ("duplicates" from polyploidy) have been lost (Fig. 1). The inference process has three substeps: (1) homology inference, (2) inference of NCS blocks between a single polyploid genome and the nonpolyploid outgroup, and (3) the merging of NCS blocks from multiple polyploid genomes and the inference of an ancestral block order. The data required for **step 1** are as follows.

1. FASTA files with the coding regions and translations of all protein-coding genes in each polyploid genome and the nonpolyploid outgroup.
2. GFF files describing the relative contig or chromosome position of the coding regions/genes in those FASTA files for all of the genomes in question.

2.1 Step 1: Homology Inference

Our pipeline requires an outgroup or reference genome that is a relatively close relative of the polyploid genomes but which lacks the polyploidy. It is conceptually convenient to think of it as the "ancestral" prepolyploid genome, although that is not technically accurate. This homology search could be conducted in several ways, with a simple BLAST search [18] being perhaps the most obvious. Indeed, our first analysis [5] used a BLAST-like approach with low sensitivity but high computational efficiency that we developed from the SeqAn library [19]. More recently, we have found that GenomeHistory [20], a tool we developed about 20 years ago, while slow, gives good coverage of the genomes, including pairwise estimates of synonymous and nonsynonymous divergence (K_s and K_a), which are used in the next steps of the analysis. We note that the homology search only compares each polyploid genome with the nonpolyploid outgroup and with itself: we do not directly compare the polyploid genomes. When the homology search is complete, we store several pieces of information: (a) any pair of genes from the polyploid genome and the outgroup that pass homolog cutoffs in terms of percent amino acid identity of the pairwise alignment [21] of their two sequences, (b) pairs of genes both either from the polyploid or the outgroup genome that pass

similar filters and are hence potential tandem duplicates, and (c) The relative position of each of the genes in either "a" or "b" in their respective genomes. These data take the form of an ordered list of genes on contigs or chromosomes, omitting any genes with no homologs in "a" or "b."

2.2 Step 2: NCS "Scaffolding"

The next pipeline step uses the putatively single-copy genes from the nonpolyploid genome to infer the set of duplicated (or more) regions created by the polyploidy in each polyploid genome. Were all of the duplicate genes created by the polyploidy still extant, the identification of such regions would be trivial. In the face of duplicate loss, it becomes more complex.

We frame this problem as first defining a set A of n NCS blocks, each pillar A_i of which consists of one gene from the nonpolyploid outgroup ($A_i \in A | 1 \leq i \leq n$). Each A_i has elements $A_i(p_1) \ldots A_i(p_k)$, representing the k (= 2 for a tetraploidy) homologous genes created by the polyploidy. Associated to A_i are also all of the genes in the polyploid genome homologous to the nonpolyploid genome gene for that pillar $\{h_1 \ldots h_h\}$. At most k of these homologs can be assigned to $A_i(p_1) \ldots A_i(p_k)$. Finally, we define $O(A_1 \ldots A_n)$ to be the order of the pillars used for our analysis. Hence, $A_{O(i)}$ represents the ith pillar in this ordering. For a given $A_{O(i)}(p_l) | 1 \leq l \leq k$, define $A_{O(i+j)}(p_l)$ such that $j = \min(x; i + 1 \leq x \leq n)$ where $A_{O(i+x)}(p_l) \neq \emptyset$. In other words, $i + j$ is the next pillar after i in $O(A_1 \ldots A_n)$ with an assigned gene for parental subgenome l. From this framework, we can create a scoring function s for comparing different combinations of homolog assignments and pillar orders:

$$s = \sum_{i=1}^{n} \sum_{l=1}^{k} \begin{cases} 1 & A_{O(i)}(p_l) \text{ and } A_{O(i+j)}(p_l) \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

This equation indicates that s is the sum of the number of positions in $O(A_1 \ldots A_n)$ where the genes in each pillar are the genomic neighbors of the genes in the next nonempty position. One might wonder why the pillar order is estimated rather than simply being taken from the outgroup. However, in many cases, the outgroups used are rather distant relatives of the true polyploid progenitors and using these genome orders, even were the sequences in question perfectly assembled, would introduce not only all the postpolyploidy rearrangements seen in the polyploid genomes but also all those that occurred in the outgroup.

Equation 1 only allows us to score a particular combination of homeolog assignment and order (e.g., one point in a large state space). Unfortunately, the number of possible such points is enormous: there are $n!$ orders alone, without including the homeolog assignments. We therefore use simulated annealing [22, 23] to search for optimal values of s. Simulated annealing is a common

optimization approach that proposes small random changes to a current point in the state space: after each such move s is recomputed. The move is then accepted if either it improves the score or if the decrease in score is below a threshold that is tuned to decrease over the annealing run. This part of the analysis requires a certain degree of "art," as it is generally necessary to make increasingly long runs of the annealing algorithm until longer runs no longer produce meaningfully higher values of s.

From a practical perspective, this step is performed by the tool POInT_genome_scaffold in the POInT distribution. This program first collapses tandem duplicates in the outgroup and polyploid genomes (group "b" above). It also allows for a K_s or K_a filter on the homologs, allowing the user to remove distant homologs. It is similarly possible to take only the top m homologs (in terms of smallest K_s or K_a) for the analysis, a feature that is useful in genomes with high degrees of nested polyploidy. The output is a POInT-specific format that identifies the outgroup gene, the (up to) k homeologs and whether or not they have synteny support and in which direction.

2.3 Step 3: Pillar Merging and Global Order Inference

In the POInT distribution, we provide a script (POInT_merge.pl) that combines the optimal runs of POInT_genome_scaffold from across multiple genomes into a common set of pillars with at least one surviving homeolog across all of these genomes. This requirement for the presence of at least one gene in every genome limits the size of resulting datasets but is necessary because POInT cannot yet model the complete loss of an ancestral gene. The merge program allows the user to include pillars with different levels of synteny support, but we invariably only include pillars where every gene present is in synteny with at least one other (the default behavior). The genes from the outgroup genome are used as indices to allow the combination of genes from the different polyploid genomes.

POInT_merge.pl produces a set of pillars order by their genome position in the outgroup genome, since the $O(A_1 \ldots A_n)$ are different for each polyploid genome. This order will generally poor; it might, for example, include a large number of synteny breaks. Hence, the final step of the assembly pipeline is to run POInT_ances_order, which again uses simulated annealing to infer a pillar order with as few synteny breaks as possible. It does this by proposing different pillar orders and counting the resulting synteny breaks. In this case, the raw number of synteny breaks itself is used as the optimality function, and we find that a number of short annealing runs with POInT_ances_order produces orders with acceptable "tracking."

We provide a number of options with the POInT_ances_order program to provide adequate performance for polyploidies of different ages. In general, older polyploidies show more

rearrangements, meaning that the ancestral order inference is more difficult. Hence, we offer a "preoptimization" step that uses a greedy algorithm to initially order the data: this subroutine works best when the initial order is very poor (has many breaks). Second, we allow the user to control the quality of the "fixed blocks" in the inferences. Obviously, if two pillars are completely connected (show no synteny breaks between them), it is unproductive to rearrange within this "block," since the number of synteny breaks cannot improve. In fact, we know that these pillars must appear in these positions in the optimal order. Hence, POInT_ances_order performs rearrangements on these inferred blocks rather than the individual pillars: especially for datasets with few synteny breaks, this approach dramatically decreases the search space size.

When the dataset has a larger number of synteny breaks, this block approach will not reduce the search space as much: in those cases, we allow the user to specify a block cutoff *m*, corresponding to a number of synteny breaks tolerated within a block. For instance, by using a setting of -m:3 in POInT_ances_order, pairs (or more) of pillars with 3 or fewer synteny breaks will be treated as a single block and no rearrangements attempted within them. By starting with relatively larger values of this parameter, one can perform first coarse and then fine optimization (smaller *m*) and significantly reduce the search time for a reasonable ancestral order.

3 Modeling Polyploid Genome Evolution with POInT

Once these steps are complete, modeling of polyploidy events is possible. POInT itself fits Markov models of postpolyploidy gene loss to the pillars inferred above [2, 10]. In the current version, these models can be specified by the user in file format illustrated with the example in Fig. 2. Transition probabilities for these models are first computed by exponentiating the instantaneous rate matrix they define [24], and they are then fit to the pillar data using numerical optimization [6]. The models have states corresponding to single-copy, duplicated, triplicated, and quadruplicated genes, as well as the potential for variations of these states, for example, *fixed* duplicates that will remain in the duplicated state permanently. Figure 2 gives two example models we have developed for POInT: more models can be downloaded from our website (http://conantlab.org/software.html).

The running time of a POInT optimization can be significant: the algorithmic complexity of the algorithm is $O(2^{2n})$ for a tetraploidy (where n is the number of genomes) and greater for higher-level polyploidies. We have therefore implemented POInT as a parallel program using the OpenMP shared memory paradigm [16]. An analysis of ~4100 pillars across 11 taxa sharing a tetraploidy hence takes on the order of a few weeks on an Intel Phi

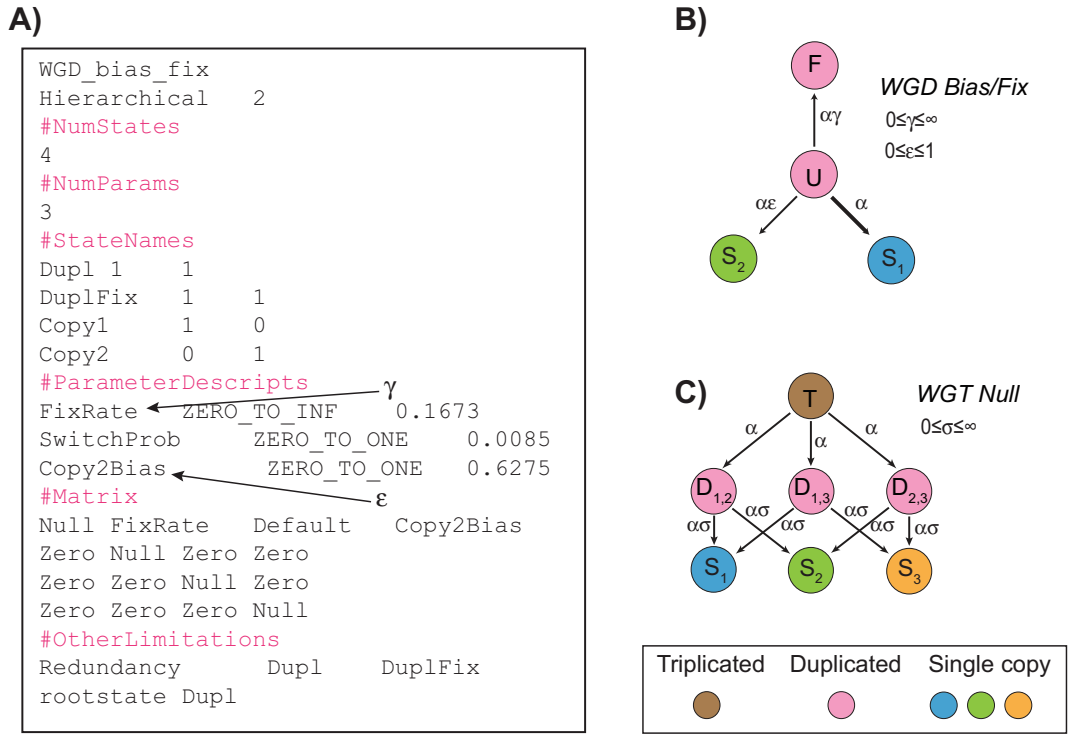


Fig. 2 Example model files required by POInT. (**a**) Description of a model file and estimated model parameters. (**b**) Example model of a WGD, *U* stands for an undifferentiated duplicated state, *F* is a fixed duplicate, and S_1 and S_2 represent single-copy states for the two parental subgenomes respectively. Here γ is the duplication retention parameter and ε is the biased fractionation parameter, while $\alpha\gamma$, $\alpha\varepsilon$, and α are transition rates between states. (**c**) Example model for a WGT. This model shows seven possible states (triplicated, duplicated, or single-copy states) after whole genome triplication and the transition rates between these states

coprocessor [25] and a roughly equivalent amount of time on 16 cores of modern high-end Intel Pentium processors.

There are a number of different ways to use POInT, and so we provide a few examples as illustrations.

Example 1 Testing the presence of duplicate fixation In the simplest use of POInT, we have a set of loci and a known phylogeny and wish to test the fit of different models of homeolog loss to these data. In this case, we would simply run POInT twice with a known input tree (provided with the -t:<Nexus treefile> option) and two different models, for instance a null model without duplicate fixation and an alternative with it. At the end of each run, a new tree file *inputtreefile.out* is created with the model parameters and the resulting log-likelihood. Since duplicate fixation is controlled by a single parameter (γ in Fig. 2), twice the difference in log-likelihood between the two models is distributed chi-square with 1 degree of freedom [26], allowing us to infer the significance (or lack thereof) of the improvement in fit from adding this parameter.

Example 2 Inferring the phylogenetic relationships between polyploid taxa If no tree is provided to POInT, it will compute the likelihood of all possible topologies for the taxa provided. Obviously, this computation could be very slow for large numbers of taxa: in our experience, it is only practical for datasets of five or fewer taxa in the context of a WGD, and three or fewer for a WGT. The topology with the highest log-likelihood is saved as a new tree file.

Example 3 Extracting orthologous genes from polyploid genomes POInT probabilistically computes all possible sets of orthology relationships between all of the loci in all of the polyploid genomes (whether or not a gene is actually present at that position in that genome). As such, an inference of the optimal orthology relationship can be made by computing the likelihood of a particular assignment relative to all of the possible assignments. Using the -p:<filename> option in POInT will result in the program saving the probability of every possible orthology relationship for every locus to that file. The header line gives the identity of all of these assignments, which can then be parsed manually or by other software.

Example 4 Testing the hypothesis of shared verses independent polyploidies When studying the yeast WGD, the question arose if the polyploidy observed in the genome of *V. polysporus* was the same event as that seen in *S. cerevisiae* [2]. In the POInT models, independent polyploidies are equivalent to having the shared root branch of the various taxa set to zero length (*V. polysporus* is the most distant relative of *S. cerevisiae* in this dataset). To test this hypothesis, we fit the pillar data in POInT while forcing the shared root branch to zero length, which is done by providing a tree file with a zero length root branch and using the argument -zerolengthfixed. We then run POInT again without this constraint. We next use the optimized tree with the forced zero-length root to simulate new genome duplications using the POInT simulation engine POInT_simulate. This program produces simulated polyploidies under an assumed model and tree topology. One can then use the main POInT code to analyze these simulations both under the forced zero-length assumption and omitting this requirement. The result is a distribution of differences in ln-likelihood for the simulated datasets that can be compared to that for the real dataset to assess if the root branch for the real dataset is inferred to be significantly nonzero, implying a shared polyploidy for the genomes in question.

Example 5 Evaluating the gene loss pattern after polyploidy - POInT can be used to statistically test for biased fractionation, that is, the preferential gene retention and unbalanced gene loss across different parental subgenomes after polyploidy. To

implement this test, POInT will be run twice, first using a null model with the biased fractionation parameter $\varepsilon = 1$ (Fig. 2). In this null model, the transition rates from the duplication state to each single-copy state are the same, modeling a scenario where gene loss in different subgenomes is equally likely. Then, in the alternative model, ε is allowed to fall between 0 and 1, introducing biased fractionation. The likelihood estimations from the two models can again be compared using a likelihood ratio test.

4 Conclusions

POInT and its associated helper programs are an extensible framework for studying the evolution of polyploid genomes. The tools are flexible in the types of polyploidy they can model and allow the user to define new Markov models of gene loss as needed. The software itself is freely available without license restrictions and runs on a variety of parallel and serial computing platforms.

Acknowledgments

The authors were supported by U.S. National Science Foundation grant NSF-IOS-1339156.

References

1. Scannell DR, Frank AC, Conant GC, Byrne KP, Woolfit M, Wolfe KH (2007) Independent sorting-out of thousands of duplicated gene pairs in two yeast species descended from a whole-genome duplication. Proc Natl Acad Sci U S A 104:8397–8402
2. Conant GC, Wolfe KH (2008) Probabilistic cross-species inference of orthologous genomic regions created by whole-genome duplication in yeast. Genetics 179:1681–1692
3. Durbin R, Eddy SR, Krogh A, Mitchison G (1998) Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge
4. Lander ES, Green P (1987) Construction of multilocus genetic linkage maps in humans. Proc Natl Acad Sci U S A 84:2363–2367
5. Emery M, Willis MMS, Hao Y, Barry K, Oakgrove K, Peng Y, Schmutz J, Lyons E, Pires JC, Edger PP, Conant GC (2018) Preferential retention of genes from one parental genome after polyploidy illustrates the nature and scope of the genomic conflicts induced by hybridization. PLoS Genet 14(3): e1007267em
6. Press WH, Teukolsky SA, Vetterling WA, Flannery BP (1992) Numerical recipes in C. Cambridge University Press, New York, NY
7. Evangelisti AM, Conant GC (2010) Nonrandom survival of gene conversions among yeast ribosomal proteins duplicated through genome doubling. Genome Biol Evol 2: 826–834
8. Scienski K, Fay JC, Conant GC (2015) Patterns of gene conversion in duplicated yeast histones suggest strong selection on a coadapted macromolecular complex. Genome Biol Evol 7(12):3249–3258
9. Casola C, Conant GC, Hahn MW (2012) Very low rate of gene conversion in the yeast genome. Mol Biol Evol 29(12):3817–3826
10. Conant GC (2014) Comparative genomics as a time machine: how relative gene dosage and metabolic requirements shaped the time-dependent resolution of yeast polyploidy. Mol Biol Evol 31(12):3184–3193
11. Conant GC (2020) The lasting after-effects of an ancient polyploidy on the genomes of teleosts. PLoS One 15(4):e0231356

12. Schoonmaker A, Hao Y, Bird D, Conant GC (2020) A single, shared triploidy in three species of parasitic nematodes. G3 10:225–233
13. Byrne KP, Wolfe KH (2005) The yeast gene order browser: combining curated homology and syntenic context reveals gene fate in polyploid species. Genome Res 15(10):1456–1461
14. Gordon JL, Byrne KP, Wolfe KH (2009) Additions, losses and rearrangements on the evolutionary route from a reconstructed ancestor to the modern *Saccharomyces cerevisiae* genome. PLoS Genet 5(5):e1000485
15. Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J, Du Croz J, Greenbaum A, Hammarling S, McKenney A, Sorensen D (1999) LAPACK Users' Guide, 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia, PA
16. Dagum L, Menon R (1998) OpenMP: an industry standard API for shared-memory programming. IEEE Comput Sci Eng 5(1):46–55
17. Stajich JE, Block D, Boulez K, Brenner SE, Chervitz SA, Dagdigian C, Fuellen G, Gilbert JG, Korf I, Lapp H, Lehvaslaiho H, Matsalla C, Mungall CJ, Osborne BI, Pocock MR, Schattner P, Senger M, Stein LD, Stupka E, Wilkinson MD, Birney E (2002) The Bioperl toolkit: Perl modules for the life sciences. Genome Res 12(10):1611–1618
18. Altschul SF, Madden TL, Schaffer AA, Zhang JH, Zhang Z, Miller W, Lipman DJ (1997) Gapped blast and Psi-blast: a new-generation of protein database search programs. Nucleic Acids Res 25(17):3389–3402
19. Doring A, Weese D, Rausch T, Reinert K (2008) SeqAn an efficient, generic C++ library for sequence analysis. BMC Bioinformatics 9: 11
20. Conant GC, Wagner A (2002) GenomeHistory: a software tool and its application to fully sequenced genomes. Nucleic Acids Res 30(15):3378–3386
21. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443–453
22. Kirkpatrick S, Gelatt CDJ, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
23. Conant GC, Wolfe KH (2006) Functional partitioning of yeast co-expression networks after genome duplication. PLoS Biol 4:e109
24. Muse SV, Gaut BS (1994) A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. Mol Biol Evol 11(5):715–724
25. Jeffers J, Reinders J (2013) Intel Xeon Phi coprocessor high performance programming. Morgan Kaufmann, Waltham, MA
26. Sokal RR, Rohlf FJ (1995) Biometry, 3rd edn. W. H. Freeman and Company, New York, NY

Part II

Omics Analysis

Chapter 7

Searching for Homologous Genes Using Daisychain

Philipp E. Bayer and David Edwards

Abstract

Genome assemblies have become a standard tool of genomics research and are relatively inexpensive to produce due to falling sequencing costs. For many species, there are now several reference-grade genome assemblies. However, comparing different assemblies or the same or related individuals is not an easy task, especially with different levels of quality of assembly and annotation. Tools are needed to visualise related genes with different IDs across genome assemblies. Here, we present a workflow to search and visualise related genes using Daisychain, a web-based tool aimed at researchers who wish to compare genes between assemblies.

Key words Genome annotation, Daisychain, Website, Annotation comparison, Genomics

1 Introduction

The first complete genome assemblies from *Drosophila melanogaster*, *Arabidopsis thaliana*, and *Homo sapiens* are now more than 20 years old [1–3] using first-generation sequencing. Since then second, then third-generation sequencing have revolutionised how genomes are assembled. Genomic reads have become longer and more accurate, with reads now commonly achieving lengths of more than 10,000 bp for the cost of a few thousand dollars per genome. These third-generation reads make it possible to assemble highly contiguous genome assemblies.

Hundreds of plant genome assemblies are now available. For *Brassica napus* (canola), there are now four second-generation sequencing assemblies [4–6] and nine [7, 8] third-generation sequencing assemblies available, along with one pangenome representing canola's gene content [9]. For *Triticum aestivum* (bread wheat), there are now at least 12 genome assemblies [10–12] and one pangenome [13] available. Breeder interest in genetic diversity of *Glycine max* (soybean) has so far led to three published soybean pangenomes [14–16] and more than 13 reference-quality genomes [17–20].

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_7,

Compared with eukaryotes, plants have larger and more complex genomes with many repeats, inversions, deletions, and duplications, making it harder to compare genes of interest across assemblies of different cultivars, with gene content specific to single or groups of cultivars plants having made it necessary to assemble plant pangenomes to reflect the true gene content of a species [21–23]. Tools such as OrthoMCL [24], GET_HOMOLOGUES-EST [25], or Orthofinder [26] are used to compare gene content across genome assemblies and pangenomes. However, these tools require command line knowledge and have no graphical interface, making it hard to find orthologous genes across the many assemblies available.

Here, we describe common steps to compare genome assemblies using Daisychain. Daisychain is a web portal based on KnetMiner's KnetMaps.js [27]. Daisychain hosts genome annotations and clustering results that allow users to compare their genes of interest with homologs and paralogs in genome assemblies and to investigate 5′/3′ neighbors of homologous genes.

2 Materials

2.1 Software

Daisychain is hosted at http://daisychain.appliedbioinformatics.com.au/.

2.2 Hardware

This protocol is written for any computer connected to the Internet with a recent Internet browser. This workflow was tested using Mozilla Firefox 89.0.2 on Windows 10.

3 Methods

3.1 Searching for Genes

We begin by opening the Daisychain web page which shows the landing page (Fig. 1). The landing page presents us with two options: either to enter a gene of interest ID or to enter a gene of interest sequence. Currently, Daisychain hosts two projects, one storing *B. napus* genome assemblies and one storing *Escherichia coli* genome sequences (*see* **Note 1**).

When we enter a gene ID, the system will search this ID and retrieve any matching gene IDs in a results tree. When we enter a gene sequence in FASTA format, the system will run blastn [28] and retrieve the highest-ranked hit in each stored assembly. Figure 2 shows an example tree of results for BnaA01g00070D.

We select genes of interest by ticking the box next to the gene ID. These genes will be displayed in the gene-based view.

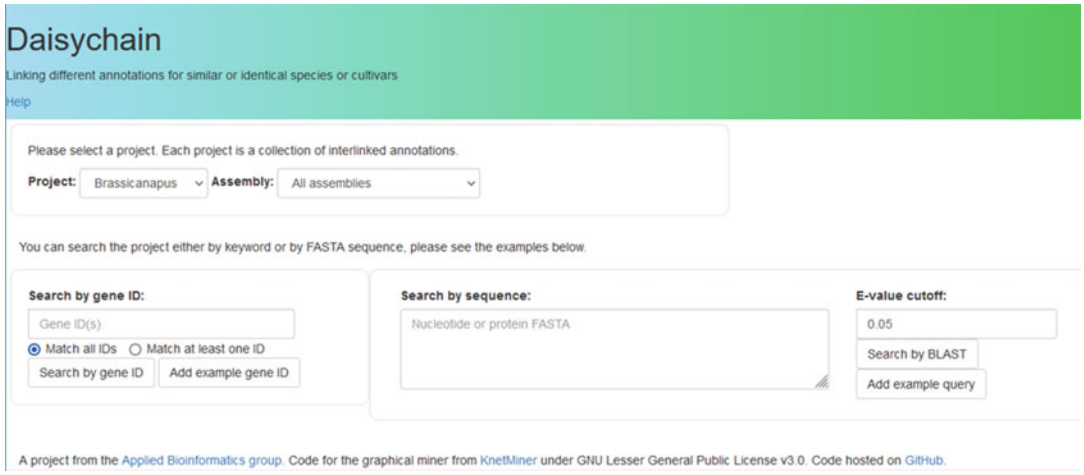


Fig. 1 Start page for Daisychain hosted at daisychain.appliedbioinformatics.com.au

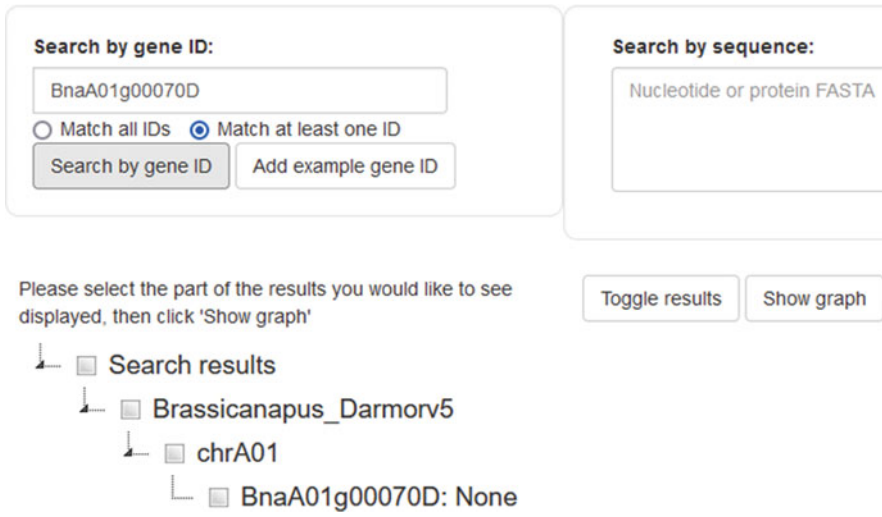


Fig. 2 Search results for gene ID BnaA01g00070D

3.2 Displaying Gene Homology

We now click on "Show graph" to move the selected genes into the gene-based view, (Fig. 3a) (*see* **Note 2**). In Fig. 3, we are presented with the gene and can use the right mouse button to search for this gene's homologous genes in other assemblies (Fig. 3b). By clicking on the assembly names, we can toggle the display of homologous genes from specific assemblies. We right-click on BnaA01g00070D and select "Show homologs" to display related genes. After these genes are displayed, we reset the view by selecting "Circular layout" instead of "CoSE layout" in the layout dropdown menu. We can now see two homologous genes in the Tapidor assembly, one in the Darmor v8.1 assembly, two in the ZS11 assembly, and four in the Darmor v5 assembly. We then right-click on BnaC01g00980D and

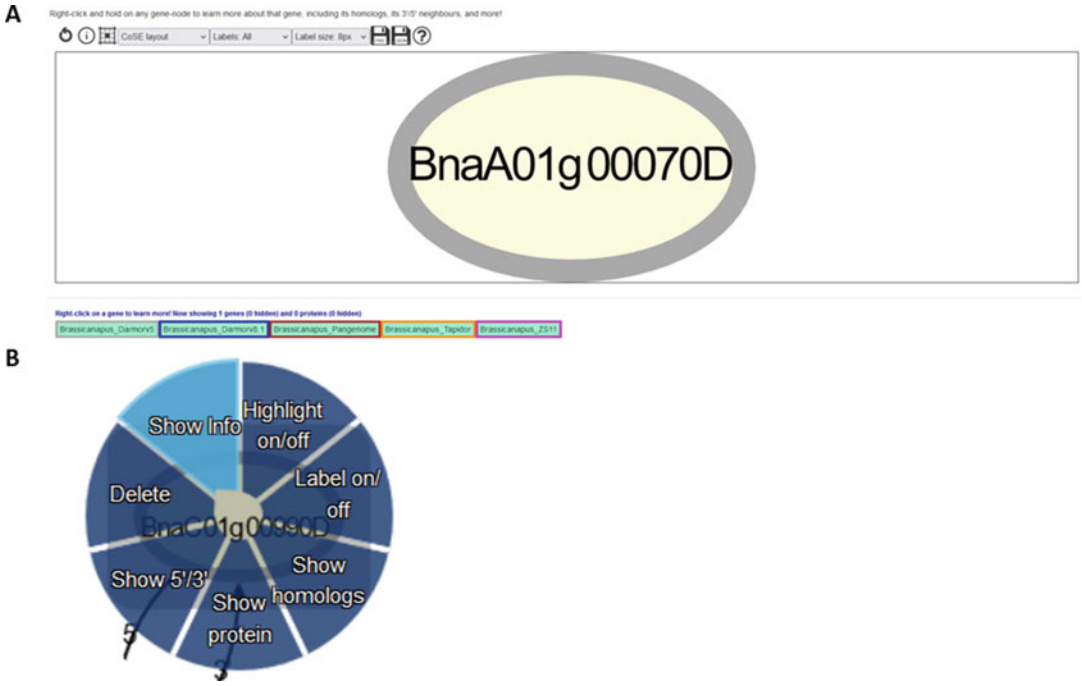


Fig. 3 (**a**) Initial gene view for BnaA01g00070D with no homologs or neighbouring genes displayed. (**b**) Context menu for a gene activated by clicking the right mouse button

select "Show 5′/3′" to display the neighbors of this gene in the Darmor v5 assembly and use the left mouse button to move the position of the genes in the display. By clicking on the assembly names below the figure we remove the display for two of the assemblies (Fig. 4).

Further information about the currently displayed genes is available. We can display the protein as another node in the graph by selecting "Show protein," or we can view the nucleotide and amino acid sequence of the gene and the protein by selecting "Show info" (*see* **Note 3**).

3.3 Displaying Gene Collinearity

Using the 5′/3′ feature we can display gene collinearity across assemblies. We want to investigate the 5′/3′ neighbors in three assemblies around the gene BnaC01g00770D2. We search for this gene, open the gene viewer, then click on "Show homologs" for this gene and deactivate display of the pangenome and the Tapidor assembly.

Then, for each homolog, we right-click and show the 5′/3′ neighbors and manually arrange the neighbors. When adding new neighbors for an assembly, homologous connections will be automatically displayed. The end results are shown in Fig. 5. The 5′/3′ neighbors of BnaC01g00770D2 show collinearity across the three assemblies with no missing genes.

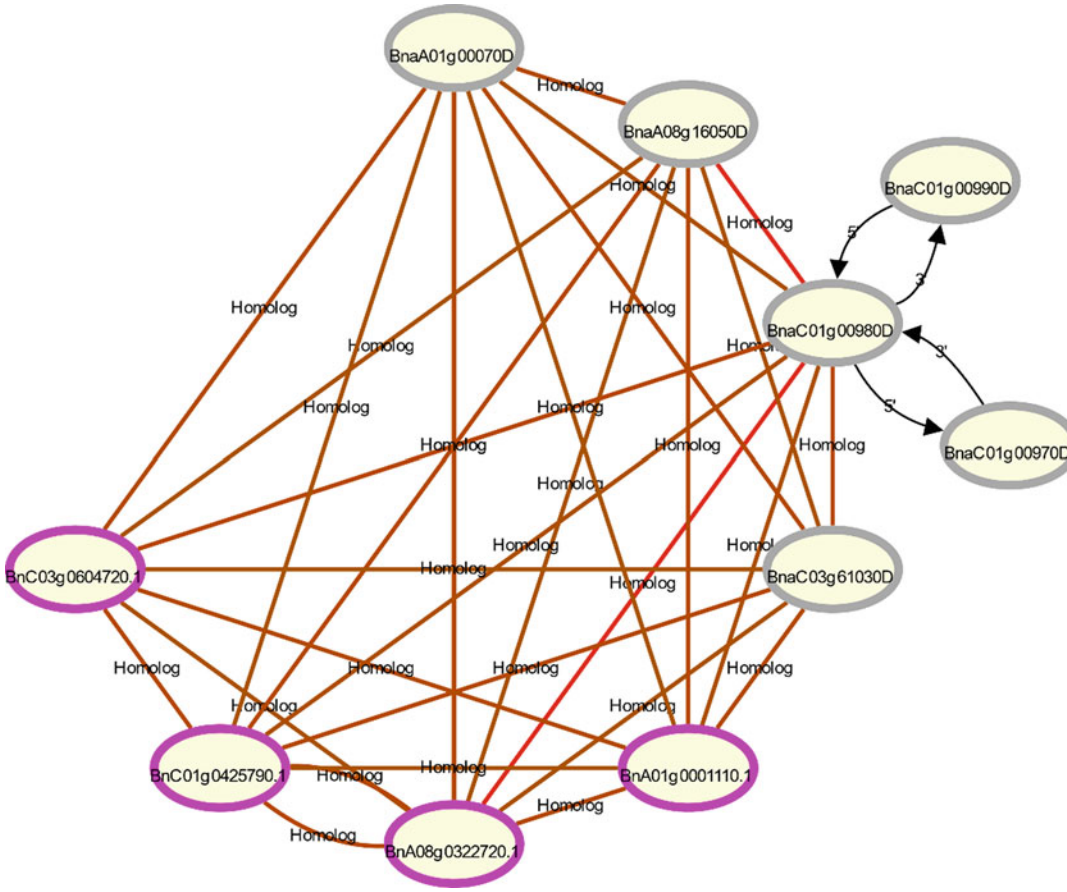


Fig. 4 Final homologous gene graph across two annotations including 5′/3′ neighbours for BnaC01g00980D (gray: Darmor-bzh v4, purple: ZS11)

At this point, we have investigated and displayed all homologous genes for our gene of interest. We have searched for gene neighbors for one of the homologous genes and displayed sequence information about the homologous genes. Daisychain is a flexible, web-based tool to allow researchers to identify homologs and paralogs between and within annotated genomes.

4 Notes

1. The example species are hosted on a public server. Daisychain is developed so that users can simply build their own genome comparison databases either for local private use or for public presentation.
2. It is also possible to display several genes at the same time. For this workflow, we have used only a single gene.
3. The graph can be exported in JSON format and PNG format by clicking on the two labels above the gene view.

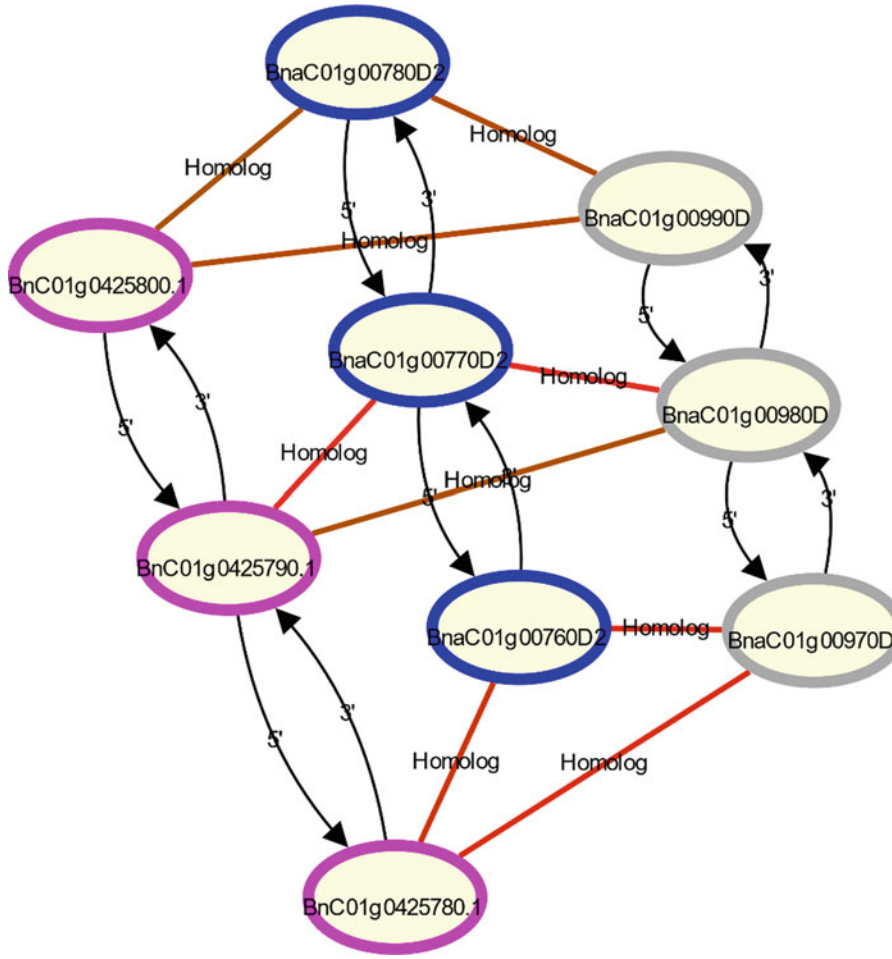


Fig. 5 Gene collinearity across three assemblies around the candidate gene BnaC01g00770D2 (gray: Darmor-bzh v4, blue: Darmor-bzh v8, purple: ZS11)

References

1. Adams MD et al (2000) The genome sequence of *Drosophila melanogaster*. Science 287(5461):2185–2195
2. Kaul S et al (2000) Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana*. Nature 408(6814):796–815
3. Venter JC et al (2001) The sequence of the human genome. Science 291(5507): 1304–1351
4. Chalhoub B et al (2014) Early allopolyploid evolution in the post-Neolithic *Brassica napus* oilseed genome. Science 345(6199):950–953
5. Bayer PE et al (2017) Assembly and comparison of two closely related *Brassica napus* genomes. Plant Biotechnol J 15(12):1602–1610
6. Sun F et al (2017) The high-quality genome of *Brassica napus* cultivar 'ZS11' reveals the introgression history in semi-winter morphotype. Plant J 92(3):452–468
7. Song JM, Guang Z, Hu J et al (2020) Eight high-quality genomes reveal pan-genome architecture and ecotype differentiation of *Brassica napus*. Nat Plants 6:34–45
8. Lee H, Chawla HS, Obermeier C et al (2020) Chromosome-scale assembly of winter oilseed rape Brassica napus. Front Plant Sci 11:496
9. Hurgobin B et al (2018) Homoeologous exchange is a major cause of gene presence/absence variation in the amphidiploid *Brassica napus*. Plant Biotechnol J 16(7):1265–1274

10. Zimin AV et al (2017) The first near-complete assembly of the hexaploid bread wheat genome, *Triticum aestivum*. Gigascience 6(11):gix097
11. International Wheat Genome Sequencing Consortium (2018) Shifting the limits in wheat research and breeding using a fully annotated reference genome. Science 361(6403): eaar7191
12. Walkowiak S, Gao L, Monat C et al (2020) Multiple wheat genomes reveal global variation in modern breeding. Nature 588:277–283
13. Montenegro JD et al (2017) The pangenome of hexaploid bread wheat. Plant J 90(5): 1007–1013
14. Liu Y et al (2020) Pan-genome of wild and cultivated soybeans. Cell 182(1):162–176.e13
15. Torkamaneh D, Lemay MA, Belzile F (2021) The pan-genome of the cultivated soybean (PanSoy) reveals an extraordinarily conserved gene content. Plant Biotechnol J 19: 1852–1862
16. Bayer PE et al (2021) Sequencing the USDA core soybean collection reveals gene loss during domestication and breeding. Plant Genome 2021:e20109
17. Schmutz J et al (2010) Genome sequence of the palaeopolyploid soybean. Nature 463(7278):178–183
18. Shen Y et al (2018) De novo assembly of a Chinese soybean genome. Sci China Life Sci 61(8):871–884
19. Valliyodan B et al (2019) Construction and comparison of three reference-quality genome assemblies for soybean. Plant J 100(5): 1066–1082
20. Chu JSC et al (2021) Eight soybean reference genome resources from varying latitudes and agronomic traits. Sci Data 8(1):1–8
21. Danilevicz MF et al (2020) Plant pangenomics: approaches, applications and advancements. Curr Opin Plant Biol 54:18–25
22. Bayer PE et al (2020) Plant pan-genomes are the new reference. Nat Plants 6:914–920
23. Golicz AA et al (2020) Pangenomics comes of age: from bacteria to plant and animal applications. Trends Genet 36(2):132–145
24. Li L, Stoeckert CJ Jr, Roos DS (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. Genome Res 13(9): 2178–2189
25. Contreras-Moreira B et al (2017) Analysis of plant pan-genomes and transcriptomes with GET_HOMOLOGUES-EST, a clustering solution for sequences of the same species. Front Plant Sci 8:184
26. Emms DM, Kelly S (2019) OrthoFinder: phylogenetic orthology inference for comparative genomics. Genome Biol 20(1):1–14
27. Hassani-Pak K et al (2021) KnetMiner: a comprehensive approach for supporting evidence-based gene discovery and complex trait analysis across species. Plant Biotechnol J 19(8): 1670–1678
28. Camacho C et al (2009) BLAST+: architecture and applications. BMC Bioinformatics 10:421

Chapter 8

Detecting MicroRNAs in Plant Genomes with miRkwood

Sylvain Legrand, Isabelle Guigon, and Hélène Touzet

Abstract

We present miRkwood, a comprehensive software tool developed to identify microRNAs and their precursor in plant genomes, with or without small-RNA-seq sequencing data. We describe how to install the software, how to set up and run it, and how to explore and analyse the results: genomic annotations, secondary structure of the precursor, alignments, reads distribution.

Key words Bioinformatics, MicroRNA, Small RNA sequencing, High throughput sequencing

1 Introduction

MicroRNAs (miRNAs) have been shown to play pivotal roles in growth and development of plants. Finding and annotating them is now greatly facilitated by joint advances in high-throughput sequencing of small RNAs (sRNA-Seq) and the analytical protocols developed to handle these data.

In this chapter, we introduce miRkwood, a bioinformatics software that has been specifically designed to identify precursors of miRNAs (pre-miRNAs) and miRNAs in sRNA-Seq data obtained from plant samples. miRkwood implements a complete workflow that gathers several complementary computational steps in an efficient and transparent manner for the user: identification of genome regions that have been enriched with aligned reads, treatment of duplications and existing annotations, analysis of the secondary structure of the stem-loop precursor, evolutionary conservation, quality of the duplex formed by the guide miRNA and the passenger strand. Experimental results show that miRkwood is highly sensitive and that it is able to cope with the heterogeneity of miRNAs and their precursors [1]. Moreover, it provides a user-friendly interface including visualization utilities that makes it easy to explore the data.

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_8,

The software miRkwood is available as a web server (http://bioinfo.cristal.univ-lille.fr/mirkwood) and as a stand-alone local version (*see* **Notes 1** and **2**). In this chapter, we give a thorough description of the stand-alone version, which is best adapted to large-scale projects and especially easy to install with a docker container, but most of the information is also useful for the web version. Results are available as a variety of formats (CSV, YAML, GFF, FASTA, ORG), which make it especially convenient to integrate miRkwood into a more complete analysis workflow for downstream analyses.

2 Materials

2.1 Installation of miRkwood

The software miRkwood is written in C/C++ and Perl. It is freely available under GNU Affero General Public License v3.0. There are three modes of installation: with a Docker container, with a virtual machine (with virtualbox and ansible), or with a bash script. In the following, we describe the installation via Docker, which is the easiest way to proceed. Instructions for the two other modes can be found at https://github.com/miRkwood-RNA/miRkwood.

1. If needed, first install Docker on the computer. It is available on Linux, MacOS and Windows 10. *See* https://docs.docker.com/get-docker/.
2. A Docker image for miRkwood is stored at https://hub.docker.com/r/iguigon/mirkwood/. You can download it with the command

```
$ sudo docker pull iguigon/mirkwood:latest
```

Note that this will automatically include all tools required by miRkwood: bedtools [2], Vienna RNA package for RNA folding [3], miRdup for duplex validation [4] and the Java lightweight Applet VARNA for structure visualization [5].

2.2 Other Requirements

1. The software miRkwood does not involve the preprocessing steps for cleaning, trimming, and mapping the sequencing reads. It means that you will need to run the appropriate tools for those tasks separately. We give a protocol in Subheading 3.1.
2. Visualization and exploration of the results require to have standard tools on your computer, such as a web browser for html files, a genome browser for GFF files.

2.3 Your Data

1. The software miRkwood necessitates a set of small RNA sequencing reads and a reference sequence, and outputs all locus for miRNA precursors in the reference sequence. Optionally, it is also likely to provide annotation files for the reference sequence.
2. sRNA-Seq has been established as the gold standard method for high-throughput identification of sRNAs ranging between 18 and 30 nucleotides in length, including miRNAs. sRNA libraries are obtained from total RNA using dedicated preparation kits and usually sequenced using the Illumina technology to obtain reads with a low error rate. These reads should be in a FASTQ file or a FASTA file. The choice between these two formats depends on the mapper that you will use to build the alignments. It is also possible to directly provide a BAM file (*see* Subheading 3.1).
3. The reference sequence should be in a FASTA file (or-multi-FASTA). Any plant genome assembly can be used, whatever its state of completion (reference, draft, or resequenced genome). Elsewhere, miRNA genes are usually transcribed by RNA polymerase II and the resulting transcripts are capped, present a poly(A) tail and as a consequence are represented within RNA-seq data. Hence, users can alternatively provide a transcriptome assembly in the absence of a genome assembly. Regarding the syntax of the file, the name of each sequence (chr1, chloroplast, ...) should appear at the beginning of the FASTA heading in order to be correctly parsed and used by miRkwood.
4. When available, the reference sequence can also come with annotations in GFF3 format. Two kinds of GFF3 files are useful: those containing already known precursors of miRNA for the genome (such as a miRBase file) and those containing any other type of functional features (such as CDS, tRNA, rRNA). The use of these files is described in Subheading 3.2.

3 Methods

3.1 Preparation of Sequencing Reads

Before running miRkwood, it is necessary to first process the raw sRNA-Seq reads: cleaning them and mapping them on the reference genome/transcriptome in order to build a BED file that will be provided to the software.

1. Clean your raw sequencing reads. If your reads have already been processed, you can skip this step. Otherwise, this is a mandatory step. First, you should remove sequence adapters using, for example, the Cutadapt software [6]:

```
$ cutadapt -a AACCGGTT -o adapter_trimmed_short_reads.fastq
raw_short_reads.fastq
```

Next, you have to run quality control, in order to filter out too short or too long sequences and to remove or to trim the low-quality sequences. This can be achieved using Prinseq [7], with this command line as example.

```
$ prinseq-lite.pl -fastq adatpter_trimmed_short_reads.fastq
-min_len 18 -max_len 25 -noniupac -min_qual_mean 25 -trim_-
qual_right 20 -ns_max_n 0
```

Here we keep only the sequences between 18 and 25 nt with a mean quality of at least 25 (phred score) and composed of nucleotides ACGT. The sequences are trimmed by quality score from the 3'-end with a value of 20 as threshold.

2. Generate a SAM/BAM file that contains the alignments of the expressed reads with the reference genome. For this task, you can use Bowtie [8] with the following parameters (exact matching). Any other read mapper can also do the job.

```
$ bowtie -v 0 -f/q --all --best --strata -S genome.fasta
clean_short_reads.fastq mapped_reads.sam
```

3. Convert the SAM/BAM file into a BED file: this is done with the custom script mirkwood-bam2bed.pl, which is provided in the docker file (*see* Subheading 2.1) or which is available at (https://bioinfo.cristal.univ-lille.fr/cgi-bin/mirkwood/web_scripts/getScript.pl?file=mirkwood-bam2bed.pl). This script accepts a SAM or a BAM file. The file type is automatically detected from the name extension. In practice, the BED file is up to 10 times smaller than the BAM file, while retaining all information needed to conduct the analysis.

```
$ mirkwood-bam2bed.pl --in mapped_reads.sam --bed mapped_-
reads.bed --min <int> --max <int>
--in <PATH>: path to your input file (format BAM or SAM)
--bed <PATH>: path to your output BED file
--min <int X> : keep only reads with length >= X (default 18)
--max <int Y> : keep only reads with length <= Y (default 25)
--depth <int N> : keep only reads with depth > N (default 1)
```

The generated BED file has the following syntax.

```
1 18092 18112 AAACGTGTAGAGAGAGACTCA 1 -
1 18094 18118 GATTCTTTTGTTTGCCACT 2 +
1 18096 18119 TCGATAGGATCAAGTACATCT 1 +
1 18100 18124 AAGAAGAAAAAGAAGAAGAAGAAG 9 +
```

In this file, each line is a unique read. The fields are, from left to right: name of the chromosome, starting position, ending position, read sequence, number of occurrences of the read

in the data, strand. Positions follow the BED numbering convention: the first base of the chromosome is considered position 0 (0-based position) and the feature does not include the ending position.

3.2 Running miRkwood

1. Choose the set of parameters. miRkwood comes with a series of options that allow to customize the search and enhance the results.
2. The first set of optional parameters enables you to select the reads that will be used for the search according to the annotation and the structure of the reference sequence.

   `--mirbase <mirna.gff3>`: This option permits to specify a GFF/GFF3 file containing genome coordinates of already known pre-miRNAs and miRNAs for the genome in question. miRkwood uses this file to detect reads that match these loci, and classify them as known RNAs. The syntax is that of miRBase GFF3 files: the hairpin precursor sequences have type "miRNA_primary_transcript", and mature sequences have type "miRNA". The recognized attributes are "name", "ID", and "derives_from". The name of each sequence should be the same as in the FASTA file provided for the genome.

   `--gff <file.gff3>`: When annotations of the reference genome are available, it is advised to filter out the alignments based on the existing annotation in order to clean the data and remove reads emanating from RNA degradation products. With this option, all reads that intersect a feature present in the given GFF/GFF3 file(s)—such as CDS, tRNA, rRNA, snoRNA—are excluded from the analysis. In these GFF/GFF3 file(s), the name of each sequence should be the same as in the FASTA file provided for the genome.

   `--min-repeats <int X>`, `--max-repeats <int Y>`: These two options allow to remove multiple mapped reads, that can be due to transposable elements for instance (*see* **Note 3**). All reads that are mapped to less than X or more than Y loci on the reference sequence are discarded. By default, $X = 0$ and $Y = 5$.
3. The second set of optional parameters concerns the selection of novel miRNAs and their precursors. By default, we propose to narrow the search in order to achieve a good balance and minimize the number of false positive predictions while maintaining a high sensitivity. It is possible to relax this selection to find all potential precursors. This increases the sensitivity of miRkwood, but also yields a larger number of false positive predictions.

`--no-filter-mfei`: MFEI is the *minimum folding free energy index*. It measures the thermodynamic stability of the precursor hairpin and is calculated using the Matthews-Turner nearest neighbor model [9]. By default, only pre-miRNA candidates with a MFEI smaller than −0.6 are selected. This allows to exploit the fact that pre-miRNA candidates with a high MFEI are more likely to be false positives. Only 4% of miRBase precursors do not pass this threshold. When this option is activated, all candidates are kept regardless of their MFEI value. Default: off.

`--no-filter-bad-hairpins`: By default, miRkwood rejects low quality hairpin precursors: hairpins found by miRkwood with a global score of 0 (*see* Subheadings 3.6 and 3.7 and **Note 4**) and no alignment with miRbase. When this option is activated, all hairpin precursors are kept. Default: off.

4. The last three options allow to calculate additional information for each precursor of novel miRNAs found by miRkwood.

   `--align`: This option permits to check if the precursor matches the Viridiplantae mature miRNAs of miRBase V22. All alignments up to three errors (mismatches, insertions, deletions) are accepted. The file for miRBase V22 is provided with miRkwood release. It can be modified or replaced by the user ({miRkwood_path}/cgi-bin/data/MirbaseFile.txt). Default: off.

   `--shuffles`: In addition to the MFEI, the significance of the stability of the hairpin structure can also be measured by comparison with other equivalent sequences. Bonnet et al. have established that the majority of the pre-miRNA sequences exhibit a Minimum Free Energy that is lower than that for shuffled sequences [10]. This option allows to compute the probability that, for a given sequence, the MFE of the secondary structure is different from a distribution of MFE computed with 300 random sequences with the same length and the same dinucleotide frequency. Note that this option is time-consuming. Default: off.

   `--varna`: This option will generate a drawing of the secondary structure of the precursor with VARNA. Note that this option is time-consuming. Default: off.

5. Launch mirkwood. The command is:

```
$ mirkwood-bed.pl [options]* --input <input.bed> --output
```

```
<output_dir> --genome <genome.fasta>
 Options:
 --mirbase <mirbase_file.gff3>
 --gff <annotations_file.gff3>
 --min-repeats <int>
 --max-repeats <int>
 --no-filter-bad-hairpins
 --no-filter-mfei
 --align
 --shuffles
 --varna
```

The three first parameters are mandatory. <input.bed> is the BED file containing the positions of all mapped reads (*see* Subheading 3.1). <output_dir> is the directory where all results files are placed. If the specified directory does not exist, it is created. <genome.fasta> is a fasta or a multi-fasta file that contains the sequences of the reference genome. All other parameters are optional and have been described in the preceding paragraph.

Execution can take several hours. Once the software has finished running, the result files are available in <output_dir>, including a file named <finished> that indicates the end of the job.

3.3 Exploring the Results

In this chapter, we have used the *Arabidopsis thaliana* reference genome (TAIR10) and the dataset SRR6192867 (SRA NCBI) [11] to illustrate how miRkwood works. Briefly, total RNAs were obtained from mature inflorescence tissues from *Arabidopsis thaliana* Col-0. sRNA libraries were prepared using the TruSeq Small RNA Library Preparation Kit (Illumina) and sequenced with an Illumina HiSeq 2500. The dataset is composed of a total of 10,608,354 50 bp raw reads. According to the preprocessing method detailed in Subheading 3.1, 7,709,581 clean reads with a size between 18 and 25 nt were obtained, including 6,833,028 reads that could be mapped to the *Arabidopsis thaliana* genome. We generated the BED file from the BAM file with mirkwood_-bam2bed with default parameters, hence removing all reads of depth 1:

```
$ mirkwood-bam2bed.pl --in <SRR6192867.sam/bam> --bed
<SRR6192867.bed>
```

The resulting BED file contains 5,682,035 reads (500,143 unique reads). We then launched miRkwood with the following command-line:

```
$ mirkwood-bed.pl
 --input SRR6192867.bed
 --mirbase Arabidopsis_thaliana_miRBase.gff3
 --gff Arabidopsis_thaliana_CDS.gff
 --gff Arabidopsis_thaliana_otherRNA.gff
 --genome TAIR10_chr_all.fas
 --min-repeats 0
 --max-repeats 5
 --shuffles
 --align
 --varna
 --output results_dir
```

Where the GFF3 files `Arabidopsis_thaliana_CDS.gff` and `Arabidopsis_thaliana_otherRNA.gff` contain annotations for CDSs and noncoding RNAs for the TAIR10 (downloaded from the TAIR project) respectively and the GFF3 file `Arabidopsis_thaliana_miRBase.gff` contains annotations for the known pre-miRNAs and miRNAs of TAIR10 for chromosomes 1–3 (downloaded from miRBase). Annotations for chromosomes 4 and 5 were removed in order to ensure that "novel" miRNAs can be detected.

3.4 Organization of the Main Results Directory

1. All results are available in the <output> directory. An example is given in Fig. 1.
2. In this directory, the files `summary.txt`, `SRR6192867_CDS.tar.gz`, `SRR6192867_filetered.bed`, `SRR6192867_miRNAs.tar.gz`, `SRR6192867_multimapped.tar.gz`, `SRR6192867_orphan_clusters.tar.gz`, `SRR6192867_orphan_hairpins.tar.gz` and `SRR6192867_otherRNAs.tar.gz` correspond to the course of the algorithm, its various stages, and are described in Subheading 3.5. The folder Results contains the detailed results for known miRNAs (Subheading 3.6) and novel miRNAs (Subheading 3.7) in a human readable form. Here you can also find csv, gff, fasta, txt and org files that offer alternative formats. These results are accompanied by folders alignment, read clouds and sequences, which are described in Subheading 3.8. Lastly, we also provide an export in YAML (*see* Subheading 3.9). `Run_options.cfg`, `log.log` and `finished` are usual utility files.

3.5 Summary

1. The file <`summary.txt`> can be consulted first. It is composed of two sections: options summary and results summary. An example is given in Fig. 2.
2. The options summary section recalls the parameters and the files used when launching miRkwood.

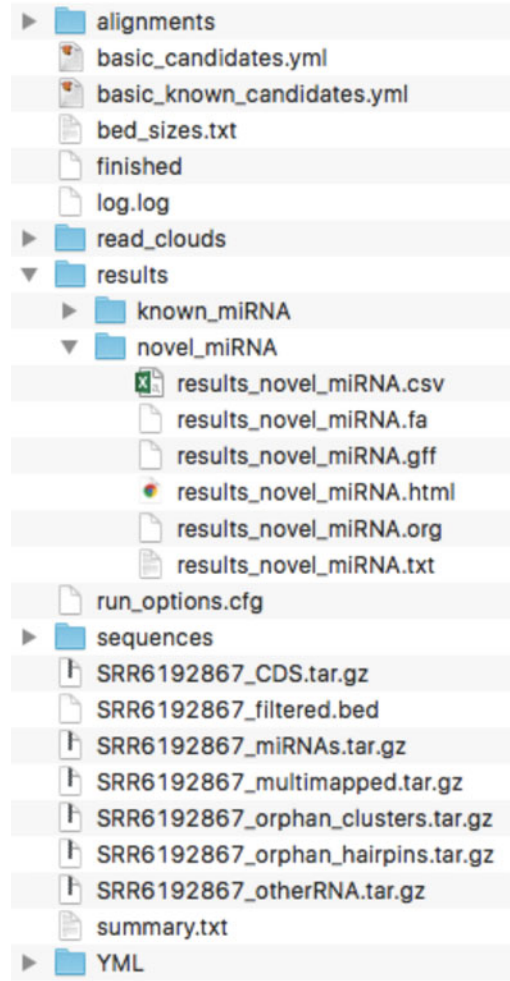


Fig. 1 Organization of the main results directory

3. The results summary section gives a general overview of the outcomes of the program. Throughout the steps of the algorithm, alignments from the initial BED file are classified into a number of categories that are visible in this file. First, option `--mirbase` enables to select matches corresponding to precursors of annotated miRNAs (*see* Subheading 3.1). This constitutes the category of *known miRNAs*, that is described in further details in Subheading 3.6. Second, matches that intersect with some feature from an annotation file (option `-gff`) are discarded, and saved separately in a compressed BED file (named after the name of the input GFF file). Third, multiple mapped reads and their alignments are discarded with options `--min-repeats` and `--max-repeats`, and are stored in the compressed BED file whose name ends with `multimapped`.

```
OPTIONS SUMMARY
===============
BED file: SRR6192867.bed
Reference species: TAIR10_chr_all
Flag conserved mature miRNAs: Yes
Select only sequences with MFEI < -0.6: Yes
Compute thermodynamic stability: Yes
Filter out features given in Arabidopsis_thaliana_CDS.gff
Filter out features given in Arabidopsis_thaliana_otherRNA.gff
Filter out known miRNAs present in Arabidopsis_thaliana_miRBase_wo_chr4-5.gff3
Filter multiple mapped reads: keep reads mapping at 0 to 5 positions
Filter low quality hairpins: Yes

RESULTS SUMMARY
===============
Total number of reads and alignments: 1503938 alignments involving 5682035 reads (500143 unique
reads)
Arabidopsis_thaliana_CDS: removing 36463 alignments involving 212082 reads (23247 unique reads) !"
#$$%&'()%*+,-#./01.23
Arabidopsis_thaliana_otherRNA: removing 23909 alignments involving 246644 reads (8577 unique reads)
!"#$$%&'()%*+4/561$78./01.23
Multiple mapped reads: removing 862356 alignments involving 311833 reads (36328 unique reads) !"
#$$%&'()%*+9:;/<90==6>./01.23
Unclassified elements: 4558 orphan clusters (involving 1702165 reads) + 213 orphan hairpins
(involving 10030 reads)
Known miRNAs: 154 sequence(s) - 1905192 reads ! '"#$$%&'()%*+9<$78?./01.23
Novel miRNAs: 1056 sequence(s) - 3555259 reads ! '"#$$%&'()%*+9<$78?./01.23

Distribution of novel miRNAs according to their quality:
    quality 6: 35 miRNAs
    quality 5: 48 miRNAs
    quality 4: 108 miRNAs
    quality 3: 174 miRNAs
    quality 2: 244 miRNAs
    quality 1: 419 miRNAs
    quality 0: 28 miRNAs
!
```

Fig. 2 Summary file obtained for the SRR6192867 dataset. The first part (*Options Summary*) recalls the parameters used when launching miRkwood. The second part (*Results Summary*) describes the intermediate steps of the workflow. The first line gives the total number of reads present in the initial BED file. The number of unique reads (obtained after merging identical reads) is indicated in brackets. The other lines correspond to the categories mentioned above. The cardinality of all these categories is 5,682,035 (the number of initial reads). In blue, we added the name of the local file that stores the corresponding list of reads. In the two last lines, we see that miRkwood identified a total number of 1210 precursors of miRNAs expressed in the data, 154 of them corresponding to known locus and 1056 being new. These predictions are supported by 5,460,451 reads, which represent 24.8% of all reads

`tar.gz`. From this step, all remaining alignments are gathered in the file ending with `filtered.bed`, which will be the basis for the discovery of novel miRNAs. Using these alignments,

miRkwood searches for clusters which are regions of high concentration of reads at a locus on the genome. Among these clusters, miRkwood selects regions whose genomic sequence is able to fold into a hairpin structure. Other clusters are called *orphan clusters* and are saved in the file suffixed by `orphan_clusters.tar.gz` and are excluded from further analysis. Amongst the hairpin precursors, only good-quality precursors are kept (except when option `--no-filter-bad-hairpins` is activated: all precursors are kept). Clusters that do not belong to such hairpins are called *orphan hairpins*, and are saved in the file whose name ends with *orphan_hairpins.tar.gz*. The clusters corresponding to good quality precursors form the last categories, that of *novel miRNAs*, and are subject to a complete analysis described in Subheadings 3.6 and 3.7 respectively.

3.6 Detailed Results for Known miRNAs

1. All results relative to known miRNAs are available in the folder `results>known_miRNA`. The main file is the html file `results_known_miRNA.html`, that can be visualized with any web browser.
2. The first part of this web page contains a series of links to local files that are present in the same folder, and visible in Fig. 1. There are five formats available.

   Tab-delimited format (CSV) → `result_known_miRNA.csv`: This file contains all information that is available in the result table (*see* below), plus the FASTA sequences and the dot-bracket secondary structures. This tabular format is supported by spreadsheets like Excel.

   FASTA → `result_known_miRNA.fa`: compilation of all pre-miRNA sequences found.

   Dot-bracket format (plain sequence + secondary structure) → `result_known_miRNA.txt`: This is the compilation of all pre-miRNA sequences found, together with the predicted secondary structure, in Vienna dot-bracket notation (Fig. 3).

   GFF format → `result_known_miRNA.gff`: This file contains the list of positions of all pre-miRNA found, in GFF3 format. It follows the guidelines of Ensembl documentation: https://www.ensembl.org/info/website/upload/gff.html.

   ORG format → `result_known_miRNA.org`: This is an equivalent of the summary and individual reports, and contains the full report of the predictions. This file can be easily edited by the user.
3. The second part of the page displays the list of all known miRNAs in a two-way table. Each row corresponds to a

```
>1__3961348-3961464, stemloop structure
UGCAAGGUAACGCGAGAUAUUAGUGCGGUUCAAUCAAAUAGUCGUCCUCUUAACUCAUGGAGAACGGUGUUGUUCGAUUGAGCCGUGCCAAUAUCACGCGGUAAACCAAAAAUGGCA
.....(((..((((.((((((.(((((((((((((.((((((((((.((((........)))).)))).))))))).)))))))))))))).)))))).))))....))).........
```

Fig. 3 Vienna dot-bracket notation. The first line contains a FASTA-like header. The second line contains the nucleic acid sequence. The last line contains the set of associated pairings encoded by brackets and dots. A base pair between bases i and j is represented by a '(' at position i and a ')' at position j. Unpaired bases are represented by dots

chr	position	strand	quality	miRBase name	miRBase ID	reads	miRNA	
							sequence	length
1	28,500-28,706	+	★	ath-MIR838	MI0005394	6	UUUUCUUCUACUUCUUGCACA	21
1	78,927-79,037	-	★★	ath-MIR165a	MI0000199	243761	UCGGACCAGGCUUCAUCCCCC	21
1	234,006-234,159	-	★	ath-MIR2112	MI0010632	6	CUUUAUAUCCGCAUUUGCGCA	21
1	1,653,221-1,653,624	-	★	ath-MIR5640	MI0019213	219	UGAGAGAAGGAAUUAGAUUCA	21

Fig. 4 Result table for known miRNAs (extract). The meaning of each column is as follows. chr is the name of the chromosome, position is the start and end positions of the miRNA precursor as documented in miRBase, +/− gives the strand of the miRNA on the genome, forward (+) or reverse (−). The quality measures the consistency between the distribution of reads along the locus and the annotation provided in miRbase (*see* **Note 4**). It ranges between 0 and 2 stars, and is calculated as follows: the locus contains more than 10 reads: add one star; more than half of the reads intersect either with the guide miRNA or the passenger strand: add one star. miRBase name and miRBase ID are the miRBase name and mirBase identifier. reads is the number of reads included in the locus (highlighted in turquoise blue when greater than 10), miRNA sequence is the sequence of the miRNA, as documented in miRbase, and miRNA length is the length of this sequence

pre-miRNA, and each column to a feature: position of the pre-miRNA, quality score (*see* **Note 4**), miRBase identifiers, number of reads, sequence of the miRNA. *See* Fig. 4 for an example. You can scroll down to all information related to a given prediction by clicking on the row (*see* next item).

4. The last part of the page gives a full report for each individual locus. Compared to the result tables, this report provides several additional pieces of information, such as a direct access to miRBase (miRBase name), the stem-loop structure in Vienna dot-bracket notation, the thermodynamic stability of the structure with the MFE (Minimum Free Energy), the AMFE (Adjusted Minimum Free Energy), and the MFEI. The report also includes a visual representation of all reads matching in the locus, called the reads cloud. An example is provided in Fig. 5.

3.7 Details Results for Novel miRNAs

1. As for known miRNAs, results relative to novel miRNAs are available in a dedicated folder: `results>novel_miRNA`, which has the same architecture as `known_miRNA`. The file `results_novel_miRNA.html` exhibits the same three parts: heading with links to local files, results table and compilation of individual reports.
2. The result table for novel miRNAs looks like the result table for known miRNAs, with some different columns: positions of the pre-miRNA, number of reads, quality of the distribution of

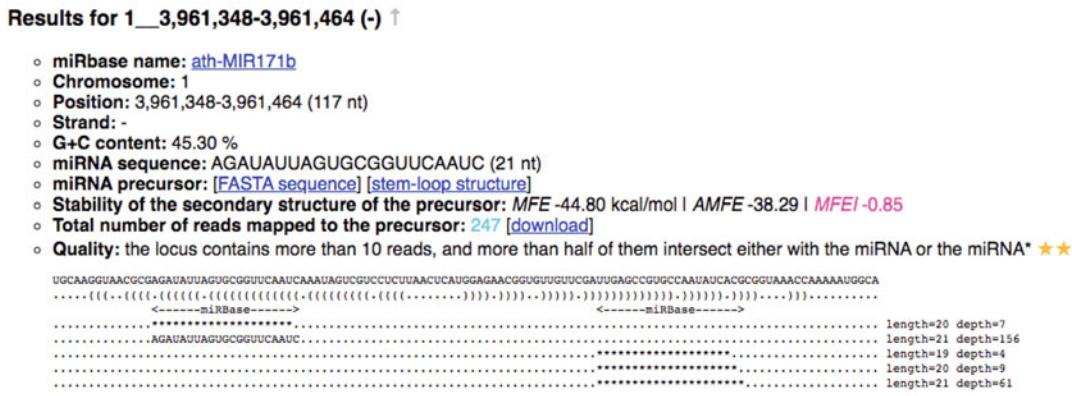


Fig. 5 Example of an individual report for a known miRNA. The first nine lines describe the attributes of the known miRNA: its name, the location of the pre-miRNA on the reference sequence, the GC content of the sequence of the pre-miRNA, the sequence of the miRNA, links to the sequence and the secondary structure of the pre-miRNA, the thermodynamic stability of the pre-miRNA measured by its MFE, AMFE, and MFEI and the total number of reads matching the pre-miRNA. The Quality is the score computed for this prediction, such as defined in the caption of Fig. 4. The graphics at the bottom of the report is the *reads cloud*. In this representation, the first line is the sequence of the precursor, the second line is its secondary structure. Each ********** string represents a unique read, whose length and depth are reported at the end of the dotted line. <-----miRBase------> indicates the positions of the miRNA(s) referenced in miRBase. The read corresponding to the miRNA is written in full letters. miRkwood finds that the miRNA ath-MIR171b is expressed in the data. A total of 237 reads were mapped on this precursor. The guide and the passenger miRNAs are supported by 156 and 61 reads, respectively. The other reads could correspond to guide and passenger miRNA variants. The grey arrow next to the title enables users to go back to the results table

reads, miRNA sequence, existence of miRBase alignments. *See* Fig. 6 for the full details.

3. For the individual reports, some of the information is the same as for known RNAs (described in Subheading 3.6): sequence of the pre-miRNA, G+C percent, stem-loop structure, MFE, MFEI, AMFE, number of reads, read clouds, read length distribution. The fields below are specific to novel miRNAs. *See* Fig. 7.

   Candidates with the same miRNA: List of other miRkwood predictions that involve the same miRNA sequence. miRkwood checks if the miRNA (when it exists) could possibly originate from a duplication. For that, it reports whether the sequence is present elsewhere in the genome, and whether this alternative location corresponds to another miRNA precursor.

   Alternative candidates: This is the set of stem-loop sequences that overlap the current pre-miRNA prediction. The choice between several alternative overlapping candidate pre-miRNAs is made according to the best MFEI.

chr	position	strand	reads	reads distribution	mfei	shuffles	miRNA			
							sequence	length	weight	alignment
5	3,867,184-3,867,333	+	1442	★★★	-1.03	0.0	UGACAGAAGAGAGUGAGCAC	20	270.6	✓✓
5	4,141,269-4,141,416	-	24		-0.75	0.005	AAAUUUGGUUUUCAUUUAGGAGGC	24	12	✓
5	4,691,014-4,691,148	+	118	★★★	-0.88	0.0	UGUGUUCUCAGGUCACCCCUG	21	54.5	✓✓
5	4,694,674-4,694,830	+	118	★★★	-0.79	0.0	UGUGUUCUCAGGUCACCCCUG	21	54.5	✓✓

Fig. 6 Result table for novel miRNAs (extract). The columns *chr*, *position*, and *strand* give the positions of the putative miRNA precursor on the reference sequence (in 1-based notation, as in the GFF format). *reads* is the number of reads included in the locus. It is highlighted in turquoise blue when it has either at least 10 reads mapping to each arm, or at least 100 reads mapping in total. This criterion is inspired from miRBase definition for high-confidence miRNAs (http:/www.mirbase.org/blog/). *reads distribution* is a score ranging from 0 to 3-stars that allows to qualify the pattern of reads mapping to a putative microRNA precursor (*see* **Note 4**). It aims at determining if the distribution of reads presents a typical 2-peaks profile, corresponding to the guide and the passenger miRNA, respectively. It is based on three criteria, each adding a star
• *Number of reads*: The locus has either at least 10 reads mapping to each arm, or at least 100 reads mapping in total
• *Precision of the precursor processing:* At least 75% of reads start in a window [−3,+3] centered around the start position of the miRNA, or [−5,+5] around the pairing position on the opposite arm of the stem-loop
• *Presence of the miRNA duplex:* There is at least one read in the window [-5,+5] around the pairing position on the strand of the passenger miRNA
MFEI has the same definition as for known miRNAs. miRNA sequence is the sequence of the most common read, provided that its frequency is at least 33%. Otherwise, we do not define any mature miRNA sequence for the locus. miRNA length is the length of this read. miRNA weight is the depth of this read divided by the number of possible alignments of this read in the genome. The column alignment appears only when the option -- align has been invoked. It is marked ✓ when an alignment between the pre-miRNA sequence and plant mature miRNA database of miRBase is found. It is double checked ☑ ☑ if at least 40% of reads overlap with the miRBase sequence. The alignments are visible in the individual report, presented in Fig. 7

Optimal MFE secondary structure: If the stem-loop structure is not the MFE structure, we also provide a link to download the MFE structure.

Stability of the miRNA duplex. Previously, it was reported that the guide miRNA and the passenger miRNA form a duplex with two nucleotide overhangs, and base-pairing between the miRNA and the other arm is extensive [12]. We formalize it with the usage of miRdup, that assesses the stability of the duplex formed by the guide and the passenger miRNAs by machine learning (with random forests) [4]. Here, it was trained on miRBase Viridiplantae V20 with default parameters.

Conserved mature miRNA: For novel miRNAs, the precursor sequence is aligned with mature miRNAs of miRbase to search for homology. Alignments with at most three errors (mismatch, deletion or insertion) against the full-length mature miRNA and that occur in one of the two arms of the stem-loop are selected.

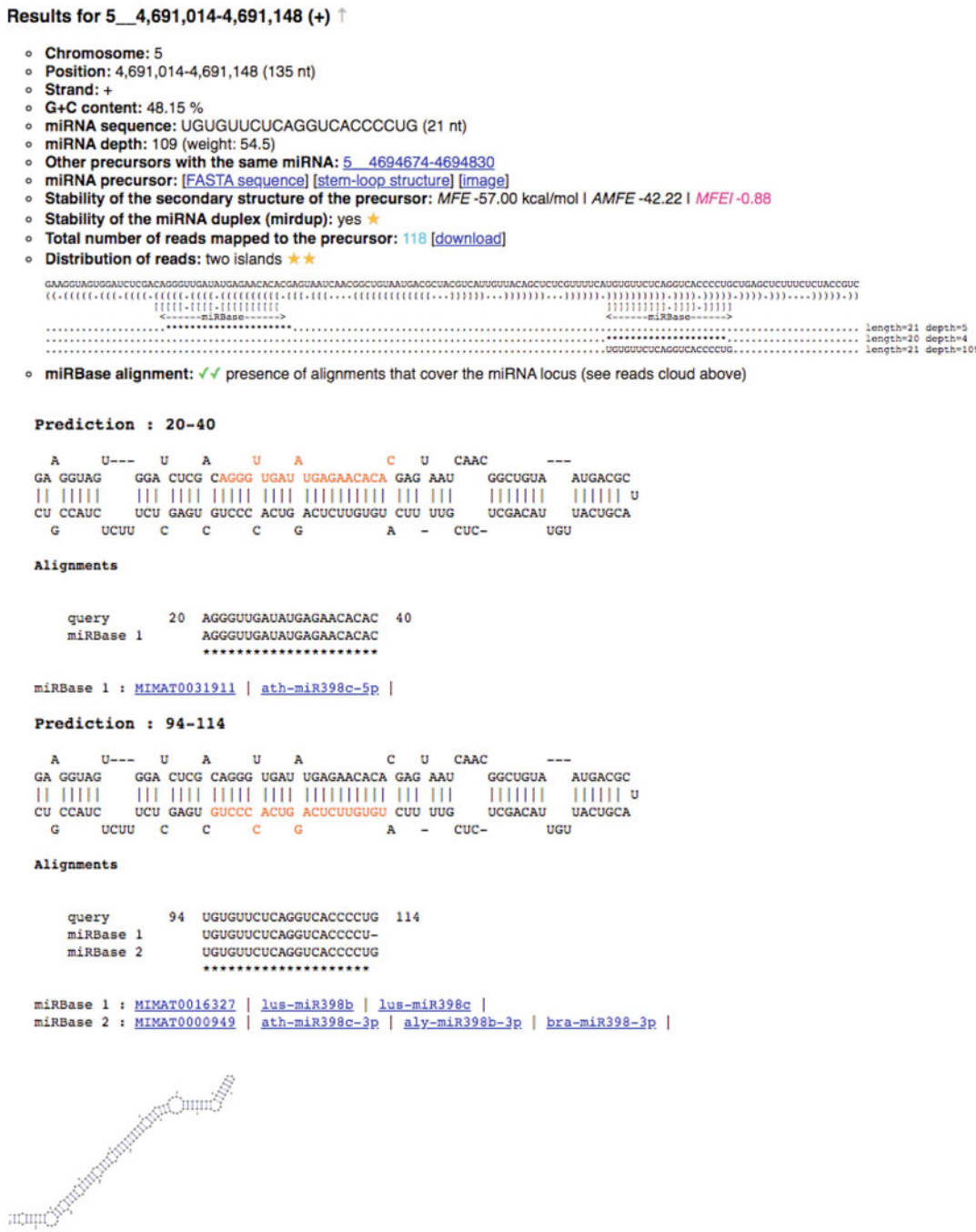


Fig. 7 Example of an individual report for a novel miRNA. In this example, miRkwood found a putative new pre-miRNA located on chromosome 5, at positions 4,691,014–4,691,148 on strand + with length 135nt. We recall that in this example miRNA/pre-miRNA miRbase annotations from chromosomes 5 and 6 were removed when running miRkwood to ensure that novel miRNAs could be predicted. The prediction indeed overlaps with the pre-miRNA ath-MIR398b (chr5: 4,691,022–4,691,137 [+]), even if miRkwood predicted a slightly longer precursor. The miRNA determined by miRkwood is 21nt long. Its sequence is UGUGUUCUCAGGUCACCCCUG and it is represented by 109 reads, with weight 54.5 = 109/2, which means that the miRNA matches another locus in the genome. This locus is on the same chromosome, at positions 4,694,674–4,694,830 (which

Finally, we provide an ASCII representation of the putative miRNA within the stem-loop precursor.

3.8 Additional Folders and Files

All alignments, images, read clouds, and sequences available in the result web pages are also directly accessible in dedicated folders.

1. Alignments: this folder contains all miRbase alignments found for all novel precursors. There is one file per precursor.
2. Images: this folder contains all secondary structures images generated with Varna (when the option `--varna` is activated). There is one file per precursor.
3. Read_clouds: this folder contains all read clouds for all precursors. There is one file per precursor.
4. Sequences: this folder contains the set of all sequences for all precursors: primary structure in FASTA format, secondary structure of the hairpin, secondary structure of the optima structure, secondary structure of alternative precursors in Vienna bracket-dot format. There is one file per sequence.

3.9 YAML Export

We provide an YAML output, that is specifically well-suited for automatic parsing of the results. YAML is a markup format, similar yet more compact than XML. The basic_known_candidates.yml and basic_candidates.yml files contain the same information as the html results files for known and novel predictions. Detailed information on each candidate is stored in an individual file in the YML folder.

4 Notes

1. In this chapter, we chose to focus on the stand-alone version on miRkwood. It is also possible to use miRkwood through its web server: https://bioinfo.cristal.univ-lille.fr/mirkwood/smallRNAseq. This service does not require any prior local

Fig. 7 (continued) correspond to ath-MIR398c). This canonical miRNA is accompanied by a 20 nt read of depth 4, which could be either an isoMir or an incomplete read. The prediction is also supported by 5 reads corresponding to the passenger miRNA. The folding of the pre-miRNA is thermodynamically stable (MFEI $= -0.88$), and the duplex formed by the miRNA and its passenger is validated by miRdup. We also observe in the reads cloud that the pre-miRNA contains two matches with miRNAs of miRBase, at positions 20–40 and 94–114 respectively. The positions of these matches are consistent with the reads distribution: they correspond exactly to the positions of the putative guide miRNA and passenger strand predicted by miRkwood. These positions are indeed identical to those of ath-miR398b-5P and ath-miR398b-3P. The miRBase identifiers with miRbase links under the alignments inform us that this miRNA is also known in *Arachis hypogaea*, *Linum usitatissimum*, *Arabidopsis lyrata*, and *Brassica rapa*, indicating that it is conserved at a large phylogenetic scale. The image at the bottom is the Varna visualization (accessed when clicking on the image link of the miRNA precursor)

installation and offers all previously described facilities. It is currently available for 14 genomes: *Arabidopsis thaliana*, *Arabidopsis lyrata*, *Brassica napus*, *Brassica rapa*, *Glycine max*, *Lotus japonicus*, *Medicago truncatula*, *Oryza sativa*, *Phaseolus vulgaris*, *Physcomitrella patens*, *Populus trichocarpa*, *Solanum lycopersicum*, *Sorghum bicolor*, *Vitis vinifera*.

2. miRwood also comes with an ab initio option, without sRNA-Seq data. In this mode, miRkwood scans the genomic sequences to search for pre-miRNA loci based on the shape and the stability of the secondary structure. This version is available on the above-mentioned website (https://bioinfo.cristal.univ-lille.fr/mirkwood/abinitio) and included in the Docker container. It uses BlastX (for coding regions predictions) [13], tRNAscan-SE (for transfer RNAs prediction) [14] and RNAmmer (for ribosomal RNAs prediction) [15].
3. The software allows to control the number of repeats for each read, that is the minimum and maximum number of alignments on the reference sequence for a given read (options min-repeats and max-repeats). Multiple mapped reads could correspond to siRNAs involved in the RNA-directed DNA methylation pathway (RdDM) that suppresses the activity of transposable elements, which are present in multiple copies in plant genomes. As a consequence, we propose to users to filter multiple mapped reads ($\geq$5 distinct locations). We offer the possibility to tune this option, which can be useful for particular contexts. For example, in the case of genomes originating from whole genome duplication events, the maximum number of alignments that are tolerated can be increased.
4. The quality score (ranging from 0 to 3 and from 0 to 6 for known and novel miRNAs, respectively) computed by miRkwood for each prediction is an important feature of the software that helps to classify the results, especially for novel miRNAs. We designed miRkwood so that it can predict miRNAs with high sensitivity, allowing users to filter the results according to whether they want to limit the number of false positives and/or identify new miRNAs. Indeed, we showed that miRkwood set up with a quality score ≥ 5 mainly detects canonical miRNAs and limits the number of false positives. Decreasing the threshold score to 4 or 3 may lead to an increase in the proportion of false positives, but potentially offers in return the detection of novel miRNAs and long-miRNAs [1]. Predictions with score 0, 1, or 2 are likely to be false-positive predictions.

References

1. Guigon I, Legrand S, Berthelot J-F, Bini S, Lanselle D, Benmounah M, Touzet H (2019) miRkwood: a tool for the reliable identification of microRNAs in plant genomes. BMC Genomics 20:532. https://doi.org/10.1186/s12864-019-5913-9
2. https://bedtools.readthedocs.io/en/latest/content/bedtools-suite.html
3. Lorenz R, Bernhart SH, Höner zu Siederdissen C, Tafer H, Flamm C, Stadler PF, Hofacker IL (2011) ViennaRNA Package 2.0. Algorithms Mol Biol 6:26. https://doi.org/10.1186/1748-7188-6-26
4. Leclercq M, Diallo AB, Blanchette M (2013) Computational prediction of the localization of microRNAs within their pre-miRNA. Nucleic Acids Res 41:7200–7211. https://doi.org/10.1093/nar/gkt466
5. Darty K, Denise A, Ponty Y (2009) VARNA: interactive drawing and editing of the RNA secondary structure. Bioinformatics 25:1974–1975. https://doi.org/10.1093/bioinformatics/btp250
6. Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.journal 17:10–12. https://doi.org/10.14806/ej.17.1.200
7. Schmieder R, Edwards R (2011) Quality control and preprocessing of metagenomic datasets. Bioinformatics 27:863–864. https://doi.org/10.1093/bioinformatics/btr026
8. Langmead B (2010) Aligning short sequencing reads with Bowtie. Curr Protoc Bioinformatics Chapter 11:Unit-11.7. https://doi.org/10.1002/0471250953.bi1107s32
9. Turner DH, Mathews DH (2010) NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. Nucleic Acids Res 38:D280–D282. https://doi.org/10.1093/nar/gkp892
10. Bonnet E, Wuyts J, Rouzé P, Van de Peer Y (2004) Evidence that microRNA precursors, unlike other non-coding RNAs, have lower folding free energies than random sequences. Bioinformatics 20:2911–2917. https://doi.org/10.1093/bioinformatics/bth374
11. Polydore S, Axtell MJ (2018) Analysis of RDR1/RDR2/RDR6-independent small RNAs in Arabidopsis thaliana improves MIRNA annotations and reveals unexplained types of short interfering RNA loci. Plant J Cell Mol Biol 94:1051–1063. https://doi.org/10.1111/tpj.13919
12. Kurihara Y, Watanabe Y (2004) Arabidopsis micro-RNA biogenesis through Dicer-like 1 protein functions. Proc Natl Acad Sci U S A 101:12753–12758. https://doi.org/10.1073/pnas.0403115101
13. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990) Basic local alignment search tool. J Mol Biol 215:403–410. https://doi.org/10.1016/S0022-2836(05)80360-2
14. Chan PP, Lowe TM (2019) tRNAscan-SE: searching for tRNA genes in genomic sequences. Methods Mol Biol 1962:1–14. https://doi.org/10.1007/978-1-4939-9173-0_1
15. Lagesen K, Hallin P, Rødland EA, Staerfeldt H-H, Rognes T, Ussery DW (2007) RNAmmer: consistent and rapid annotation of ribosomal RNA genes. Nucleic Acids Res 35:3100–3108. https://doi.org/10.1093/nar/gkm160

Chapter 9

Pangenome Analysis of Plant Transcripts and Coding Sequences

Bruno Contreras-Moreira, Álvaro Rodríguez del Río, Carlos P. Cantalapiedra, Rubén Sancho, and Pablo Vinuesa

Abstract

The pangenome of a species is the sum of the genomes of its individuals. As coding sequences often represent only a small fraction of each genome, analyzing the pangene set can be a cost-effective strategy for plants with large genomes or highly heterozygous species. Here, we describe a step-by-step protocol to analyze plant pangene sets with the software GET_HOMOLOGUES-EST. After a short introduction, where the main concepts are illustrated, the remaining sections cover the installation and typical operations required to analyze and annotate pantranscriptomes and gene sets of plants. The recipes include instructions on how to call core and accessory genes, how to compute a presence–absence pangenome matrix, and how to identify and analyze private genes, present only in some genotypes. Downstream phylogenetic analyses are also discussed.

Key words Pangenome, Pangene set, Crops, Model plants, Wild plants, Polyploids, Scripting

1 Introduction

1.1 Background

A pangenome can be defined as the sum of the core genome, shared by all individuals of a species, plus the dispensable genome, which includes partially shared and population-specific genes [1]. Among plants, previous works have compared ecotypes of model species and cultivars of crops such as maize, barley, soybean, or rice, revealing that dispensable genes play important roles in evolution, and in the complex interplay between plants and the environment (reviewed at [2]). Moreover, accounting for accessory genomic features improves the association of genotypes to phenotypes beyond SNPs called on a single reference genome [3, 4]. For these reasons we prefer to rename dispensable genes as "accessory" or "shell" genes, terms borrowed from microbiology [5]. In summary, pangenomes are the new reference sequences for plant genomic studies [6].

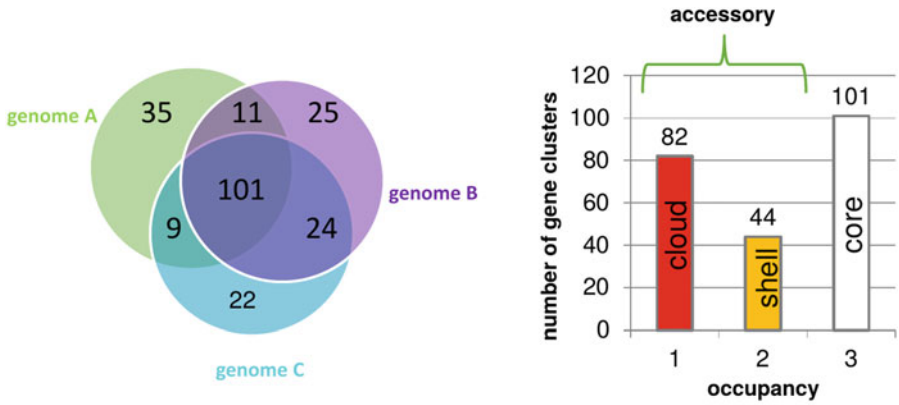
Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_9,

The data structures used to represent pangenomes are evolving in parallel with the sequencing technologies, and the state of the art are the so called pangenome graphs, which significantly improve variant calling by read mapping [4, 7] and have multiple implementations (see for instance [8]). However, there are currently no community accepted solutions for visualizing or setting coordinate systems in genome graphs. More importantly, the input genome sequences used to construct a graph need to be highly contiguous. This contrasts with most plant genome assemblies found in the public archives, which are often fragmented, particularly for species with large, highly repetitive, heterozygous and polyploid genomes. For these reasons, approaches that do not require a fully assembled reference genome are of interest for plant breeding and ecology. Some of these strategies reduce the natural complexity of genomes by computing the frequency of nucleotide words and looking for enriched subsets of sequences [9, 10]. Other strategies, like the one presented in this protocol around GET_HOMOLOGUES-EST [11], take transcripts or coding sequences (CDS) as the genomic unit of interest (see examples at [12, 13]). Therefore, a more appropriate name for this approach would be pangene set analysis.

Compared to whole genome sequencing (WGS) and assembling, pangene analyses have the advantage of sampling only the expressed part of the genome, the transcriptome, which is only a fraction of the complete genome. Recent technological advances, such as single-molecule long-read sequencing, are producing transcripts with unprecedented accuracy [14], even without a reference genome [15]. Their main disadvantage is that genetic variation in nonexpressed sequences cannot be sampled. Nevertheless, the definition of pangene sets is a necessary step for pangenome projects of crops and plants, as a way to deduce consistent gene collections with uniform nomenclature across genotypes.

1.2 Pangenome History and Concepts

In 2002, Welch and colleagues [16] compared the genome sequences of three strains of the bacteria *Escherichia coli*, two of them pathogens, and found shared genes, which mostly conserve their positions in the genome, and also accessory genes, which are encoded only in some strains. Three years later, Tettelin and collaborators [1] analyzed 8 strains of a gram-positive bacteria and, for the first time, defined the pangenome as a core genome shared by all strains, plus a dispensable genome consisting of partially shared and strain-specific genes. In addition, a mathematical model suggested that the pangenome of *Streptococcus agalactiae* might be very large and open, as novel genes would continue to be identified in newly sequenced strains. In 2007, Morgante and coworkers [17] took these ideas and proposed a role for transposable elements (TEs) in generating the dispensable genomic components that differentiate maize inbred lines B73 and Mo17. A few years later they concluded that these components might not be dispensable



genome A	genome B	genome C	occupancy	class
gene 1A	gene 41B	-	2	shell
gene 2A	gene 42B	genes 114C, 120C	3 (inparalogues)	core
gene 3A	-	-	1	cloud
gene 4A	gene 112B	gene 122C	3 (single-copy)	core

Fig. 1 Pangene set of three genomes (A, B and C). In this example there are 101 core loci and 44 (11 + 9 + 24) shell loci are found in two accessions. Moreover, 82 (35 + 25 + 22) cloud loci are annotated only in one genome. The pangene set size is 82 + 44 + 101 = 227 loci. The table below shows four example clusters and their occupancy, computed using the rules in Table 1

Table 1
Definition of occupancy-based classes used by the software GET_HOMOLOGUES-EST

Class or compartment	Definition
Core	Genes contained in all considered genomes/taxa
Soft-core	Genes contained in 95% of the considered genomes/taxa, as in [43]
Cloud	Genes present only in a few genomes/taxa, generally ≤2. The cutoff is defined as the class next to the most populated noncore cluster class
Shell	Remaining genes, present in several genomes/taxa

after all [18]. In order to make this classification problem more tractable, dispensability scores are now being computed [19].

Figure 1 summarizes the computational analysis of the pangene set of three toy genomes with GET_HOMOLOGUES-EST. Note that occupancy is defined as the number of genomes/cultivars/ecotypes present in a sequence cluster and in this example takes values from 1 to 3 (see also definitions in Table 1). Core loci are found in all three genomes and hence have occupancy = 3 in the example. Accessory loci are allocated to shell and cloud compartments. Although not shown in the figure, it is often convenient to

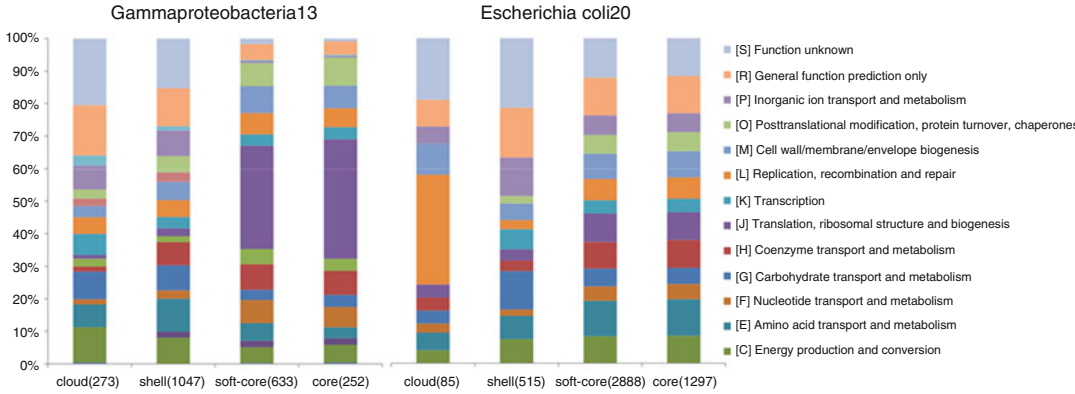


Fig. 2 Functional annotations (COGs) of protein-coding genes classified in occupancy classes in two bacterial sets

define a fourth class, the soft-core, which is a relaxed core that tolerates assembly and annotation errors.

In addition to occupancy differences, previous benchmarks have shown that occupancy-based classes differ in their functional annotations. The examples in Fig. 2 highlight the different functions of core and accessory genes in two bacterial clades; note that similar observations have been made in plants [11, 12].

2 Materials

The GET_HOMOLOGUES software was originally designed for the analysis of bacterial genomes, and has been described in [20, 21]. That software was then adapted to the study of intraspecific eukaryotic pangene sets, as described in [11], taking the GET_HOMOLOGUES-EST name. Its source code and documentation can be found at https://github.com/eead-csic-compbio/get_homologues. Table 2 summarizes the main differences between the two flavors of the software. In this protocol we will use GET_HOMOLOGUES-EST (*see* **Note 1**).

2.1 Installation and Up-To-Date Documentation

GET_HOMOLOGUES-EST is an open source software package, written in Perl, R, and bash, available for Linux and MacOS systems. A manual is available at http://eead-csic-compbio.github.io/get_homologues/manual-est (*see* **Note 2**). Some of the scripts included in the package require additional software dependencies, as described in detail in the manual.

There are two ways to install GET_HOMOLOGUES-EST, as a bundled release or by cloning the GitHub repository. Both options, described below, will need the installation of a few dependencies in order to follow this tutorial. Assuming we are in Ubuntu, these can be installed as follows.

Table 2
Summary of features and differences of GET_HOMOLOGUES and GET_HOMOLOGUES-EST

Version	Primary input	Primary engine	Align coverage	COGS	Isoforms filtered
GET_HOMS	Peptides	BLASTP / DIAMOND	Query sequence	Yes	No
GET_HOMS-EST	Nucleotides	BLASTN	Shortest sequence	No	Yes

Note that GET_HOMOLOGUES can also cluster user-selected nucleotide features in GenBank files, and in this case BLASTN is used as well

```
$ sudo apt-get -y install r-base r-base-dev parallel
$ sudo cpan -i Inline::C Inline::CPP
```

Instead, the Docker container described in Subheading 2.1.3 is self-contained and does not require any extra installation steps.

2.1.1 *Bundled Release*

The simplest way to get the current release of GET_HOMOLOGUES is to check https://github.com/eead-csic-compbio/get_homologues/releases, download it and extract it in your local filesystem. Let us assume we have a dedicated folder called *soft*.

```
$ cd soft

# make sure you choose the right version
$ wget -c https://github.com/eead-csic-compbio/get_homologues/releases/download/v3.4.2/get_homologues-x86_64-20210305.tgz
# or
# wget -c https://github.com/eead-csic-compbio/get_homologues/releases/download/v3.4.2/get_homologues-macosx-20210305.tgz

$ tar xvfz get_homologues-x86_64-20210305.tgz
```

Releases include scripts (Perl, R and bash), documentation, sample data and the required binary dependencies, which are listed in Table 3 and discussed in the manual.

2.1.2 *GitHub Clone/Pull*

A more sustainable way of obtaining the software is to use the software *git*. For this you might need to install *git* in your system. Note also that the *git* protocol requires having network port 9418 open, which might be blocked by your firewall. This method does not include the binary dependencies, which must downloaded during installation.

Table 3
Dependencies of GET_HOMOLOGUES-EST

Software	Source	Citation
mcl v14–137	http://micans.org/mcl	[44]
NCBI Blast-2.8.1+	http://blast.ncbi.nlm.nih.gov	[45]
BioPerl v1.5.2	http://www.bioperl.org	[46]
HMMER 3.1b2	http://hmmer.org	
Pfam	http://pfam.xfam.org	[24]
PHYLIP 3.695	http://evolution.genetics.washington.edu/phylip	
Transdecoder r20140704	http://transdecoder.sf.net	[47]
MVIEW 1.60.1	https://github.com/desmid/mview	[48]
diamond 0.8.25	https://github.com/bbuchfink/diamond	[49]

```
$ cd soft
$ git clone https://github.com/eead-csic-compbio/get_homologues.git

# this creates a folder called get_homologues which does not
# include the binary dependencies; these must be downloaded:

$ cd get_homologues
$ perl install.pl no_databases
```

This approach makes future updates very simple, after moving to the git repository.

```
$ cd soft/get_homologues
$ git pull
```

2.1.3 Environment and Optional Data (Pfam and SwissProt)

Whatever way we installed the software we can now define an environment variable pointing to our install location.

```
# set GET_HOMOLOGUES path, for example:
$ export GETHOMS=~/soft/get_homologues
# or
# export GETHOMS=~/soft/get_homologues-x86_64-20210305
```

In order to annotate protein domains and to translate open reading frames (ORFs) a couple of databases must be downloaded and formatted. These are Pfam [24] and SwissProt [25], which you can download and format as follows in the terminal.

```
$ cd $GETHOMS
$ perl install.pl
```

You should be able to control the installation process by typing Y or N in the terminal. This script will also tell you of any missing dependencies.

2.1.4 Docker Container

A way to use GET_HOMOLOGUES-EST with all dependencies preinstalled is the Docker image available at https://hub.docker.com/r/csicunam/get_homologues. This container also includes GET_PHYLOMARKERS (*see* Subheading 3.3.6).

2.1.5 High Performance Cluster (HPC) Configuration

In order to prepare your installation to run on a computer cluster please follow the instructions in section "Optional Software Dependencies" of the manual. Three job managers are currently supported: gridengine, LSF and Slurm. The default configuration is for gridengine, but this can be changed by creating a file named "HPC.conf" and setting appropriate values and paths for your HPC cluster (*see* **Note 3**). This file should be placed at $GETHOMS. The sample configuration file looks like this.

```
$ cat sample.HPC.conf
# cluster/farm configuration file, edit as needed (use spaces
or tabs)
# comment lines start with #
# PATH might be empty or set to a path/ ending with '/'
# QARGS might be empty or contain specific queue name,
resources, etc
#
# example configuration for LSF
#PATH   /lsf/10.1/linux3.10-glibc2.17-x86_64/bin/
TYPE    lsf
SUBEXE  bsub
CHKEXE  bjobs
DELEXE  bkill
ERROR   EXIT
#
# example configuration for slurm
TYPE    slurm
SUBEXE  sbatch
CHKEXE  squeue
DELEXE  scancel
ERROR   F
```

Running GET_HOMOLOGUES-EST on a computer cluster distributes batches of BLASTN/HMMER jobs and the actual clustering tasks (isoforms, orthologues, inparalogues), which is recommended if you plan to analyze a large number of sequence sets.

3 Methods

3.1 Overview of the Pipeline and the get_homologues-est.pl *Script*

Unlike its predecessor, which was designed for the analysis of genes within fully sequenced genomes, GET_HOMOLOGUES-EST has been adapted to the large size of plant genomic data sets, and adds new features to adequately handle redundant and fragmented transcript sequences, as those usually obtained from transcript profiling experiments, as well as incomplete/fragmented gene models from WGS assemblies. Sequence clusters can be produced using the bidirectional best-hit (BDBH) or the OrthoMCL [26] (OMCL) algorithms. The granularity of the clusters can be tuned by a configurable filtering strategy based on a combination of BLAST pairwise alignment parameters and, optionally, hmmscan-based scanning of Pfam domain composition of the proteins encoded in each cluster. By default redundant sequences are removed from input sets if they overlap a longer identical sequence over a length $\geq$40 or when they are completely matched. This is to prevent BLAST results being biased by transcript isoforms [27]. If your input are annotated CDS sequences this behavior can be disabled with option −i 0 .

As summarized in Fig. 3, the main script *get_homologues-est.pl* can produce different outputs.

- Sequence clusters based on nucleotide similarity, which provide higher resolution than peptide comparisons.
- Presence-absence pangenome matrices in different formats, which can be easily transformed and exported for different applications.
- Simulations on the growth of the pangenome at the gene level using different fits [1, 28] and mixture models [29].
- Average nucleotide sequence identity (ANI) matrices, which contain average sequence identity values among pairs of genomes, computed from clustered sequences. These values are derived from BLASTN alignments.
- Matrices of percentage of conserved sequence clusters (POCS), which contain the % of clusters that contain sequence from every two species being compared [30].

As assembled transcripts are often incomplete, and gene models might contain errors, by default GET_HOMOLOGUES-EST computes alignment coverage with respect to the shortest sequence. This adds robustness against split genes, partial genes, and genes or transcripts with retained introns [11].

3.2 Other Scripts

A few scripts are bundled with GET_HOMOLOGUES-EST to assist in the interpretation of results. We will use some of them in this protocol.

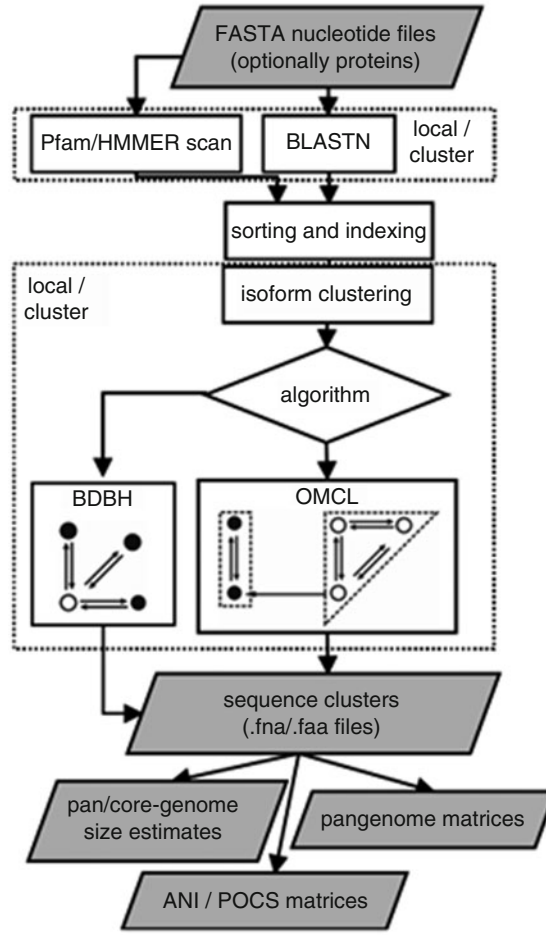


Fig. 3 Features of GET_HOMOLOGUES-EST. Flowchart of the main tasks and deliverables. BLASTN and optional Pfam scans, as well as BDBH and OMCL clustering, can be run on a local computer, preferably multicore, or over an HPC cluster. Resulting clusters are post-processed to produce pangenome or average nucleotide identity matrices, as well as to estimate pan-, soft-core-, and core-genomes. Note that both clustering algorithms can be fine-tuned by customizing an array of parameters, of which alignment coverage (−C) and same Pfam domain composition (−D) are perhaps the most important. While OMCL is adequate for most applications, BDBH is faster for the calculation of core sequences within large datasets

- *compare_clusters.pl* primarily calculates the intersection between cluster sets, which can be used to select clusters supported by different algorithms or settings. This script can also produce pangenome matrices and Venn diagrams.
- *parse_pangenome_matrix.pl* can be used to analyze pangenome sets, in order to find transcripts/genes present in a group A of strains which are absent in set B. This script can also be used for

calculating and plotting cloud, shell, and core genome compartments with mixture models.

- *plot_pancore_matrix.pl* plots pan/soft/core-genome sampling results and fits regression curves with help from R functions.
- *check_BDBHs.pl* can be used, after a previous *get_homologues-est.pl* run, to find out the bidirectional best hits of a sequence identifier chosen by the user. It can also retrieve its Pfam annotations.
- *annotate_cluster.pl* produces a multiple alignment view of the supporting local BLAST alignments of sequences in a cluster. It can also annotate Pfam domains and find private sequence variants to an arbitrary group of sequences.
- *plot_matrix_heatmap.sh* calculates ordered heatmaps with attached row and column dendrograms from tab-separated numeric matrices, which can be presence/absence pangenomic matrices or identity matrices as those produced by get_homologues-est with flag −A. Requires R packages *ape* [31], *dendextend*, *factoextra*, and *gplots* (*see* **Note 4**).
- *pfam_enrich.pl* calculates the enrichment of a set of sequence clusters in terms of Pfam domains, by using Fisher's exact test.

Apart from these, scripts *transcripts2cds.pl* and *transcripts2cdsCPP.pl* are also bundled to assist in the analysis of transcripts. They can be used to annotate potential ORFs contained within raw transcripts, which might be truncated or contain retained introns. The CPP version is faster but requires some optional Perl dependencies (which you should have installed earlier). These scripts use TransDecoder and BLASTX (*see* Table 3) to scan protein sequences in SWISSPROT (*see* **Note 5**).

3.3 Protocol for Plant Transcripts and CDS Sequences

In this section we present a step-by-step protocol for the analysis of *Hordeum vulgare* WGS-based cDNA sequences (from reference cultivar *Morex*) and de novo assembled transcriptomes. These sequence sets were first used in [11]. For convenience, these barley sequences can be obtained as explained below from files included in the *test_barley* folder, which should be bundled with your copy of GET_HOMOLOGUES (*see* **Note 6**).

3.3.1 Preparing Input Sequences, Outgroups and Extracting CDS from Transcripts

The main script reads in sequences in FASTA format, which might be GZIP- or BZIP2-compressed, from a folder containing at least two files with extension .fna, one per cultivar/ecotype. You should consider adding sequences from one or more outgroup taxa if you plan to run downstream phylogenomic analyses with the resulting clusters (*see* **Note 7**).

By default only sequences with length ≥20 bases are considered (*see* **Note 8**).

Files with .fna extension allow clustering any kind of nucleotide sequences, such as conserved intergenic regions or transposons. However, if you are only interested in the analysis of CDS sequences, .fna files might optionally have twin .faa files with translated amino acid sequences in the same order. Note that you will need .faa files to run Pfam-based analyses and the phylogenomics recipes of GET_PHYLOMARKERS [33] (*see* Subheading 3.3.6).

Peptide files can be downloaded from resources such as Ensembl Plants [34] or Phytozome [35], or deduced with script *transcripts2cds.pl*, as in the examples of this protocol.

```
$ $GETHOMS/transcripts2cds.pl -n 3 $GETHOMS/sample_tran-
scripts.fna

# ./transcripts2cds.pl -p 0 -m -d $GETHOMS/db/uniprot_sprot.
fasta -E 1e-05 -l 50 -g 1 -n 3 -X 0
# input files(s):
# sample_transcripts.fna

## processing file sample_transcripts.fna ...
# parsing re-used transdecoder output (_sample_transcripts.
fna_l50.transdecoder.cds.gz) ...
# running blastx...
# parsing blastx output (_sample_transcripts.fna_E1e-05.
blastx.gz) ...
# calculating consensus sequences ...
# input transcripts = 9
# transcripts with ORFs = 7
# transcripts with no ORFs = 2
# output files: sample_transcripts.fna_l50_E1e-05.transcript.
fna , sample_transcripts.fna_l50_E1e-05.cds.fna ,
sample_transcripts.fna_l50_E1e-05.cds.faa , sample_tran-
scripts.fna_l50_E1e-05.noORF.fna
```

If some of your input files contain high-quality sequences, such as full-length cDNA sequences, you can add the tag "flcdna" to their file names. By doing this you make sure that the length of these sequences is used downstream in order to estimate coverage alignment with high confidence. For example, if we wanted to add cultivar *Haruna Nijo* sequences [32] to our analysis, we could put them in a file named HarunaNijo.flcdna.fna.

Other analyses you might want to carry out on your input sequences include computing their coding potential or their gene completeness (*see* **Note 9**).

Let us now get all the barley sequences and extract CDS from them, which takes several hours.

```
$ cd ${GETHOMS}/test_barley
$ cd seqs

# download all transcriptomes
$ wget -c -i wgetlist.txt

# Extract CDS sequences (run transcripts2cds.pl for each
transcriptome).
# Choose cdsCPP.sh if dependency Inline::CPP is available in
your system
# the script will use 20 CPU cores, please adapt it to your
system
./cds.sh

# clean and compress (we want only in the CDS files)
$ rm -f _* *noORF* *transcript* # might be useful, but not used
$ gzip *diamond*

# put cds sequences aside
$ mv *cds.f*gz ../cds && cd ..
```

Alternatively, it is possible to get precomputed CDS sequences (170 MB) ready to go.

```
$ cd ${GETHOMS}/test_barley
$  wget  http://floresta.eead.csic.es/plant-pan-genomes/bar-
ley_transcripts/cds.tgz .
$ tar xvfz cds.tgz

# Make sure files with lists of accessions are in place
$ ls cds/*list

# Either way, make sure files with lists of accessions are in
place; these will be used later on to compare input subsets:

$ ls cds/*list

#cds/leaf.list
#cds/ref.list
#...
```

3.3.2 Clustering Sequences

In this step, we will use the main script *get_homologues-est.pl*. The only required option is −d, which indicates an input folder or directory. It is important to remark that in principle only files with extensions .fna / .fa / .fasta and optionally .faa are considered when parsing the −d directory. The use of an input folder allows for new files to be added there in the future, reducing the computing

required for updated analyses. For instance, if a user does a first analysis with 5 input genomes today, it is possible to check how the resulting clusters would change when adding 10 new genomes tomorrow, by copying the new .fna input files to the preexisting −d folder, so that all previous BLASTN searches are reused.

All remaining flags are options that can modify the default behavior of the program, which is to use the BDBH algorithm in order to compile core clusters of DNA sequences based on BLASTN megablast searches requiring 95% sequence identity (−S 95) and 75% alignment coverage (−C 75). The available clustering algorithms are as follows.

- BDBH (default): starting from a reference genome, keep adding genomes stepwise while storing the sequence clusters that result from merging the latest bidirectional best hits.
- OMCL (−M): uses the Markov Cluster Algorithm to group sequences, with inflation (−F) controlling cluster granularity, as described in [26].

As a rule of thumb, use OMCL (−M) for pangenome analyses, as it can handle sequences missing from the reference, and the BDBH algorithm for rapid calculations of core gene/transcript sets.

Moreover, by default redundant isoforms are filtered out (−i 40). However, when working with curated CDS sequences, with one selected sequence per gene, this is not necessary. You can disable this by adding −i 0 to your command line.

By default the input set with least sequences is taken as a reference. However, if you have a previously annotated input set you might want to choose that as a reference, using option −r.

Option −t allows you to control the minimum occupancy of the output clusters. By default only core clusters are produced, while with −t 0 clusters of any size, including singletons and cloud clusters, are output. The latter are named 'control' clusters as they will be used to compute the background frequencies of Pfam domains in later sections.

Option −A, used in the last example below, requests an ANI matrix. In this case it is combined with flag −e to exclude clusters with inparalogues and therefore minimize biases when averaging observed identities. ANI matrices are useful to robustly compute the similarity among your input genotypes and to cluster them, so that redundancy can easily be managed. If you have outgroups among the input sequences, you can also check them in the ANI matrix.

```
# precompute (-o) sequence similarities and protein domain
frequencies (-D);
# clusters are not produced
```

```
# Creates cds_est_homologues/tmp/all.pfam and cds_est_homologues/tmp/all.bpo
$GETHOMS/get_homologues-est.pl -d cds -D -m cluster -o &> log.cds.pfam

# alternatively, if not running in a SGE cluster, taking for instance 4 CPUs
# $GETHOMS/get_homologues-est.pl -d cds -D -n 4 -o &> log.cds.pfam

# calculate 'control' cds clusters (-t 0) using OMCL (-M)
# cluster_list = cds_est_homologues/Alexis_0taxa_algOMCL_e0_.cluster_list
# cluster_directory = cds_est_homologues/Alexis_0taxa_algOMCL_e0_
$GETHOMS/get_homologues-est.pl -d cds -M -t 0 -m cluster &> log.cds

# get non-cloud clusters (-t 3) and percentage of conserved sequence clusters (POCS) matrix (-P)
# cluster_list = cds_est_homologues/Alexis_3taxa_algOMCL_e0_.cluster_list
# cluster_directory = cds_est_homologues/Alexis_3taxa_algOMCL_e0_
# percent_conserved_sequences_file = cds_est_homologues/Alexis_3taxa_algOMCL_e0_POCS.tab
$GETHOMS/get_homologues-est.pl -d cds -M -t 3 -P -m cluster &> log.cds.t3

# single-copy (-e) clusters with high occupancy (-t 10) & Average Nucleotide Identity (-A)
# [Note that flag -e filters out clusters with inparalogues]
# Only high-confidence clusters, containing sequences from at least 10 barleys
# This produces cds_est_homologues/Alexis_10taxa_algOMCL_e1_Avg_identity.tab
# as a summary of 2254 clusters
$GETHOMS/get_homologues-est.pl -d cds -M -t 10 -m cluster -A -e &> log.cds.t10.e

# single-copy (-e) core clusters for phylogenomic analyses (see section 3.3.6)
# number_of_clusters = 883
# cluster_directory = cds_est_homologues/Alexis_alltaxa_algOMCL_e1_
$GETHOMS/get_homologues-est.pl -d cds -M -m cluster -e &> log.cds.core.e
```

```
# Make heatmap and dendrograms based on ANI
# This requires some R dependencies, see install instructions
with
$GETHOMS/plot_matrix_heatmap.sh -M

# You may also want to edit/shorten the labels in ANI tab file
# This produces several outfiles:
# 1) Alexis_10taxa_algOMCL_e1_Avg_identity_heatmap.pdf (Fig-
ure 4)
# 2) Alexis_10taxa_algOMCL_e1_Avg_identity_BioNJ.ph (Neighbor
Joining Newick file)
# 3) ANDg_meand_silhouette_width_statistic_plot.pdf (silhou-
ette width statistics to compute K)
# 4) ANDg_hc_plot_cut_at_mean_silhouette_width_k4.pdf
(K optimal silhouette clusters)

# note that two decimals are used (-d 2)
$ $GETHOMS/plot_matrix_heatmap.sh -i cds_est_homologues/Alex-
is_10taxa_algOMCL_e1_Avg_identity.tab \
 -H 10 -W 15 -t "ANI of single-copy transcripts (occupancy >
9)" -N -o pdf -d 2
```

It is a good idea to inspect the log files to check the version of the software and to make sure that no errors occurred. The logs contain useful information such as the output mask.

```
# mask=Alexis_0taxa_algOMCL_e0_ (_0taxa_algOMCL)
```

This mask is a prefix added to the output folder. In this case it tells that the reference genomes is "Alexis," that no occupancy cutoff was set (0taxa), the clustering algorithm used was OMCL and that clusters containing inparalogues were included (e0, *see* **Note 10**).

Note that internally sequences are numbered with natural numbers, called sequence ids. These numbers start counting from the reference genome, in this case Alexis. This is important because the resulting clusters are also numbered accordingly.

The resulting heatmap summarizing the ANI matrix in the example is shown in Fig. 4. In our experience, ANI matrices computed from core genes can recapitulate the expected phylogeny of your input set [11].

A sample of the resulting POCS matrix is shown in Table 4.

Note that the examples above use −m cluster to parallelize most tasks (*see* Subheading 2.1.5). If your cluster manager is not supported you can always use −m dryrun to generate batch files that should be able to execute.

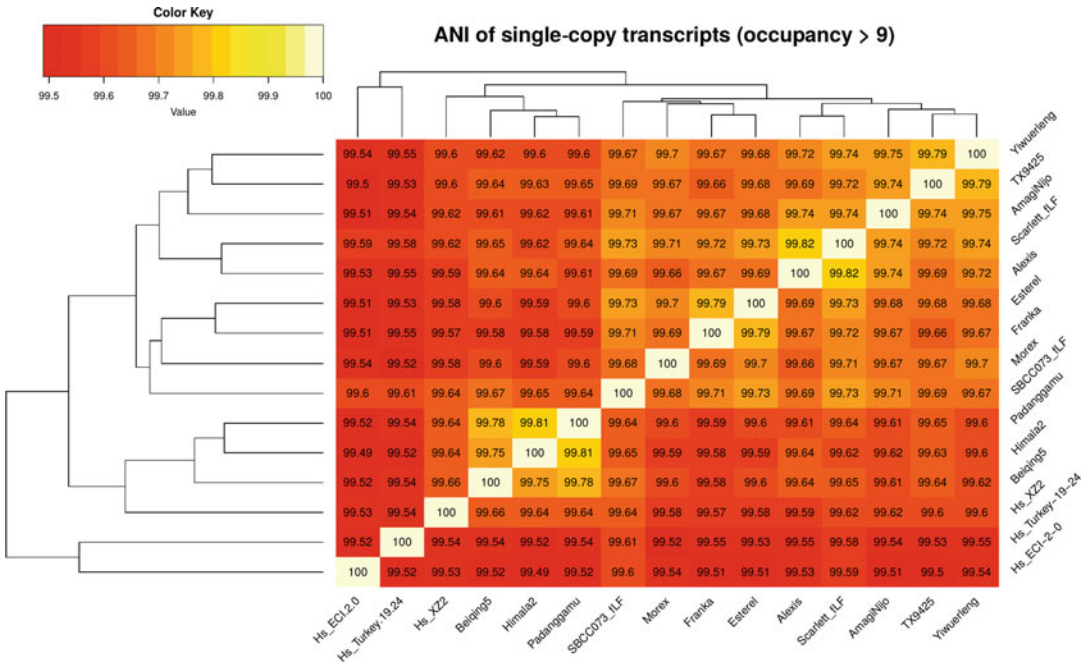


Fig. 4 Average nucleotide % identity matrix and ordered heat map of 2254 single-copy core barley CDS. Values are average nucleotide identities among sequences clustered together by GET_HOMOLOGUES-EST. This figure was produced with script *plot_matrix_heatmap.sh*, which calls heatmap.2 function from the gplots R package. The dendrograms were computed by complete linkage clustering and Euclidean distances computed among ANI columns

Table 4
Excerpt of percentage of conserved sequences (POCS) matrix

Genomes	Alexis	AmagiNijo	Beiqing5	Esterel	Franka	Himala2
Alexis	100	71.54	70.76	72.43	72.06	70.46
AmagiNijo	71.54	100	70.89	71.38	70.90	70.40
Beiqing5	70.76	70.89	100	71.14	71.18	71.80
Esterel	72.43	71.38	71.14	100	73.66	70.79
Franka	72.06	70.90	71.18	73.66	100	70.74
Himala2	70.46	70.40	71.80	70.79	70.74	100

Values are percentages of conserved sequence clusters among pairs of taxa. In other words, these values summarize how many clusters of one genome contain also sequences from another

If you are running GET_HOMOLOGUES-EST on a multicore Linux/macOS box, you can use options −o −n to indicate how many cores should be used to parallelize BLASTN and HMMER jobs. Once those jobs are completed, you can still use

−m dryrun to produce batch files for the remaining steps, which can then be parallelized with the command-line tool *parallel*.

```
# 1) run BLASTN (and HMMER) in batches
$ ${GETHOMS}/get_homologues-est.pl -d cds -o

# 2) run in -m dryrun mode
$ ${GETHOMS}/get_homologues-est.pl -d cds -m dryrun
# ...
# EXIT: check the list of pending commands at cds_est_homolo-
gues/dryrun.txt
parallel < cds_est_homologues/dryrun.txt

# repeat 2) until completion
$ ${GETHOMS}/get_homologues-est.pl -d cds -m dryrun
# ...
```

3.3.3 Annotation of Clusters

The main output of GET_HOMOLOGUES-EST are the clusters generated in the previous section. For instance, file cds_est_homologues/Alexis_3taxa_leaf.list_algOMCL_e0_.cluster_list lists all noncloud clusters, with occupancy $\geq$3. Let us see one of those clusters, 3_TR4758-c0_g1_i1.fna, which contains sequence number 3 of cultivar Alexis.

```
$ cluster 3_TR4758-c0_g1_i1 size=22 taxa=14 file: 3_TR4758-
c0_g1_i1.fna aminofile: 3_TR4758-c0_g1_i1.faa
: Alexis.trinity.fna.bz2_l50_E1e-05.diamond.cds.fna.gz.nucl
:  AmagiNijo.trinity.fna.bz2_l50_E1e-05.diamond.cds.fna.gz.
nucl
:  AmagiNijo.trinity.fna.bz2_l50_E1e-05.diamond.cds.fna.gz.
nucl
: Beiqing5.trinity.fna.bz2_l50_E1e-05.diamond.cds.fna.gz.nucl
: Esterel.trinity.fna.bz2_l50_E1e-05.diamond.cds.fna.gz.nucl
...
:  Yiwuerleng.trinity.fna.bz2_l50_E1e-05.diamond.cds.fna.gz.
nucl
```

In general, clusters are numbered after the sequence id of the first sequence from the reference set included. In this example, the sequence TR4758-c0_g1_i1 was assigned the id 3. If a cluster does not contain any sequence from the reference, then it takes the number from the id of the sequence from the next species.

It contains 22 sequences from 14 barleys, so it belongs to the soft-core occupancy class. As our input are CDS we have two FASTA files for this cluster, one with the nucleotide sequences and another with the amino acid sequences.

```
$ cds_est_homologues/Alexis_3taxa_leaf.list_algOMCL_e0_/
3_TR4758-c0_g1_i1.fna
$ cds_est_homologues/Alexis_3taxa_leaf.list_algOMCL_e0_/
3_TR4758-c0_g1_i1.faa
```

Their contents look like this:

```
>TR4758|c0_g1_i1 [Alexis.trinity.fna.bz2_l50_E1e-05.diamond.
cds.fna.gz] | aligned:1-10896 (10896)
ATGGCAGCGGCGGCGATGGCAGCGCACAGGGCCAGTTTCCCACTGCGGCTGCAGCA-
GATCCTGTCTGGCAGCCGCGCCGTGTCGCCGGCGATCAAAG...
```

and:

```
>TR4758|c0_g1_i1 [Alexis.trinity.fna.bz2_l50_E1e-05.diamond.
cds.fna.gz]
MAAAAMAAHRASFPLRLQQILSGSRAVSPAIKVESEPPAKVKAFIDRVINIPLH-
DIAIPLSGFHWEFNKGN...
```

In this section, we will see how to further annotate these clusters.

For instance, you might want to check the BLASTN evidence supporting this cluster, named after the CDS TR4758|c0_g1_i1, which was assigned sequence id = 3. Note that option −D would output also the Pfam domains called in these sequences, provided that $GETHOMS/get_homologues-est.pl −D was called beforehand.

```
$ $GETHOMS/check_BDBHs.pl -i 3 -d cds_est_homologues/ -e
# construct_taxa_indexes: number of taxa found = 14
# reading redundant isoforms of last run of get_homologues-est
...

# query = 3
# query fullname = TR4758|c0_g1_i1 evidence:transdecoder<-
blastx match:... (nr)

# list of bidirectional best-hits:
dir query sbjct bits Eval %ident cover Pfam annotation
: [AmagiNijo.trinity.fna.bz2_l50_E1e-05.diamond.cds...]
> 3 36115 20072 0 99.9 100.0 NA TR8481|c0_g1_i1 evidence:...
< 36115 3 20072 0 99.9 100.0 NA

: [Beiqing5.trinity.fna.bz2_l50_E1e-05.diamond.cds...]
> 3 71423 16081 0 99.9 100.0 NA TR10181|c0_g1_i1 evidence:...
< 71423 3 16081 0 99.9 100.0 NA
...
```

Another useful script to annotate individual clusters is *annotate_cluster.pl*, which produces a multiple alignment (MSA) view of the locally aligned sequences that make up the cluster. It works by aligning all sequences in the cluster to the longest/user-selected sequence (*see* **Note 11**). Figure 5 summarizes the output of this script, which is particularly useful for raw transcripts.

There are many possible uses for clusters produced by GET_HOMOLOGUES-EST.

- Exploration of genetic variation in expressed/coding sequences. For instance, clusters can be used to compute rates of nonsynonymous to synonymous codon substitutions (*see* **Note 12**).
- As raw material for phylogenomic analyses. Section 3.3.6 provides a guide on how to perform this with the software GET_PHYLOMARKERS, but you can use your favorite tools as well. A particular use case would be to ascertain the genomic composition and the phylogeny of allopolyploids, which are challenging to assemble (*see* **Note 13**).

3.3.4 Pangenome Analyses

The raw clusters from the previous sections can be analyzed in bulk in order to discover properties of the resulting pangenome or pantranscriptome.

We can simulate how a pangenome grows as new genomes are added. In the main script this option is named genome composition (option −c). This is really a simulation where the input sequence sets are sampled in random order and added sequentially. After a new genome is added the algorithm checks whether previous clusters gain sequences or new clusters are added. The process is repeated 20 times (*see* **Note 14**). This kind of analyses can be done with the following.

- CDS sequences from WGS genomes, as in the pioneer work of Tettelin [1]. This is probably the best possible data, provided that the assemblies are of similar quality and the gene annotation obtained with the same methodology.
- Transcripts from the same tissue of several genotypes. Note that samples should be from the same developmental stage, and that might be quite difficult to manage even in controlled experiments.
- Conserved noncoding sequences or annotated transposons.
- Random genome/transcriptome fragments of the same size (k-mers), as done in a recent barley study [39].

In this protocol we will compare leaf tissue CDS sequences from 14 barley cultivars and ecotypes. Note that we use option −z to make a soft-core simulation in addition to the default core and pan simulations. Examples of the resulting plots are shown in Fig. 6.

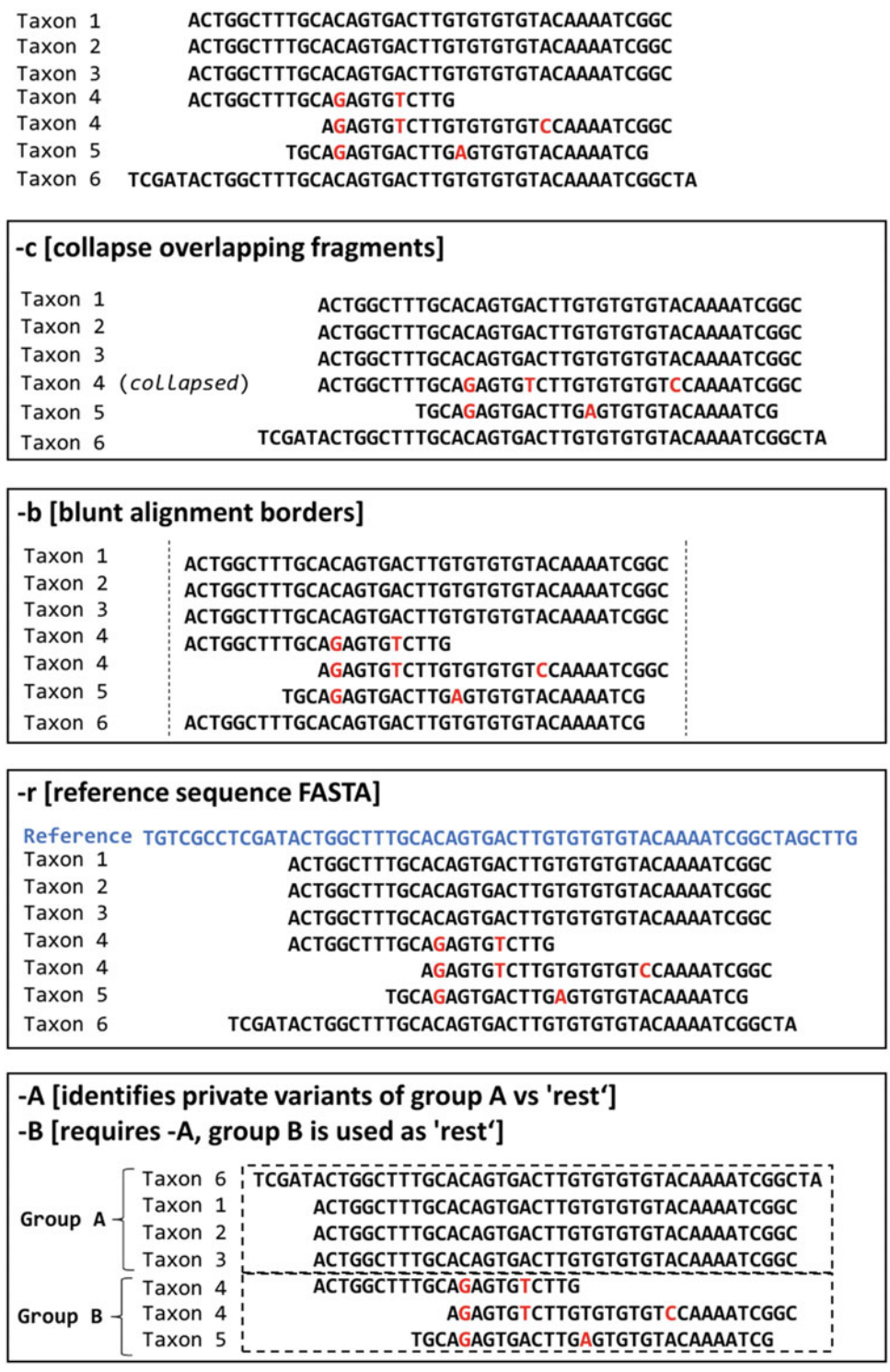


Fig. 5 Summary of script *annotate_clusters.pl*. A cluster of transcript sequences (top) is processed for further downstream analyses by alignment to an external reference sequence (−r), collapsing overlapping sequences of the same taxon (−c) or making the alignment block blunt (−b). In addition, private variants to a group of taxa can be extracted (bottom), and Pfam domains called by translating the longest sequence

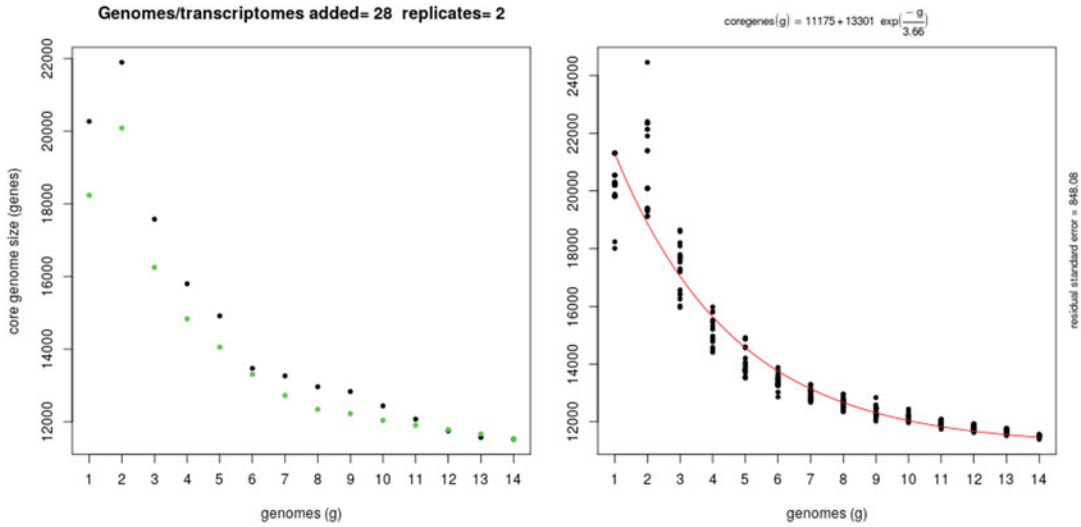


Fig. 6 Soft-core leaf pantranscriptome after comparing 14 barley cultivars and ecotypes. (Left) Snapshot of the simulation after 2 replicates are complete. The data points for the first replicate are in black, and those for the second in green. (Right) Fitted Tettelin function after 20 replicates are complete

Moreover, we also produced pangenome matrices and used them to estimate the size of the different pangenome compartments. The matrices are created with the script *compare_clusters.pl* with option −m. Note that option −n uses nucleotide clusters for building the matrix. In the example we actually call this script twice; the second time cloud clusters are filtered out, as they were found to be the most unreliable in our benchmarks [11].

Another script, *parse_pangenome_matrix.pl* can be used to consume the matrices and produce compartment plots and size estimates. In this example it estimates a core genome of 9815 CDS and a pangenome of 33,648 CDS clusters and produces Fig. 7.

```
# leaf clusters and pantranscriptome growth simulations with
soft-core (-z)
# produces 3 genome composition matrices in cds_est_homolo-
gues/:
# [core|soft-core|pan]_genome_leaf.list_algOMCL.tab
$ $GETHOMS/get_homologues-est.pl -d cds -c -z \
 -I cds/leaf.list -M -t 3 -m cluster &> log.cds.leaf.t3.c

# make pangenome growth plots

# first core plot with two types of fits (Tettelin & Will-
enbrock)
$ $GETHOMS/plot_pancore_matrix.pl -i cds_est_homologues/cor-
e_genome_leaf.list_algOMCL.tab \
```

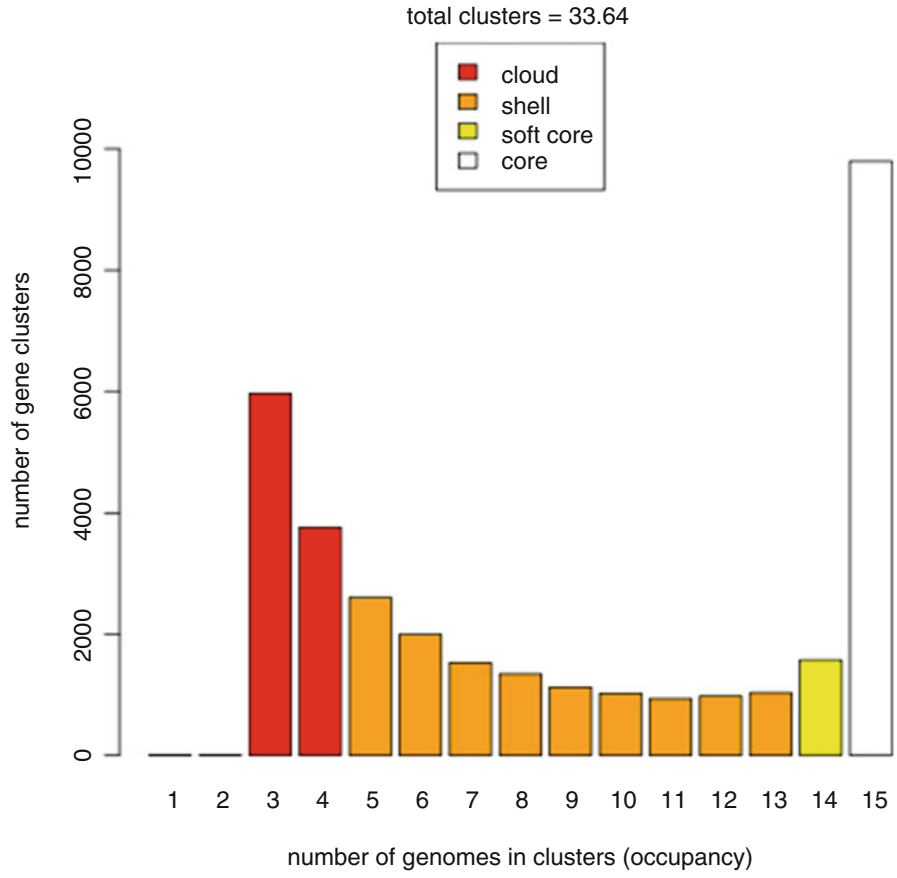


Fig. 7 Distribution of leaf CDS clusters from 15 barleys as a function of their occupancy. Occupancy classes are colored as core, soft-core, and shell members. Cloud clusters were filtered out (−t 3)

```
 -f core_both
# outfiles:
# core_genome_leaf.list_algOMCL.tab_core_both.log (fitted va-
lues and function)
# core_genome_leaf.list_algOMCL.tab_core_both.png/pdf (plots)

# now soft-core making simulation snapshots for figure
$ $GETHOMS/plot_pancore_matrix.pl -i cds_est_homologues/soft-
core_genome_leaf.list_algOMCL.tab \
 -a animation

# finally pan
$ $GETHOMS/plot_pancore_matrix.pl -i cds_est_homologues/pan_-
genome_leaf.list_algOMCL.tab -f pan

## produce pangenome matrices and allocate clusters to occu-
pancy classes
```

```
# all occupancies, produces folder clusters_cds/
$ $GETHOMS/compare_clusters.pl -d cds_est_homologues/Alexis_0taxa_algOMCL_e0_ \
 -o clusters_cds -m -n &> log.compare_clusters.cds

# excluding cloud clusters, the most unreliable in our transcript benchmarks,
# produces folder clusters_cds_t3/
$ $GETHOMS/compare_clusters.pl -d cds_est_homologues/Alexis_3taxa_algOMCL_e0_ \
 -o clusters_cds_t3 -m -n &> log.compare_clusters.cds.t3

## check the log files for the number of clusters
$ cat log.compare_clusters.cds
# ...
# pangenome_file = clusters_cds/pangenome_matrix_t0.tab (and transposed)
# pangenome_genes = clusters_cds/pangenome_matrix_genes_t0.tab (and transposed)
# pangenome_phylip file = clusters_cds/pangenome_matrix_t0.phylip
# pangenome_FASTA file = clusters_cds/pangenome_matrix_t0.fasta
# pangenome CSV file (Scoary) = $ clusters_cds/pangenome_matrix_t0.tr.csv

# plot compartments and perform mixture model pangenome size estimates
# creates _list.txt files with clusters in each compartment
$ $GETHOMS/parse_pangenome_matrix.pl -m clusters_cds_t3/pangenome_matrix_t0.tab -s \
 &> log.parse_pangenome_matrix.cds.t3

$ cat log.parse_pangenome_matrix.cds.t3
# matrix contains 33644 clusters and 15 taxa
# ...
# pangenome size estimates (Snipen mixture model PMID:19691844): $ clusters_cds_t3/pangenome_matrix_t0__shell_estimates.tab
Core.size Pan.size BIC LogLikelihood
2 components 9815 33648 198886.077102331 -99427.4031661459
```

As you can see from the output, script *compare_clusters.pl* produces several versions of the same matrix, which describe a pangene set (*see* **Note 15**).

- *pangenome_matrix_t0.tab* is a numeric matrix with tab-separated (TSV) columns, with taxa/genomes as rows and

sequence clusters as columns, in which cells with natural numbers indicate whether a given taxa contains 1+ sequences from a given cluster. It can be read and edited with any text editor or spreadsheet software, and is also produced in transposed form for convenience. For example, users might want to sort the clusters by position on a reference genome and use these matrices to visualize results.

- *pangenome_matrix_genes_t0.tab* is similar to the previous one, but contains the actual sequence names in each cluster instead.
- *pangenome_matrix_t0.phylip* is a reduced binary matrix in a format suitable for PHYLIP discrete character analysis software,
- *pangenome_matrix_t0.fasta* is a reduced binary matrix in FASTA format suitable for binary character analysis software such as IQ-TREE, which can compute bootstrap and aLRT support.
- *pangenome_matrix_t0.tr.csv* is a transposed, reduced binary matrix in CSV format suitable for pangenome-wide association analysis with software Scoary (*see* **Note 16**).

These presence-absence matrices can be used directly to infer population phylogenies (*see* **Note 17**), particularly if your data are CDS sequences from WGS genomes [12].

Moreover, it is possible to annotate pangenome clusters with script *make_nr_pangenome_matrix.pl* and a user-provided FASTA file of curated sequences. For instance, you might want to identify clusters containing genes from transposable elements (*see* **Note 18**), or low-copy genes of flowering plants (*see* **Note 19**).

3.3.5 Analysis of Accessory Genes/ Transcripts

In this section, we will see how to extract accessory genes of interest and how to check whether they are enriched in some protein domains with respect to the rest of the genome. For this you need to define lists of cultivars/genotypes to be compared.

```
# find [-t 3] clusters from cultivar SBCC073 which are absent
from reference Morex
$ $GETHOMS/parse_pangenome_matrix.pl -m
clusters_cds_t3/pangenome_matrix_t0.tab \
 -A cds/SBCC073.list -B cds/ref.list -g &> log.acc.SBCC073
$ mv clusters_cds_t3/pangenome_matrix_t0__pangenes_list.txt \
 clusters_cds_t3/SBCC073_pangenes_list.txt

$ head clusters_cds_t3/SBCC073_pangenes_list.txt
# genes present in set A and absent in B (5531):
172_TR8839-c0_g1_i4.fna
...

# how many SBCC073 non-cloud clusters are there?
$ perl -lane 'if($F[0] =~ /SBCC073/){ foreach $c (1 .. $#F){ if
```

```
($F[$c]>0){ $t++ } }; print $t }' \
 clusters_cds_t3/pangenome_matrix_t0.tab
21041

# Pfam enrichment tests (-c control set, -x experiment set)

# negative control, is the core genome enriched in some Pfam
domains
# with respect to the complete genome?
$ $GETHOMS/pfam_enrich.pl -d cds_est_homologues -c clusters_cds
-n \
  -x clusters_cds_t3/pangenome_matrix_t0__core_list.txt -e -p 1 \
  -r SBCC073 > SBCC073_core.pfam.enrich.tab

# positive control, is the core depleted in some Pfam domains?
$ $GETHOMS/pfam_enrich.pl -d cds_est_homologues -c clusters_cds
-n \
  -x clusters_cds_t3/pangenome_matrix_t0__core_list.txt -e -p 1 \
  -r SBCC073 -t less > SBCC073_core.pfam.deplet.tab

# are SBCC073 accessory genes enriched in some Pfam domains?
# Note that the experiment sequences are output in FASTA format
$ $GETHOMS/pfam_enrich.pl -d cds_est_homologues -c clusters_cds
-n \
  -x clusters_cds_t3/SBCC073_pangenes_list.txt -e -p 1 -r SBCC073 \
  -f SBCC073_accessory.fna > SBCC073_accessory.pfam.enrich.tab
```

The top enriched domains of the last example are shown in Table 5.

If you are looking for accessory genes and transcripts, as in the first example of this section, there are several points that should be considered before doing any kind of biological interpretation.

- You should try to compare genotypes with assemblies of similar quality and gene models called with the same methodology. Otherwise, missing genes are most likely due to these

Table 5
Top 3 enriched Pfam domains encoded in accessory (Exp) CDS found in barley landrace SBCC073 with respect to the complete CDS set (Ctr)

PfamID	Counts (exp)	Counts (ctr)	Freq (exp)	Freq (ctr)	*p*-value	*p*-value (adj)	Description
PF13976	14	19	7.37e−03	9.71e−04	2.99e−07	3.85e−04	GAG-pre-integrase domain
PF01535	59	305	3.10e−02	1.55e−02	5.95e−06	5.74e−03	PPR repeat
PF00560	32	128	1.68e−02	6.54e−03	1.14e−05	8.81e−03	Leucine rich repeat

Both raw and adjusted Fisher's test *P*-values are shown

methodological differences. In our experience these differences show up when calling accessory genes. For this reason we usually leave out cloud genes/transcripts, as singletons have a good chance of being artifacts.

- A small assembly error might cause a genuine gene to be split in two gene models. In our *B. distachyon* benchmarks these would be clustered together in most cases, but it is worth double-checking [11].
- When working with transcripts, a CDS sequence might be missing due to insufficient sequencing depth, weak expression or a tissue sampled in the wrong developmental stage.
- Soft-core genes and transcripts are robust to those factors to some extent.

3.3.6 Downstream Phylogenomic Analyses

GET_HOMOLOGUES-EST produces two types of output which can be further processed with the GET_PHYLOMARKERS pipeline [33] (https://github.com/vinuesa/get_phylomarkers) to compute robust molecular phylogenies:

- Single-copy core clusters (twin .fna and .faa FASTA files). These can be CDS sequences annotated in WGS genomes, or deduced from transcriptomes, which can be used to compute gene trees which are quality-controlled and eventually concatenated. In our experience this provides useful insights at the genus level or beyond. As mentioned on Subheading 3.3.1, outgroups will be required to root the resulting trees.
- Pangenome matrices, ideally of CDS sequences from WGS genomes, as our benchmarks with transcript-based matrices do not always reconstruct the expected topologies [11].

Both strategies use IQ-TREE [42] and are explained in detail on a dedicated tutorial (*see* **Note 20**); here, we will briefly demonstrate how to use them.

```
# set GET_PHYLOMARKERS path, for example:
$ cd soft
$ git clone https://github.com/vinuesa/get_phylomarkers.git
$ export GETPHYLO=~/soft/get_phylomarkers

# install R dependencies
$ cd $GETPHYLO
$ ./install_R_deps.R

## pangenome matrix phylogeny

# Search for the best maximum likelihood tree using 10 indepen-
dent IQ-TREE runs
```

```
# with a previously computed pangenome matrix in FASTA format
(section 3.3.4)
# Note: the script can also run PHYLIP pars, which are much
slower
$ ${GETPHYLO}/estimate_pangenome_phylogenies.sh -f pangen-
ome_matrix_t0.fasta -r 10 -S UFBoot
#...
>>> wrote file sorted_lnL_scores_IQ-TREE_searches.out ...
>>> Best IQ-TREE run was: UFBoot_run9.log:BEST SCORE FOUND :
-250734.229 ...
>>> wrote file best_PGM_IQT_UFBoot_run9_GTR2+FO+I.treefile
...
... found in iqtree_PGM_10_runs/iqtree_10_runs
... done!

## single-copy core cluster phylogeny

# Move to folder with single-copy core .faa & .fna clusters,
compute codon alignments
# and trees of coding sequences using molecular model selection
and produce a
# species-tree from the concatenated, top-scoring alignments.
# See all options with ./run_get_phylomarkers_pipeline.sh -H
# For individuals of the same species you can also try -m 0.5
$ cd Alexis_alltaxa_algOMCL_e1_
${GETPHYLO}/run_get_phylomarkers_pipeline.sh -R 1 -t DNA -f
EST
#...
[13:44:16] # running compute_MJRC_tree treefile I ...
 >>> Wrote file IQT_MJRC_tree.nwk ...
[13:44:16] # running ModelFinder on the concatenated alignment
with GTR. This will take a while ...
 >>> wrote file concat_cdnAlns.fnainf.log ...
 >>> Best-fit model: GTR+F+ASC+R2 ...
[13:44:26] # running IQ-tree on the concatenated alignment
with best model GTR+F+ASC+R2 -abayes -bb 1000. This will take a
while ...
 >>> wrote file sorted_IQ-TREE_searches.out ...
# >>> Best IQ-TREE run was: BEST SCORE FOUND : -6661.697 ...
[13:44:40] >>> wrote file concat_nonRecomb_KdeFilt_iqtree_GTR
+F+ASC+R2.treefile ...

# The final concatenated & labelled tree, with
# Bayesian posterior probabilities/bootstrap values per node,
can be found at
# cd get_phylomarkers_run_AIR1tDNA_k1.5_m0.75_Tmedium*/Phi-
Pack/non_recomb_cdn_alns/top_*/
```

```
# concat_nonRecomb_KdeFilt_iqtree_GTR+F+ASC+G4_ed.sptree
```

The resulting trees are in Newick format and can be plotted with tools such as FigTree or iToL or any other preferred software (*see* **Note 21**).

4 Notes

1. Read [22, 23] for other alternatives.
2. The bacterial manual at http://eead-csic-compbio.github.io/get_homologues/manual can also be useful.
3. This will only work if GET_HOMOLOGUES-EST is installed where the compute cluster management software is accessible. This is usually not the case for Docker containers.
4. See https://cran.r-project.org/package=ape, https://cran.r-project.org/package=dendextend, https://cran.r-project.org/package=factoextra and https://cran.r-project.org/package=gplots.
5. See benchmark in [11]; DIAMOND can also be used with significant gains in computing time and very small sensitivity losses, as shown in the manual.
6. Similar analyses can be performed with the *Arabidopsis thaliana* and *Brachypodium hybridum* sequences at http://floresta.eead.csic.es/plant-pan-genomes/
7. An outgroup is a taxon or lineage that falls outside the clade being studied (ingroup) but it is closely related. It helps root the tree.
8. Global variable $MINSEQLENGTH controls the minimum length of sequences to be considered; the default value is 20 but you can edit in the source to increase it. Similarly, $MAXSEQLENGTH limits the max length for input sequences, 25 Kb by default. In our barley tests, we found some high-quality full-length cDNAs reach 23 Kb [32].
9. Software choices include CPC2 and RNAsamba [36, 37], trained on human truth sets, and BUSCO [38].
10. Inparalogues are sequences whose best BLASTN hit is from the same genome.
11. It does not necessarily conserve the original sequence order. Any sequence stretches masked by BLAST will appear as Xs. Clusters of transcripts might contain sequences that do not match the longest sequence; instead, they align toward the 5′ or 3′ of other sequences and are not included in the produced cumulative MSA.

12. This requires single-copy clusters with at least 4 sequences (−e −t 4), see more at $GETHOMS/user_utils/dNdS.
13. See https://github.com/eead-csic-compbio/allopolyploids
14. You can control this with global variable $NOFSAMPLESREPORT.
15. Similar matrices can be obtained for clades in Ensembl Plants with scripts at https://github.com/Ensembl/plant-scripts/tree/master/phylogenomics
16. See https://github.com/AdmiralenOla/Scoary
17. You might want to try *compare_clusters.pl* -T for a PHYLIP pars parsimony tree or GET_PHYLOMARKERS pipeline, as explained on Subheading 3.3.6.
18. See for instance the nrTEplants library at https://github.com/Ensembl/plant-scripts/releases/download/v0.3/nrTEplantsJune2020.fna.bz2, described at https://github.com/Ensembl/plant_tools/tree/master/bench/repeat_libs
19. See https://github.com/mossmatters/Angiosperms353 [40] or http://sftp.kew.org/pub/treeoflife/current_release/fasta/by_gene [41].
20. See https://vinuesa.github.io/get_phylomarkers, it includes documentation on how to run a Docker container bundled with both GET_HOMOLOGUES and GET_PHYLOMARKERS. This minimizes trouble installing dependencies and produces reproducible analyses.
21. See http://tree.bio.ed.ac.uk/software/figtree and https://itol.embl.de

Acknowledgments

A first draft of this protocol was funded by Centro de Bioinformática y Biología Computacional de Colombia—BIOS for a workshop organized by Marco Cristancho at Manizales, Colombia, in March 2017. We also received funding from Fundación ARAID and the Spanish Ministry of Economy and Competitivity (CSIC13-4E-249, AGL2013-48756-R, AGL2016-80967-R, CGL2016-79790-P). PV acknowledges support from CONACyT Mexico (A1-S-11242) and PAPIIT-UNAM (IN206318 and IN209321). We thank Brett Chapman for proofreading the manuscript.

References

1. Tettelin H, Masignani V, Cieslewicz MJ et al (2005) Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial "pan-genome". Proc Natl Acad Sci U S A 102:13950–13955
2. Golicz AA, Bayer PE, Bhalla PL, Batley J, Edwards D (2020) Pangenomics comes of age: from bacteria to plant and animal applications. Trends Genet 36:132–145
3. Yano K, Yamamoto E, Aya K, Takeuchi H, Lo PC, Hu L, Yamasaki M, Yoshida S, Kitano H, Hirano K, Matsuoka M (2016) Genome-wide association study using whole-genome sequencing rapidly identifies new genes influencing agronomic traits in rice. Nat Genet 48: 927–934
4. Della Coletta R, Qiu Y, Ou S, Hufford MB, Hirsch CN (2021) How the pan-genome is changing crop genomics and improvement. Genome Biol 22:3
5. Laing C, Buchanan C, Taboada EN, Zhang Y, Kropinski A, Villegas A, Thomas JE, Gannon VP (2010) Pan-genome sequence analysis using Panseq: an online tool for the rapid analysis of core and accessory genomic regions. BMC Bioinformatics 11:461
6. Bayer PE, Golicz AA, Scheben A, Batley J, Edwards D (2020) Plant pan-genomes are the new reference. Nat Plants 6:914–920
7. Eizenga JM, Novak AM, Sibbesen JA, Heumos S, Ghaffaari A, Hickey G, Chang X, Seaman JD, Rounthwaite R, Ebler J, Rautiainen M, Garg S, Paten B, Marschall T, Sirén J, Garrison E (2020) Pangenome graphs. Annu Rev Genomics Hum Genet 21:139–162
8. Sheikhizadeh S, Schranz ME, Akdel M, de Ridder D, Smit S (2016) PanTools: representation, storage and exploration of pan-genomic data. Bioinformatics 32:i487–i493
9. Voichek Y, Weigel D (2020) Identifying genetic variants underlying phenotypic variation in plants without complete genomes. Nat Genet 52:534–540
10. Arora S, Steuernagel B, Gaurav K et al (2019) Resistance gene cloning from a wild crop relative by sequence capture and association genetics. Nat Biotechnol 37:139–143
11. Contreras-Moreira B, Cantalapiedra C, Garcia-Pereira M, Gordon S, Vogel J, Igartua E, Casas A, Vinuesa P (2017) Analysis of plant pan-genomes and transcriptomes with get_HOMOLOGUES-Est, a clustering solution for sequences of the same species. Front Plant Sci 8:184
12. Gordon SP, Contreras-Moreira B, Woods DP et al (2017) Extensive gene content variation in the Brachypodium distachyon pan-genome correlates with population structure. Nat Commun 8:2184
13. Gordon SP, Contreras-Moreira B, Levy JJ et al (2020) Gradual polyploid genome evolution revealed by pan-genomic analysis of Brachypodium hybridum and its diploid progenitors. Nat Commun 11:3670
14. Wang B, Tseng E, Regulski M, Clark TA, Hon T, Jiao Y, Lu Z, Olson A, Stein JC, Ware D (2016) Unveiling the complexity of the maize transcriptome by single-molecule long-read sequencing. Nat Commun 7:11708
15. Minio A, Massonnet M, Figueroa-Balderas R, Vondras AM, Blanco-Ulate B, Cantu D (2019) Iso-seq allows genome-independent transcriptome profiling of Grape Berry development. G3 (Bethesda) 9:755–767
16. Welch RA, Burland V, Plunkett G, Redford P, Roesch P, Rasko D, Buckles EL, Liou SR, Boutin A, Hackett J, Stroud D, Mayhew GF, Rose DJ, Zhou S, Schwartz DC, Perna NT, Mobley HL, Donnenberg MS, Blattner FR (2002) Extensive mosaic structure revealed by the complete genome sequence of uropathogenic Escherichia coli. Proc Natl Acad Sci U S A 99:17020–17024
17. Morgante M, De Paoli E, Radovic S (2007) Transposable elements and the plant pan-genomes. Curr Opin Plant Biol 10: 149–155
18. Marroni F, Pinosio S, Morgante M (2014) Structural variation and genome complexity: is dispensable really dispensable? Curr Opin Plant Biol 18:31–36
19. Sielemann K, Weisshaar B, Pucker B (2021) Reference-based QUantification of gene dispensability (QUOD). Plant Methods 17:18
20. Contreras-Moreira B, Vinuesa P (2013) GET_HOMOLOGUES, a versatile software package for scalable and robust microbial pangenome analysis. Appl Environ Microbiol 79: 7696–7701
21. Vinuesa P, Contreras-Moreira B (2015) Robust identification of orthologues and paralogues for microbial pan-genomics using GET_HOMOLOGUES: a case study of pIncA/C plasmids. Methods Mol Biol 1231: 203–232
22. Golicz AA, Batley J, Edwards D (2016) Towards plant pangenomics. Plant Biotechnol J 14:1099–1105

23. Vernikos GS (2020) A review of pangenome tools and recent studies. In: Tettelin H, Medini D (eds) The pangenome: diversity, dynamics and evolution of genomes. Springer International, Cham, pp 89–112
24. Mistry J, Chuguransky S, Williams L, Qureshi M, Salazar GA, Sonnhammer ELL, Tosatto SCE, Paladin L, Raj S, Richardson LJ, Finn RD, Bateman A (2021) Pfam: the protein families database in 2021. Nucleic Acids Res 49:D412–D419
25. Bateman A, Martin MJ, Orchard S et al (2021) UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Res 49:D480–D489
26. Li L, Stoeckert CJ, Roos DS (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. Genome Res 13: 2178–2189
27. Pertea G, Huang X, Liang F, Antonescu V, Sultana R, Karamycheva S, Lee Y, White J, Cheung F, Parvizi B, Tsai J, Quackenbush J (2003) TIGR gene indices clustering tools (TGICL): a software system for fast clustering of large EST datasets. Bioinformatics 19: 651–652
28. Willenbrock H, Hallin PF, Wassenaar TM, Ussery DW (2007) Characterization of probiotic Escherichia coli isolates with a novel pan-genome microarray. Genome Biol 8:R267
29. Snipen L, Almoy T, Ussery DW (2009) Microbial comparative pan-genomics using binomial mixture models. BMC Genomics 10:385
30. Qin QL, Xie BB, Zhang XY, Chen XL, Zhou BC, Zhou J, Oren A, Zhang YZ (2014) A proposed genus boundary for the prokaryotes based on genomic insights. J Bacteriol 196: 2210–2215
31. Popescu AA, Huber KT, Paradis E (2012) Ape 3.0: new tools for distance-based phylogenetics and evolutionary analysis in R. Bioinformatics 28:1536–1537
32. Sato K, Tanaka T, Shigenobu S, Motoi Y, Wu J, Itoh T (2016) Improvement of barley genome annotations by deciphering the Haruna Nijo genome. DNA Res 23:21–28
33. Vinuesa P, Ochoa-Sanchez LE, Contreras-Moreira B (2018) GET_PHYLOMARKERS, a software package to select optimal orthologous clusters for phylogenomics and inferring pan-genome phylogenies, used for a critical Geno-taxonomic revision of the genus Stenotrophomonas. Front Microbiol 9:771
34. Howe KL, Contreras-Moreira B, De Silva N et al (2019) Ensembl genomes 2020-enabling non-vertebrate genomic research. Nucleic Acids Res 48:D689–D695
35. Goodstein DM, Shu S, Howson R, Neupane R, Hayes RD, Fazo J, Mitros T, Dirks W, Hellsten U, Putnam N, Rokhsar DS (2012) Phytozome: a comparative platform for green plant genomics. Nucleic Acids Res 40: D1178–D1186
36. Kang YJ, Yang DC, Kong L, Hou M, Meng YQ, Wei L, Gao G (2017) CPC2: a fast and accurate coding potential calculator based on sequence intrinsic features. Nucleic Acids Res 45:W12–W16
37. Camargo AP, Sourkov V, Pereira GAG, Carazzolle MF (2020) RNAsamba: neural network-based assessment of the protein-coding potential of RNA sequences. NAR Genom Bioinform 2:lqz024
38. Seppey M, Manni M, Zdobnov EM (2019) BUSCO: assessing genome assembly and annotation completeness. Methods Mol Biol 1962:227–245
39. Jayakodi M, Padmarasu S, Haberer G et al (2020) The barley pan-genome reveals the hidden legacy of mutation breeding. Nature 588:284–289
40. Johnson MG, Pokorny L, Dodsworth S, Botigué LR, Cowan RS, Devault A, Eiserhardt WL, Epitawalage N, Forest F, Kim JT, Leebens-Mack JH, Leitch IJ, Maurin O, Soltis DE, Soltis PS, Wong GK, Baker WJ, Wickett NJ (2019) A universal probe set for targeted sequencing of 353 nuclear genes from any flowering plant designed using k-Medoids clustering. Syst Biol 68:594–606
41. Baker WJ, Bailey P, Barber V et al (2021) A comprehensive phylogenomic platform for exploring the angiosperm tree of life. bioRxiv
42. Nguyen LT, Schmidt HA, von Haeseler A, Minh BQ (2015) IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Mol Biol Evol 32: 268–274
43. Kaas RS, Friis C, Ussery DW, Aarestrup FM (2012) Estimating variation within the genes and inferring the phylogeny of 186 sequenced diverse Escherichia coli genomes. BMC Genomics 13:577
44. Enright AJ, Van Dongen S, Ouzounis CA (2002) An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res 30:1575–1584
45. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25:3389–3402

46. Stajich JE, Block D, Boulez K et al (2002) The Bioperl toolkit: Perl modules for the life sciences. Genome Res 12:1611–1618
47. Haas BJ, Papanicolaou A, Yassour M et al (2013) De novo transcript sequence reconstruction from RNA-seq using the trinity platform for reference generation and analysis. Nat Protoc 8:1494–1512
48. Brown NP, Leroy C, Sander C (1998) MView: a web-compatible database search or multiple alignment viewer. Bioinformatics 14:380–381
49. Buchfink B, Xie C, Huson DH (2015) Fast and sensitive protein alignment using DIAMOND. Nat Methods 12:59–60

Chapter 10

Metagenomics Bioinformatic Pipeline

Diego Garfias-Gallegos, Claudia Zirión-Martínez, Edder D. Bustos-Díaz, Tania Vanessa Arellano-Fernández, José Abel Lovaco-Flores, Aarón Espinosa-Jaime, J. Abraham Avelar-Rivas, and Nelly Sélem-Mójica

Abstract

Microbial communities' taxonomic and functional diversity has been broadly studied since sequencing technologies enabled faster and cheaper data obtainment. Nevertheless, the programming skills needed and the amount of software available may be overwhelming to someone trying to analyze these data. Here, we present a comprehensive and straightforward pipeline that takes shotgun metagenomics data through the needed steps to obtain valuable results. The raw data goes through a quality control process, metagenomic assembly, binning (the obtention of single genomes from a metagenome), taxonomic assignment, and taxonomic diversity analysis and visualization.

Key words Metagenomics, Trimming, Assembly, Binning, Taxonomic assignment, Diversity, Data visualization

1 Introduction

Shotgun metagenomics is a sequencing approach used to obtain the genomic information of all the microorganisms present in a diverse sample. Scientists use metagenomics data to analyze the microbial community in terms of taxonomic composition, the genes present in the whole community, or the signs of selection over a single taxon, to name a few [1].

Regardless of the specific analyses, there are common steps needed to work with metagenomics data. First of all, users must have an organized file system (Subheading 3.1) and the beginning of the data processing must be to remove low-quality sequences from the sequenced reads. We achieve this by using FastQC [2] to visualize the quality (Subheading 3.2) and Trimmomatic [3] to filter low-quality reads and trim low-quality bases (Subheading 3.3).

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_10,

Since the data consists of reads that are usually short (100–250 bases), an assembly step that creates large sequences from short reads is often necessary (Subheading 3.4). The assembly of metagenomic reads faces specific challenges, and algorithms must acknowledge them. We assemble the metagenome using metaSPAdes [4], a software designed to meet difficulties like the differences in coverage, the conserved regions across several taxa, and the presence of multiple strains of one taxon in the community.

If the researcher intends to study a specific taxon in the community, the individual genomes of the community can be separated if they meet certain criteria, like having enough sequencing depth. The metagenomic binning process generates these genomes that are called MAGs (metagenome-assembled genomes). MaxBin is an algorithm that uses differences in coverage and tetranucleotide composition to separate the contigs or scaffolds of a metagenomic assembly into bins [5]. We use this binning strategy and analyze the quality of the resulting MAGs with CheckM (Subheading 3.5). This software calculates MAGs completeness and contamination by counting known genetic markers of a given taxon or lineage or by its localization on a phylogenetic tree [6].

Most metagenomics studies need to perform a taxonomic assignment to know the community's composition or the taxon of the obtained MAGs. We use Kraken2 [7] for the assignment (Subheading 3.6) and the R package Phyloseq [8] for taxonomic composition visualization (Subheading 3.7). Also, we leverage this tool's capability by performing diversity analyses (Fig. 1), which can be very useful in microbial ecology studies (Subheading 3.8).

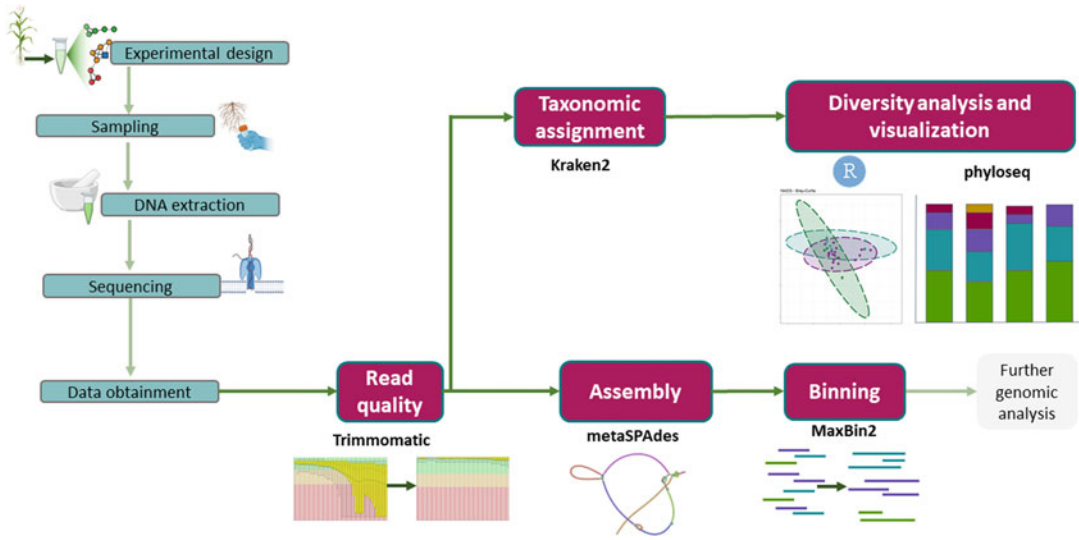


Fig. 1 A basic representation of the tasks in metagenomic research. Blue squares represent the experimental steps while red squares show the bioinformatic pipeline tackled with this protocol

2 Materials

2.1 Hardware

A system with at least 64 Gb of RAM, 300 Gb of hard drive, and a multicore CPU (with at least six logic cores) are required to process the data used in the pipeline presented here. Most laptops and desktops do not fulfill these requirements, so renting a virtual machine is advised to obtain the needed hardware (*see* **Note 1**).

2.2 Software

This entire pipeline can run in the command line of a 64-bit Linux/Unix system or in Windows V10 with Windows subsystem for Linux (https://docs.microsoft.com/en-us/windows/wsl/install-win10) installed.

2.2.1 For Bash

Bash enables fast and customizable use of algorithms designed to analyze metagenomic data, generating valuable outputs for further data inspection. This pipeline uses the following open-source software that runs on a bash terminal (*see* **Note 2**):

SRA toolkit v2.11.0 (https://github.com/ncbi/sra-tools), FastQC v0.11.7 (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/), Trimmomatic v0.38 (http://www.usadellab.org/cms/?=trimmomatic), Kraken2 v2.1.1(http://ccb.jhu.edu/software/kraken2/), MaxBin2 v2.2.7 (https://anaconda.org/bioconda/maxbin2), metaSPAdes v3.14.1 (https://cab.spbu.ru/software/spades/), kraken-biom v1.0.1 (https://github.com/smdabdoub/kraken-biom), and CheckM v1.1.3 (https://ecogenomics.github.io/CheckM/).

2.2.2 For R

R allows users to use community-designed packages to dissect the Shell obtained outputs and generate publishable results. R install instructions are located at https://www.r-project.org/, and R Studio at https://www.rstudio.com/.

This pipeline uses the R packages Phyloseq v 1.36.0 (https://github.com/joey711/phyloseq), and ggplot2 v 3.3.3 (https://ggplot2.tidyverse.org/).

2.3 Files

The entire analysis is carried out with paired-end fastq files of short reads from Illumina sequencing. Each sample produces two fastq files of similar size, each with a differential designation for forward and reverse reads (e.g., 1.fastq and 2.fastq, respectively). Also, the Kraken-database files must be downloaded and located in the instance chosen for the analysis [7]. The files for the example used in this chapter are taken from shotgun sequenced maize root microbiome from Fadiji et al. [9].

3 Methods

All the next text lines beginning with the "$" and ">" symbols correspond to bash and R code, respectively. The output lines will be highlighted in gray as follows: results. Both commands and outputs will have `Courier New` as its particular font and single line spacing. Alongside the entire pipeline, names, directories, and functions will not be named using special characters (e.g., "´", "ñ", "^", "‘") or spaces. The authors advise applying this rule to this pipeline and for the entire bioinformatics exercises of the practitioners.

3.1 Project Organization and Data Obtainment

When beginning a metagenomics project, file and output management must be a top priority. Start by creating a directory where every project file will be saved by using bash commands "cd" and "mkdir." In this example, the directory will be inside the "Documents" folder of our home location (*see* **Note 3**):

```
$ cd ~/Documents/
$ mkdir meta-project
```

Here, create folders for the raw data and the generated outputs from the programs that will be used.

```
$ mkdir raw-reads
$ mkdir results
$ mkdir results/taxonomy
$ mkdir results/assemblies
$ mkdir results/fastqc
$ mkdir results/trimmed-reads
$ mkdir results/untrimmed-reads
$ mkdir r-analysis
$ tree
```

```
.
|── r-analysis
|── raw-reads
|── results
 |── assemblies
 |── fastqc
 |---- taxonomy
 |── trimmed-reads
 |── untrimmed-reads

9 directories, 0 files
```

Next, move to the "raw-reads" folder to download the data to begin the pipeline. From the "Data availability" section in the published work [9], SRA numbers of three of the samples from this study can be obtained. Download the data using the command *fastq-dump* from the "SRA-toolkit", with the next lines of code (*see* **Note 4**).

```
$ cd raw-reads/
$ fastq-dump -I --split-files SRR11131028
$ fastq-dump -I --split-files SRR11131029
$ fastq-dump -I --split-files SRR11131030
```

Six files will be displayed as the output by enlisting the folder contents, each pair of files (_1.fastq and _2.fastq corresponding to forward and reverse, respectively) belong to the information of one sample:

```
$ ls

SRR11131028_1.fastq SRR11131029_1.fastq SRR11131030_1.fastq
SRR11131028_2.fastq SRR11131029_2.fastq SRR11131030_2.fastq
```

3.2 Assessing Read Quality

'FastQC' is used to visualize the read quality of each sample. It is important to be in the folder where the ".fastq" files are located.

```
$ cd ~/Documents/meta-project/raw-reads/
```

'FastQC' accepts uncompressed and compressed files. Right now the files are decompressed, so the program can be run using a wildcard as follows.

```
$ fastqc *.fastq
```

A series of lines will be displayed as an indicator that FastQC has begun the analysis.

```
Started analysis of SRR11131028_1.fastq
Approx 5% complete for SRR11131028_1.fastq
Approx 10% complete for SRR11131028_1.fastq
Approx 15% complete for SRR11131028_1.fastq
Approx 20% complete for SRR11131028_1.fastq
Approx 25% complete for SRR11131028_1.fastq...
```

This will produce a pair of files for each sample in the folder where the reads are located.

```
SRR11131028_1_fastqc.html SRR11131029_2_fastqc.html
SRR11131028_1_fastqc.zip SRR11131029_2_fastqc.zip
SRR11131028_2_fastqc.html SRR11131030_1_fastqc.html
SRR11131028_2_fastqc.zip SRR11131030_1_fastqc.zip
SRR11131029_1_fastqc.html SRR11131030_2_fastqc.html
SRR11131029_1_fastqc.zip SRR11131030_2_fastqc.zip
```

If raw data files were compressed, a command using the file extension and a wildcard can be used. As an example, if the user has ".gz" compressed files, the next line will also run the program "FastQC" on the reads.

```
$ fastqc *.gz
```

The ".html" files contain the FastQC report that can be explored using standard internet browsers (Fig. 2).

FastQC outputs are saved in a different folder to follow a good project organization. With the next line of commands, the ".html" and "zip" files will be moved to the new location.

```
$ mv *.zip ~/Documents/meta-project/results/fastqc/
$ mv *.html ~/Documents/meta-project/results/fastqc/
```

Move to this location. Decompress the "zip" files to see what's inside using a "for" loop (*see* **Note 5**).

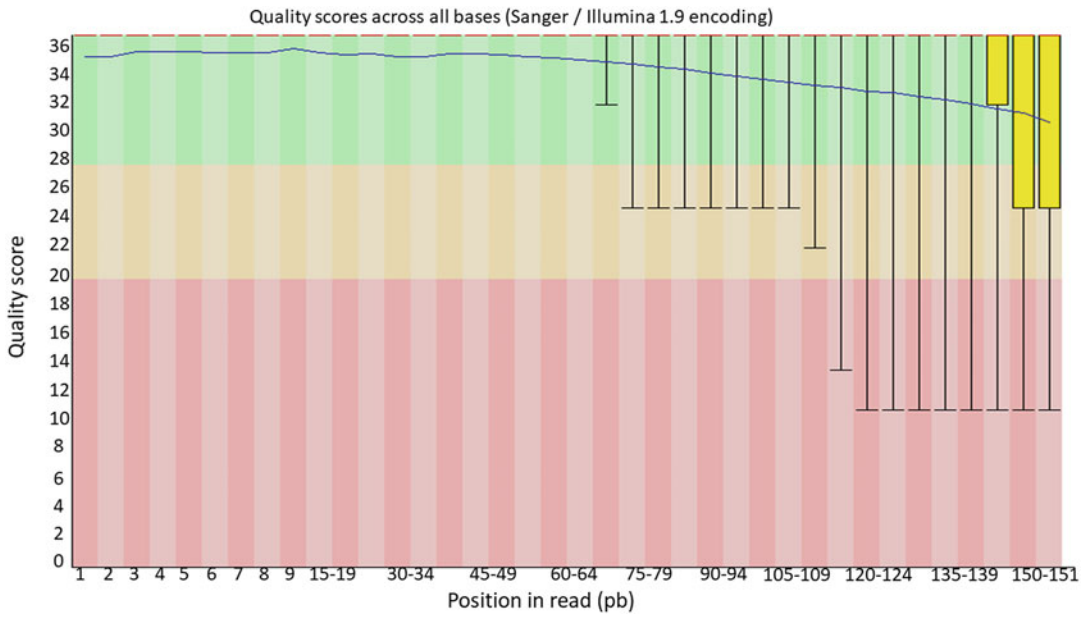


Fig. 2 Per base sequence quality output from FastQC from the SRR11131029_2.fastq file

```
$ cd~/Documents/results/meta-project/fastqc/
$ for filename in *.zip; do unzip $filename; done
$ ls

SRR11131028_1_fastqc SRR11131029_2_fastqc
SRR11131028_1_fastqc.html SRR11131029_2_fastqc.html
SRR11131028_1_fastqc.zip SRR11131029_2_fastqc.zip
SRR11131028_2_fastqc SRR11131030_1_fastqc
SRR11131028_2_fastqc.html SRR11131030_1_fastqc.html
SRR11131028_2_fastqc.zip SRR11131030_1_fastqc.zip
SRR11131029_1_fastqc SRR11131030_2_fastqc
SRR11131029_1_fastqc.html SRR11131030_2_fastqc.html
SRR11131029_1_fastqc.zip SRR11131030_2_fastqc.zip
```

The "unzip" command created new folders for each "zip" file. Inside, there are files concerning the process ("summary" and "_data.txt"), a copy of the ".html" report, and files used to generate this report.

```
$ cd SRR11131028_1_fastqc
$ ls

Icons Images fastqc.fo fastqc_data.txt fastqc_report.html sum-
mary.txt
```

3.3 Trimming and Filtering

With the per-base quality graphs generated with "FastQC," the distribution of the quality of each base across all the reads from the samples was visualized. This helps to establish quality thresholds, which will be used to remove low-quality sequences to reduce the false-positive rates due to sequencing error. Return to the folder where the raw reads are located:

```
$ cd ~/Documents/meta-project/raw-reads/
```

Here, "Trimmomatic" is used to remove the undesired low-quality data. As observed in the report from FastQC, the minimum expected length for this data will be set at 35 bases. Also, because the quality distribution was high, the quality threshold will be stringent. This will be fulfilled by the next command (*see* **Notes 6** and **7**):

```
$ for file in *_1.fastq; do base=$(basename ${file} _1.fastq)
 trimmomatic PE ${file} ${base}_2.fastq \
 ${base}_1.trim.fastq ${base}_1.unpair.fastq \
 ${base}_2.trim.fastq ${base}_2.unpair.fastq \
 SLIDINGWINDOW:4:20 MINLEN:35
 done
```

For each pair of files, "Trimmomatic" will print a message as follows, showing the beginning of the process, the number of paired-reads that survived, the unpaired that also surpass the threshold (unpaired reads happen when one strand does not survive the trimming process), and the ones that do not.

```
TrimmomaticPE: Started with arguments:
SRR11131028_1.fastq SRR11131028_2.fastq SRR11131028_1.trim.
fastq SRR11131028_1.unpair.fastq SRR11131028_2.trim.fastq
SRR11131028_2.unpair.fastq SLIDINGWINDOW:4:20 MINLEN:35
Quality encoding detected as phred33
Input Read Pairs: 23863111 Both Surviving: 22306594 (93.48%)
Forward Only Surviving: 976011 (4.09%) Reverse Only Surviving:
262847 (1.10%) Dropped: 317659 (1.33%)
TrimmomaticPE: Completed successfully
```

Two new files per sample file have been generated by "Trimmomatic."

```
SRR11131028_1.trim.fastq SRR11131028_2.trim.fastq
SRR11131029_1.trim.fastq SRR11131030_1.trim.fastq
SRR11131029_2.trim.fastq SRR11131030_2.trim.fastq
SRR11131028_1.unpair.fastq SRR11131029_1.unpair.fastq
SRR11131030_1.unpair.fastq SRR11131028_2.unpair.fastq
SRR11131029_2.unpair.fastq SRR11131030_2.unpair.fastq
```

With the trimming process, information will be lost, depending on the quality of the original data. So we need to consider the trade-off between having good quality and not losing data (Fig. 3). A good equilibrium is to maintain at least 70 or 80% of the original data after the trimming (*see* **Note 8**).

Move the new files to their respective folders inside the "results" location.

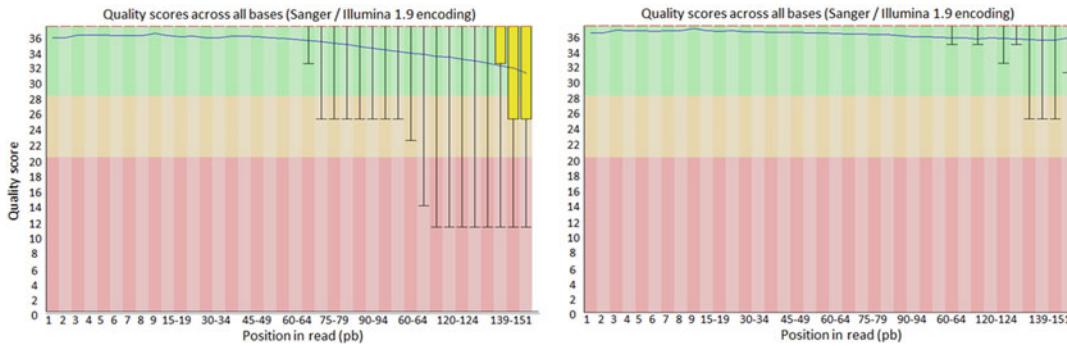


Fig. 3 Comparison of FastQC per base sequence quality output before(left) and after(right) trimming

```
mv *.trim.fastq ~/Documents/meta-project/results/trimmed-reads
mv *.unpair.fastq ~/Documents/meta-project/results/unpaired-reads
```

3.4 Metagenome Assembly

To obtain a metagenomic assembly, the reads will be assembled into contigs and these contigs into scaffolds. Navigate to the location where the trimmed data was stored after the trimming.

```
$ cd ~/Documents/meta-project/results/trimmed-reads
$ ls

SRR11131028_1.trim.fastq SRR11131028_2.trim.fastq
SRR11131029_1.trim.fastq SRR11131030_1.trim.fastq
SRR11131029_2.trim.fastq SRR11131030_2.trim.fastq
```

The software "metaSPAdes" has been demonstrated to have the best performance in assembling metagenomes from different types of samples [5], so we use it in this pipeline. Run the next command to perform the assembly with each pair of files from each sample.

```
$ for file in *_1.trim.fastq; do base=$(basename ${file} _1.trim.fastq)
 do ; metaspades.py -1 ${file} \
 -2 ${base}_2.trim.fastq \
  -o ~/Documents/meta-project/results/assemblies/assembly-${base}
 done
```

When "metaSPAdes" has finished, this message will be printed on the screen.

```
======= SPAdes pipeline finished.

SPAdes log can be found here: /Documents/meta-project/results/assemblies/assembly-SRR11131030/spades.log

Thank you for using SPAdes!
```

The output folder of each assembly will have the next set of files and folders (*see* **Note 9**).

```
$ cd ~/Documents/meta-project/results/assemblies
$ ls -F assembly-SRR11131028/
```

```
assembly_graph_after_simplification.gfa
assembly_graph.fastg
assembly_graph_with_scaffolds.gfa
before_rr.fasta
contigs.paths
corrected/
dataset.info
first_pe_contigs.fasta
input_dataset.yaml
contigs.fasta
scaffolds.fasta
K21/
K33/
K55/
misc/
params.txt
pipeline_state
run_spades.sh
run_spades.yaml
scaffolds.paths
spades.log
strain_graph.gfa
tmp/
```

The file `scaffolds.fasta`, is where the assembly of each sample is located. Use the next lines of code to rename and move these files from each folder.

```
$ ls | while read line; do base=$(echo $line | cut -d'-' -f2) \
 cp $line/scaffolds.fasta $base-scaffolds.fasta
 done
$ ls

SRR11131028-scaffolds.fasta assembly-SRR11131028
SRR11131029-scaffolds.fasta assembly-SRR11131029
SRR11131030-scaffolds.fasta assembly-SRR11131030
```

3.5 Metagenome Binning

The next step is the binning of the assembled scaffolds to obtain metagenome-assembled genomes (MAGs) (*see* **Note 10**). Make sure that the actual working directory is where the assemblies are:

```
$ cd ~/Documents/meta-project/results/assemblies
```

And create the folder where the output from the binning process will be located.

```
$ mkdir SRR11131028-maxbin
```

Now, run 'MaxBin' on one of the assemblies (*see* **Note 11**):

```
$ run_MaxBin.pl -thread 12 -contig SRR11131028-scaffolds.fasta
\
  -reads  ~/Documents/meta-project/results/trimmed_fastq/
SRR11131028_1.trim.fastq \
  -reads2  ~/Documents/meta-project/results/trimmed_fastq/
SRR11131028_2.trim.fastq \
 -out SRR11131028-maxbin/SRR11131028
```

When "MaxBin" has finished, the new folder will contain the next set of files (*see* **Note 12**).

```
$ ls SRR11131028-maxbin/

SRR11131028-MaxBin.001.fasta SRR11131028-MaxBin.log
SRR11131028-MaxBin.002.fasta SRR11131028-MaxBin.marker
SRR11131028-MaxBin.003.fasta  SRR11131028-MaxBin.marker_o-
f_each_bin.tar.gz
SRR11131028-MaxBin.004.fasta SRR11131028-MaxBin.noclass
SRR11131028-MaxBin.005.fasta SRR11131028-MaxBin.summary
SRR11131028-MaxBin.abund1 SRR11131028-MaxBin.tooshort
```

Inside the ".summary" file, a summary of each of the bins is located.

```
cat SRR11131028-maxbin/*.summary

Bin name Abundance Completeness Genome size GC content
SRR11131028-MaxBin.001.fasta 5196.80 16.8% 648896 42.6
SRR11131028-MaxBin.002.fasta 66.86 29.9% 4707236 42.9
SRR11131028-MaxBin.003.fasta 14.10 50.5% 5072656 50.0
SRR11131028-MaxBin.004.fasta 12.85 50.5% 22375323 41.7
SRR11131028-MaxBin.005.fasta 10.02 71.0% 6172746 53.9
```

Next, the quality of the created bins needs to be assessed. Create a folder to locate the output from "CheckM," which will be the program to use.

```
$ mkdir SRR11131028-checkm
```

And run the lineage workflow of "CheckM" with the next line of code (*see* **Note 13**).

```
$ checkm lineage_wf \
 -x fasta SRR11131028-maxbin/ SRR11131028-checkm -t 12
-------------------------------------------------------------------------------
 Bin Id Marker lineage # genomes # markers # marker sets 0 1 2 3 4 5+ Completeness
```

```
Contamination Strain heterogeneity
-------------------------------------------------------------------------------
 SRR11131028-MaxBin.005 f__Enterobacteriaceae (UID5054) 223 874 303 154 383 276 52 9 0
83.62 50.03 48.15
 SRR11131028-MaxBin.004 o__Flavobacteriales (UID2815) 123 324 204 137 110 53 20 2 2
54.71 35.26 1.58
 SRR11131028-MaxBin.003 k__Bacteria (UID3187) 2258 181 110 95 65 14 5 1 1 41.13 15.55
7.14
 SRR11131028-MaxBin.002 f__Flavobacteriaceae (UID2817) 81 511 283 438 70 3 0 0 0 9.46
0.20 0.00
 SRR11131028-MaxBin.001 k__Bacteria (UID1453) 901 171 117 146 20 4 1 0 0 4.61 2.01
28.57
-------------------------------------------------------------------------------
```

In this example, five MAGs were generated from sample SRR11131028. If the quality is high enough (high completeness [>90%] and low contamination [<5%]) these genomes can be used to search for metabolic screening, evaluation of selection, and phylogenetic studies with other lineages [11, 12] (*see* **Note 14**).

3.6 Taxonomic Assignment

The taxonomic assignment process will take place inside the "taxonomy" folder.

```
$ cd ~/Documents/meta-project/results/taxonomy
```

First, a database must be created before the taxonomic assignment of the reads. With the next lines, download and build the standard-Kraken database.

```
$ kraken2-build --standard --db $kraken-db
```

After database creation, move to the folder where the trimmed reads are located as they are needed by "Kraken2".

```
$ cd ~/Documents/meta-project/results/trimmed-reads
```

Now, perform the taxonomic assignment using "Kraken2" with the next command.

```
$ for file in *_1.trim.fastq; do base=$(basename ${file} _1.
trim.fastq)
 mkdir ~/Documents/meta-project/results/taxonomy/$base
  kraken2  --db  ~/Documents/meta-project/results/taxonomy/kra-
ken-db \
 --paired --fastq-input $file ${base}_2.trim.fastq \
  --output  ~/Documents/meta-project/results/taxonomy/$base/
```

```
$base.kraken \
  --report  ~/Documents/meta-project/results/taxonomy/$base/
$base.report
 done
```

At the end of the process, the ".kraken" and ".report" files will be generated. Inside the ".report" files there is a readable presentation of the taxonomic assignment (*see* **Note 15**).

```
$ cd ~/Documents/meta-project/results/taxonomy
$ head SRR11131028/SRR11131028.report

 94.25 21023116 21023116 U 0 unclassified
 5.75 1283478 213 R 1 root
 5.72 1276346 0 R1 131567 cellular organisms
 5.72 1276346 34580 D 2 Bacteria
 2.59 578382 15306 P 1224 Proteobacteria
 2.11 470599 4689 C 1236 Gammaproteobacteria
 1.64 366885 3706 O 91347 Enterobacterales
 1.28 285798 605 F 1903409 Erwiniaceae
 1.26 282119 4195 G 53335 Pantoea
 0.70 155982 134588 S 553 Pantoea ananatis
```

3.7 Diversity Tackled with R

Before entering into R, a ".biom" object needs to be created. This object will hoard the taxonomic assignment data generated with "kraken2". "kraken-biom" is an algorithm that uses the ".report" outputs to generate the needed file. Move to where the reports are located and use "kraken-biom" (*see* **Note 16**):

```
$ cd ~/Documents/meta-project/results/taxonomy
$ kraken-biom SRR11131028/SRR11131028.report \
 SRR11131029/SRR11131029.report \
 SRR11131030/SRR11131030.report --fmt json -o taxonomy.biom
$ ls
SRR11131028.report SRR11131029.report
SRR11131030.report taxonomy.biom
```

With these files, open "RStudio" to begin the taxonomic analysis (*see* **Note 17**). Create a new project located in the folder "taxonomy" to have easy access to the abovementioned files. Then create a new *R Script* inside the *File* window, *New File* option. The next lines of code will be written inside this script (*see* **Note 18**).

Packages in R are like lab equipment. The lab itself (R Studio) has its utilities and tools, but if a certain analysis is needed, one can acquire new equipment, such as an electrophoresis chamber or

thermocycler (installing packages), and expand the analysis capability of the entire lab. Install the needed packages by the next commands.

```
> install.packages("phyloseq")
> install.packages("ggplot2")
```

Packages need not be installed again (certain updates could be required in the future), but every time this or other scripts that use this code needs to be run, they must be called to the "R" environment, this would be like taking the thermocycler from its shelf and put it in the current working space (*see* **Note 19**).

```
> library("phyloseq")
> library("ggplot2")
```

Use the command "import-biom()" inside "phyloseq" to load the taxonomic information from the "taxonomy.biom" file in a new R-object called "merged_metagenomes".

```
> merged_metagenomes <- import_biom("taxonomy/taxonomy.biom")
> merged_metagenomes

phyloseq-class experiment-level object
otu_table() OTU Table: [ 4129 taxa and 3 samples ]
tax_table() Taxonomy Table: [ 4129 taxa by 7 taxonomic ranks ]
```

Inside the new object, there is the taxonomic information from the three analyzed samples. Next, use the "unique()" command to explore inside the "tax_table" section of "merged_metagenomes", the different phyla that "Kraken2" identified (*see* **Note 20**).

```
> unique(merged_metagenomes@tax_table@.Data[,2])
[1] "p__Proteobacteria"
[2] "p__Bacteroidetes"
[3] "p__Chlorobi"
[4] "p__Ignavibacteriae"
[5] "p__Candidatus Cloacimonetes"
[6] "p__Gemmatimonadetes"
[7] "p__Firmicutes"
```

With the "View()" command, a new window with the information inside the "tax_table" would be displayed (Fig. 4).

```
> View(merged_metagenomes@tax_table@.Data)
```

	Rank1	Rank2	Rank3	Rank4	Rank5	Rank6	Rank7
91347	k__Bacteria	p__Proteobacteria	c__Gammaproteobacteria	o__Enterobacterales	f__	g__	s__
1903409	k__Bacteria	p__Proteobacteria	c__Gammaproteobacteria	o__Enterobacterales	f__Erwiniaceae	g__	s__
53335	k__Bacteria	p__Proteobacteria	c__Gammaproteobacteria	o__Enterobacterales	f__Erwiniaceae	g__Pantoea	s__
553	k__Bacteria	p__Proteobacteria	c__Gammaproteobacteria	o__Enterobacterales	f__Erwiniaceae	g__Pantoea	s__ananatis
470934	k__Bacteria	p__Proteobacteria	c__Gammaproteobacteria	o__Enterobacterales	f__Erwiniaceae	g__Pantoea	s__vagans

Fig. 4 "tax_table" information with unnecessary characters and uninformative rank names

	Kingdom	Phylum	Class	Order	Family	Genus	Species
91347	Bacteria	Proteobacteria	Gammaproteobacteria	Enterobacterales			
1903409	Bacteria	Proteobacteria	Gammaproteobacteria	Enterobacterales	Erwiniaceae		
53335	Bacteria	Proteobacteria	Gammaproteobacteria	Enterobacterales	Erwiniaceae	Pantoea	
553	Bacteria	Proteobacteria	Gammaproteobacteria	Enterobacterales	Erwiniaceae	Pantoea	ananatis
470934	Bacteria	Proteobacteria	Gammaproteobacteria	Enterobacterales	Erwiniaceae	Pantoea	vagans

Fig. 5 New "tax_table" with trimmed information

Each taxonomic rank name has four unnecessary characters at the beginning. Moreover, the column (rank) names do not give useful information. To trim the data, use the next piece of code (Fig. 5).

```
> merged_metagenomes@tax_table@.Data <-
 substring(merged_metagenomes@tax_table@.Data, 4)
> colnames(merged_metagenomes@tax_table@.Data)<- c("Kingdom",
"Phylum",
 "Class", "Order", "Family", "Genus", "Species")
> View(merged_metagenomes@tax_table@.Data)

> unique(merged_metagenomes@tax_table@.Data[,"Phylum"])

[1] "Proteobacteria" "Bacteroidetes"
[3] "Chlorobi" "Ignavibacteriae"
[5] "Candidatus Cloacimonetes" "Gemmatimonadetes"
[7] "Firmicutes" "Cyanobacteria"
[9] "Actinobacteria" "Tenericutes"
[11] "Chloroflexi" "Deinococcus-Thermus"
[13] "Armatimonadetes" "Coprothermobacterota"
```

Use the "sample_sums()" command to display the number of classified reads inside "merged_metagenomes".

```
> sample_sums(merged_metagenomes)

SRR11131028 SRR11131029 SRR11131030
 1187468 31006410 2187930
```

Furthermore, by the "summary()" function, some statistical summary can be obtained from the "otu_table" of the phyloseq-object.

```
> summary(merged_metagenomes@otu_table@.Data)

 SRR11131028 SRR11131029 SRR11131030
 Min. : 0.0 Min. : 0 Min. : 0.0
 1st Qu.: 2.0 1st Qu.: 0 1st Qu.: 4.0
 Median : 7.0 Median : 0 Median : 14.0
 Mean : 287.6 Mean : 7509 Mean : 529.9
 3rd Qu.: 21.0 3rd Qu.: 0 3rd Qu.: 49.0
 Max. :284785.0 Max. :13620772 Max. :665847.0
```

As the information is trimmed, it is possible to search for information at a certain taxonomic level. The next example is used to find out how many OTUs have been classified as belonging to the Firmicutes phylum.

```
> sum(merged_metagenomes@tax_table@.Data[,"Phylum"] == "Firmicutes")

[1] 643
```

Mitochondrial, chloroplast, and virus identified reads have this information at the "Family," "Class," and "Kingdom" level, respectively. Search if this is the case for this set of data by the next lines of code.

```
> sum(merged_metagenomes@tax_table@.Data[,"Kingdom"] != "Bacteria")
[1] 0
> sum(merged_metagenomes@tax_table@.Data[,"Family"] == "mitochondria")
[1] 0
> sum(merged_metagenomes@tax_table@.Data[,"Class"] == "Chloroplast")
[1] 0
```

There is no presence of these undesired reads in our data, but if needed the next piece of code can be used to filter them (*see* **Note 21**).

```
merged_metagenomes <- subset_taxa(merged_metagenomes,
 Kingdom == "Bacteria" &
 Family != "mitochondria" &
 Class != "Chloroplast")
```

This data is ready to obtain alpha-diversity indexes and plot them. The "plot_richness()" command inside "phyloseq" will produce this desired output (Fig. 6).

```
> plot_richness(physeq = merged_metagenomes,
 measures = c("Observed","Chao1","Shannon"))
```

Use the command "head()" over the "otu_table" data to see that the information is expressed in absolute abundance (total number of reads in each sample assigned to an OTU).

```
> head(merged_metagenomes@otu_table@.Data)

 SRR11131028 SRR11131029 SRR11131030
91347 3706 10 528
1903409 605 0 29
53335 4195 0 22
553 155982 2 163
470934 108834 0 1114
549 10455 0 50
```

To compare the abundances of the OTUs between samples, and to obtain beta-diversity indexes, the data must be transformed to relative abundances. This can be made with the "transform_sample_counts()" function from "phyloseq."

```
> percentages = transform_sample_counts(merged_metagenomes,
 function(x) x*100 / sum(x) )
> head(merged_metagenomes@otu_table@.Data)

 SRR11131028 SRR11131029 SRR11131030
91347 0.31209262 3.225140e-05 0.024132399
1903409 0.05094874 0.000000e+00 0.001325454
53335 0.35327268 0.000000e+00 0.001005517
553 13.13568029 6.450279e-06 0.007449964
470934 9.16521540 0.000000e+00 0.050915706
549 0.88044478 0.000000e+00 0.002285265
```

There are different distance methods to obtain the distance between the relative abundance matrices to obtain beta-diversity. To enlist which of them are part of the "phyloseq" library, use the command "distanceMethodList."

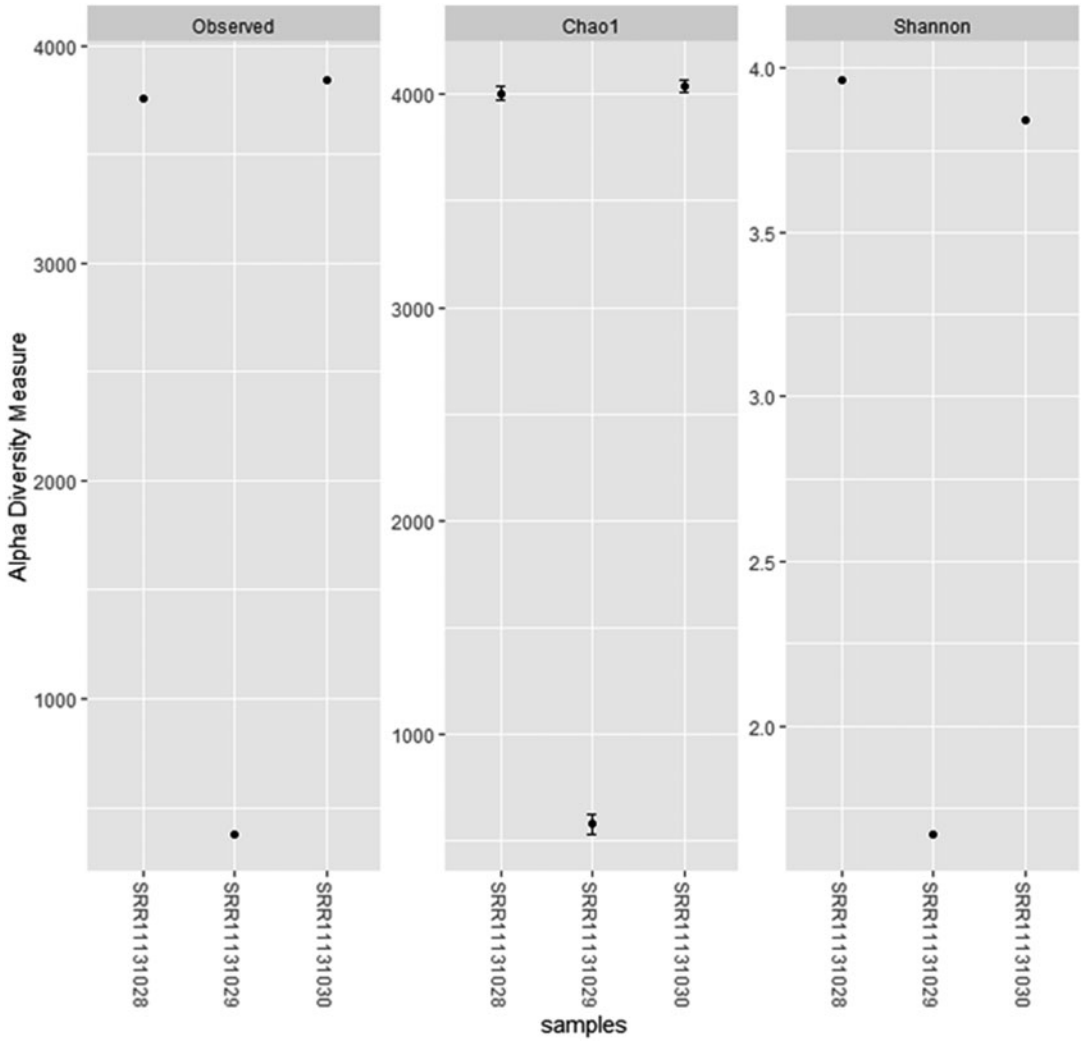


Fig. 6 Alpha diversity of the 3 samples with three different metrics(from left to right): Observed, Chao1, and Shannon

```
> distanceMethodList

$UniFrac
[1] "unifrac" "wunifrac"

$DPCoA
[1] "dpcoa"

$JSD
[1] "jsd"

$vegdist
 [1] "manhattan" "euclidean" "canberra" "bray" "kulczynski"
```

```
"jaccard" "gower"
 [8] "altGower" "morisita" "horn" "mountford" "raup" "bino-
mial" "chao"
[15] "cao"

$betadiver
 [1] "w" "-1" "c" "wb" "r" "I" "e" "t" "me" "j" "sor" "m" "-2"
"co" "cc" "g"
[17] "-3" "l" "19" "hk" "rlb" "sim" "gl" "z"

$dist
[1] "maximum" "binary" "minkowski"

$designdist

[1] "ANY"
```

In this example, NMDS with Bray-Curtis distance [10] will be used to obtain the beta-diversity. The "ordinate()" is used for this purpose (*see* **Note 22**).

```
> meta.ord <- ordinate(physeq = percentages, method = "NMDS",
 distance = "bray")
```

The result can be plotted using the "plot_ordination()" command as follows (Fig. 7).

```
> plot_ordination(physeq = percentages, ordination = meta.ord)
```

3.8 A Focused Taxonomic Exploration

It is possible to take a taxonomic rank and explore the diversity inside it within all the samples. First, agglomeration of the data at the desired taxonomic level is needed; (Fig. 8) this can be made by "tax_glom()".

```
> glom <- tax_glom(percentages, taxrank = 'Phylum')
> View(glom@tax_table@.Data)
```

With the abundance information concatenated at phylum level, a "data.frame" object with the information inside the "percentage" object will be created using the "psmelt()" command (*see* **Note 23**).

```
> percentages <- psmelt(glom)
> str(percentages)
```

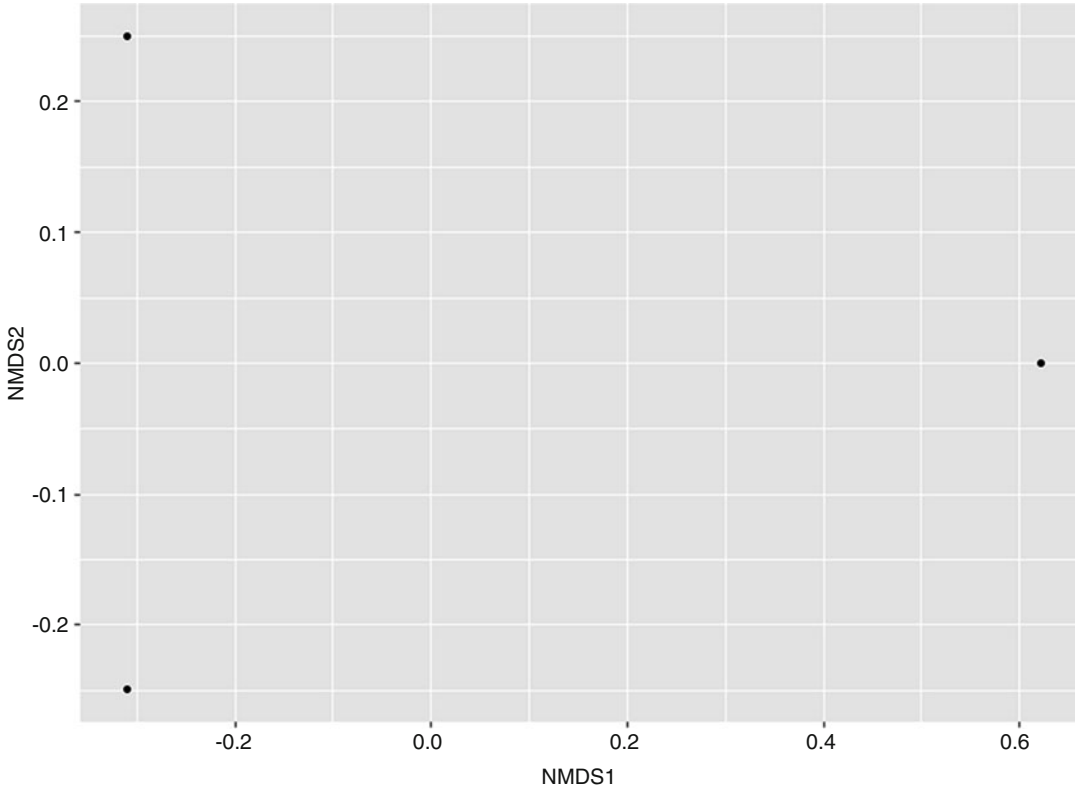


Fig. 7 Beta diversity with NMDS. Each of the points represents a different sample

	Kingdom	Phylum	Class	Order	Family	Genus	Species
2067572	Bacteria	Proteobacteria	*NA*	*NA*	*NA*	*NA*	*NA*
171549	Bacteria	Bacteroidetes	*NA*	*NA*	*NA*	*NA*	*NA*
1096	Bacteria	Chlorobi	*NA*	*NA*	*NA*	*NA*	*NA*
1134405	Bacteria	Ignavibacteriae	*NA*	*NA*	*NA*	*NA*	*NA*
456827	Bacteria	Candidatus Cloacimonetes	*NA*	*NA*	*NA*	*NA*	*NA*
173480	Bacteria	Gemmatimonadetes	*NA*	*NA*	*NA*	*NA*	*NA*
1491	Bacteria	Firmicutes	*NA*	*NA*	*NA*	*NA*	*NA*

Fig. 8 Taxonomy table inside the phyloseq object after the "tax_glom()" command. It is observed that the abundance has been agglomerated at the phylum level, leaving the inferior taxonomic levels empty

```
'data.frame': 108 obs. of 5 variables:
 $ OTU : chr "2067572" "2067572" "2067572" "171549" ...
 $ Sample : chr "SRR11131030" "SRR11131029" "SRR11131028"
"SRR11131029" ...
 $ Abundance: num 75.4 54.1 46.5 44.2 28.8 ...
 $ Kingdom : chr "Bacteria" "Bacteria" "Bacteria" "Bacteria"
```

```
...
 $ Phylum : chr "Proteobacteria" "Proteobacteria" "Proteobac-
teria" "Bacteroidetes" ...
```

With this new object, use "ggplot2" to generate a bar-plot that displays the abundance of the different phyla of each sample (Fig. 9).

```
> rel.plot <- ggplot(data=percentages,
 aes(x=Sample, y=Abundance, fill=Phylum))+
 geom_bar(aes(), stat="identity", position="stack")
> rel.plot
```

The myriad of colors represented in the above figure makes it difficult to distinguish which phyla are represented by which colors in the bars. Rename the minority phyla in a new category which will include all the OTUs with less than 0.5% (*see* **Note 24**) of relative abundance to have fewer colors in the plot.

```
> percentages$Phylum[percentages$Abundance < 0.5] <- "Phyla <
0.5% abund."
> unique(percentages$Phylum)
[1] "Proteobacteria" "Bacteroidetes"
[3] "Firmicutes" "Cyanobacteria"
[5] "Actinobacteria" "Tenericutes"
[7] "Phyla < 0.5% abund."
```

Finally, a distinguishable set of colors is presented in the next plot (Fig. 10):

```
> rel.plot <- ggplot(data=percentages,
 aes(x=Sample, y=Abundance, fill=Phylum))+
 geom_bar(aes(), stat="identity", position="stack")
> rel.plot
```

With this pipeline and using the tools in "phyloseq" and "ggplot2", it is possible to obtain results that can support/refute hypotheses or generate new ones. Furthermore, the code in this protocol ends with some publication-quality plots, which is expected to help future learners and scientists deliver helpful information when needed. Finally, all the outputs obtained in this pipeline can be submitted to other workflows and analyses to bring more beneficial information regarding the samples of interest.

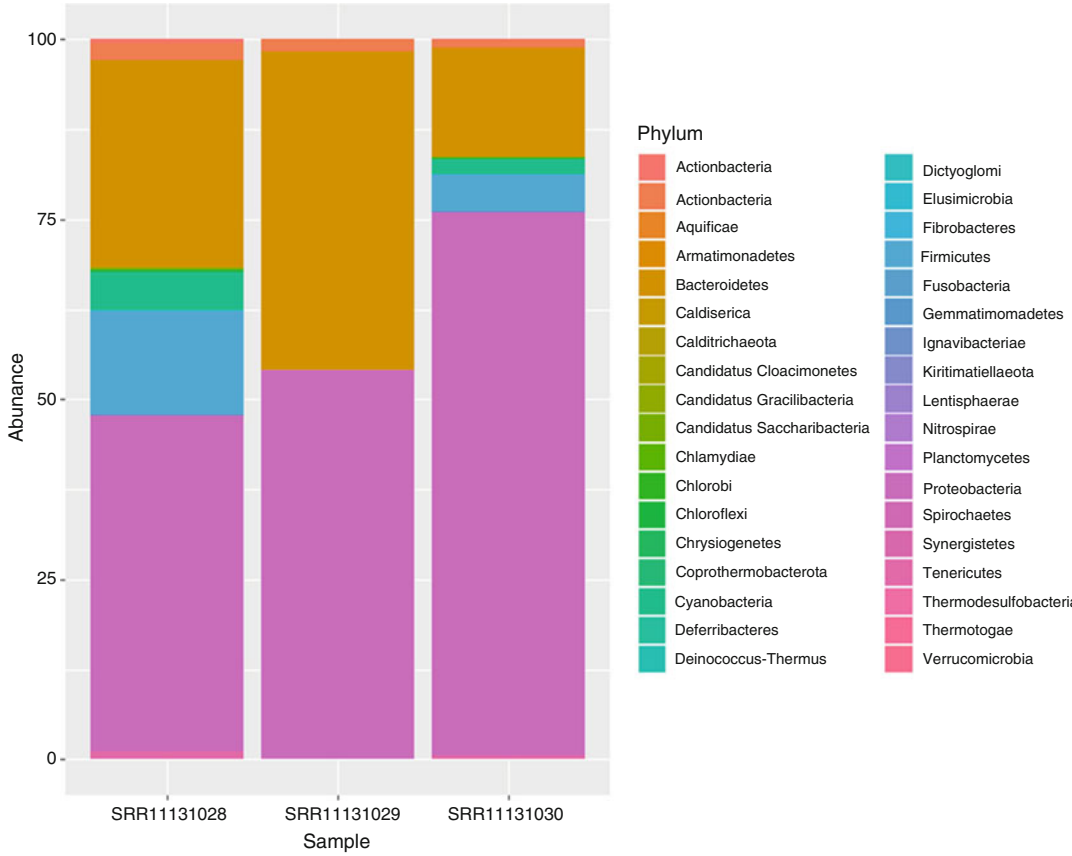


Fig. 9 Bar plots of the relative abundance at the phylum level of the three samples. Each color marks a different phylum, but metagenomic analyses usually have more elements than colors differentiable by the human eye

4 Notes

1. A virtual machine is available for rent at AWS ami-0e7fb76a881ab5e09 (Metagenomics—18 March [The Carpentries Incubator]).
2. All the Bash software was installed inside a Conda environment, which is useful to maintain dependencies in order. Is advised to install all the needed software at the same time in order to avoid compatibility issues.
3. A brief Linux introduction is available in the lesson "Introduction to the command line" part of the "Metagenomics Workshop Overview" (https://carpentries-incubator.github.io/metagenomics-workshop/). Users will need at least this Linux knowledge through this lesson.
4. The flag "--split-files" is to separate the forward and reverse reads in different files, and the "-I" flag is used to add a "_1" and "_2" to the files with the forward and reverse files,

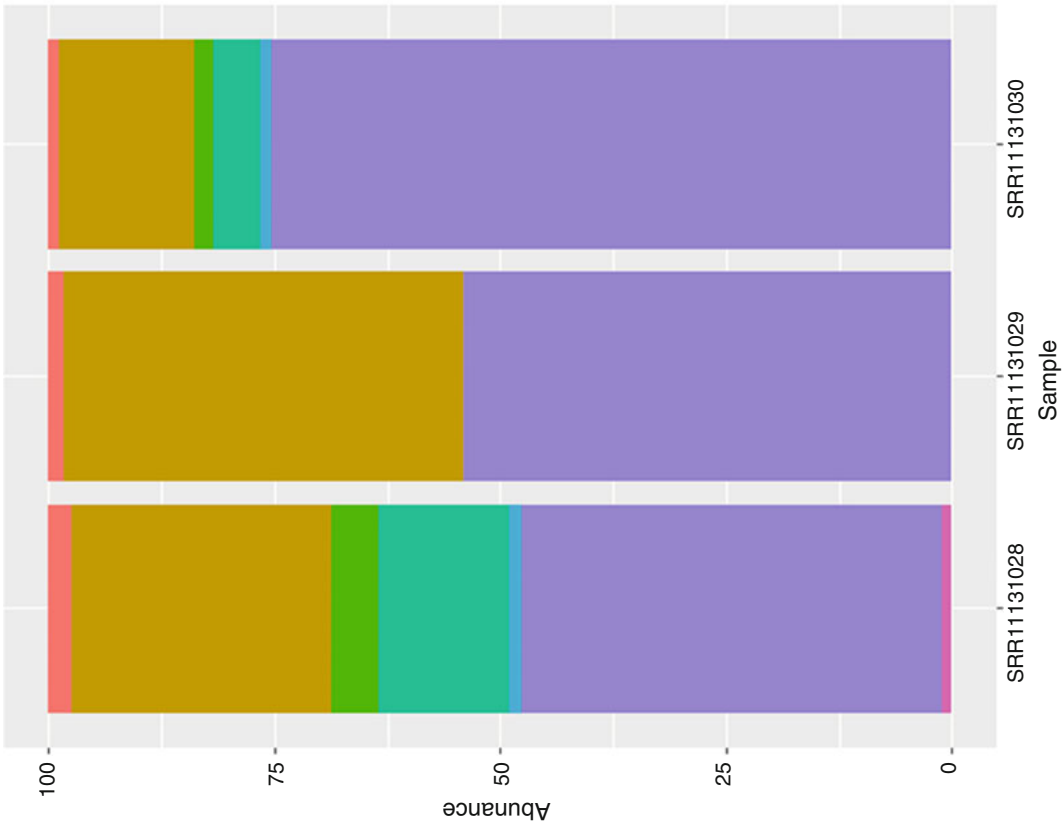


Fig. 10 To visualize relevant phyla, bar plots with the thresholded relative abundance inside the three samples of the principal phyla are shown. After rearranging the phyla categories using only those with abundances $>0.5\%$, the data is easier to visualize

respectively. This option must not be added if the user is working with single-end reads.

5. Different compression formats use different compression commands that work differently: gunzip can accept a list of compressed files with a wildcard, but unzip must be used within a loop to decompress more than one file at a time.
6. In the next line of code, the "\" is used to split a big line of code. Bash will continue reading the next line as part of the same code sentence. Is important to write it after a space, and begin the next line with no spaces at all. One code sentence can be split in each blank space as desired. Note that it is not equivalent to ";", which splits different code sentences when written in the same line.
7. In this loop the "basename" command is used to make a variable ("base") that contains the ID of the samples, so we can call the "_2.fastq" files and not only the "_1.fastq" files that are already in the variable "file," and also to add the ID to the names of the new files that Trimmomatic will create.
8. After the trimming process, if the data does not reach the 70–80% recommendation different actions can be taken according to the question the data wants to answer. If taxonomic identification is important, an inspection of the unpaired reads is advised. But for the assembly, the user can reduce the quality threshold to obtain more reads, putting special attention to false negatives and false positives in the interpretation of the results.
9. Most programs will give a lot of output because they make available the intermediate results of the algorithm and some extra details, alongside the expected output (the assembly in this case). They can be useful if a detailed exploration of the data processing is needed. For example, the "assembly_graph.fastg" can be used to visualize the assembly with Bandage (https://github.com/rrwick/Bandage).
10. The term "bins" is used to refer technically to the groups of scaffolds, we call them MAGs when we consider them genomes.
11. Most programs have a mandatory output argument ("-out" in MaxBin). In some cases, it refers to the directory where you want your output located (like in metaSPAdes), but it can also refer to a prefix that will be added to the names of the output files along with the directory path. In the case of MaxBin the user needs to specify this prefix, if only a directory path is given, the output files will be hidden since the separator it uses is a dot.
12. Alongside the bins, there are files with useful information for the user. ".log" displays the steps that the binning carried out;

".abund1" and "abund2" hoards the abundance of each of the reads in the forward and reverse files respectively; ".marker" displays the marker genes in each bin; inside the ".noclass" there are unbinned sequences; ".tooshort" holds the short sequences that are too short for being contigs/scaffolds.

13. The lineage workflow is recommended because it locates the MAG in a phylogenetic tree to know which genetic markers to use. It will use the taxonomic level that is more appropriate for each case. When the computing power or the time is a limiting factor the taxonomic workflow can be used instead. This workflow is also useful if you need to analyze every MAG with the same markers.
14. The level of completeness and contamination is sample-dependent. If a high contamination percentage is displayed, but the sample consists of undescribed lineages, consider using the result in the rest of the pipeline.
15. A detailed description of this and the other kraken2 outputs can be found on the GitHub page for this program: https://github.com/DerrickWood/kraken2/blob/master/docs/MANUAL.markdown
16. Kraken-biom can be used in the R terminal window as well. How the user orders the ".report" files in the kraken-biom command, is how they will be allocated in the ".biom" file. An order that can be changed later, but is advised to get it properly from the beginning.
17. R studio and its packages can experiment conflicts with versions. Updating the R studio software is recommended before starting. Also, the use of R online can surpass these issues.
18. The R command "getwd()" displays the actual working directory. If the location is not the desired one, it can be changed by using the "setwd()" command, specifying inside the parentheses the path to the desired folder. Also, each of the scripts of the project can begin with the specification of the desired working-directory.
19. Warnings (highlighted in red as errors), can be displayed when loading packages. This will not hamper the procedure as an error would, and the user can continue with the pipeline.
20. To move inside the phyloseq object, the "@" operator is used instead of "$" that is usual for "data.frame" exploration. Phyloseq objects are a special kind of object that resembles a list with matrices and data.frames inside.
21. Other taxonomic assignment programs can assign these OTUs to other taxonomic levels. A wide review of all the taxonomic levels with the specified code is advised.

22. Some distance methods do not allow the presence of zeros. So the trimming of the missing data can be done with the next command.

```
> prune_species(speciesSums(merged_metagenomes) > 0, merged_-
metagenomes)
```

23. The data type for each taxonomic level inside the data.frame, can be changed from character to other of the four R data types. This can be changed by the next command, only if needed.

```
> precentages$Phylum <- as.character(percentage$Phylum)
```

24. The threshold to obtain only eight or nine OTUs in each taxonomic level changes according to the data being analyzed. Exploration with different values for the threshold is advised to obtain the desired results.

Acknowledgments

The authors would like to thank all the learners, helpers, and instructors who gave feedback on the consolidation of this pipeline. To Ahmed Moustafa for his advice and support in the writing of this chapter. Finally, we thank the Mexican population, which supports the training of scientists within the country. The research reported in this publication was supported by CONACyT "MicroAgrobioma" grant 320237.

References

1. Levy A, Salas Gonzalez I, Mittelviefhaus M et al (2018) Genomic features of bacterial adaptation to plants. Nat Genet 50(1):138–150. https://doi.org/10.1038/s41588-017-0012-9
2. Andrews S (2010) FastQC: a quality control tool for high throughput sequence data. http://www.bioinformatics.babraham.ac.uk/projects/fastqc
3. Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 30(15): 2114–2120. https://doi.org/10.1093/bioinformatics/btu170
4. Nurk S, Meleshko D, Korobeynikov A et al (2017) MetaSPAdes: a new versatile metagenomic assembler. Genome Res 27(5): 824–834. https://doi.org/10.1101/gr.213959.116
5. Wu YW, Tang YH, Tringe SG et al (2014) MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. Microbiome 2(1):26. https://doi.org/10.1186/2049-2618-2-26
6. Parks DH, Imelfort M, Skennerton CT et al (2015) CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res 25(7):1043–1055. https://doi.org/10.1101/gr.186072.114
7. Wood DE, Salzberg SL (2014) Kraken: ultrafast metagenomic sequence classification using

exact alignments. Genome Biol 15(3):R46. https://doi.org/10.1186/gb-2014-15-3-r46

8. McMurdie PJ, Holmes S (2013) Phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. PLoS One 8(4):e61217. https://doi.org/10.1371/journal.pone.0061217
9. Fadiji AE, Ayangbenro AS, Babalola OO (2020) Organic farming enhances the diversity and community structure of endophytic archaea and fungi in maize plant: a shotgun approach. J Soil Sci Plant Nutr 20(4): 2587–2599. https://doi.org/10.1007/S42729-020-00324-9
10. Bray JR, Curtis JT (1957) An ordination of the upland forest communities of southern Wisconsin. Ecol Monogr 27(4):325–349. https://doi.org/10.2307/1942268
11. Weissman JL, Hou S, Fuhrman JA (2021) Estimating maximal microbial growth rates from cultures, metagenomes, and single cells via codon usage patterns. PNAS 118(12): e2016810118. https://doi.org/10.1073/pnas.2016810118
12. Wilkins LGE, Ettinger CL, Jospin G et al (2019) Metagenome-assembled genomes provide new insight into the microbial diversity of two thermal pools in Kamchatka, Russia. Sci Rep 9(1):1–15. https://doi.org/10.1038/s41598-019-39576-6

Chapter 11

Rhizosphere and Endosphere Bacterial Communities Survey by Metagenomics Approach

Victoria Mesa

Abstract

The diversity of microbes associated with plant roots is in the order of tens of thousands of species. It is estimated that only 0.1–1.0% of the living bacteria present in soils can be cultured under standard conditions. The microbial marker-gene sequence data and the next-generation sequencing technologies have enabled systemic studies of root-associated microbiomes. Molecular techniques can be used to generate comprehensive taxonomic profiles of the microorganisms present in roots. The aim of this chapter is to provide a standard method for the obtention of rhizosphere and endosphere fractions, and a generic workflow of the Quantitative Insights Into Microbial Ecology version 2 (QIIME2) software to analysis of 16S rRNA marker-gene.

Key words Endosphere, Rhizosphere, Microbial diversity, Next-generation sequencing, 16S rRNA gene, Bioinformatics

1 Introduction

The root-associated microbiome are critical for plant productivity, well-being, and performance of plants [1]. The shape of the microbial community is mediated by a large number of interactions between roots, microbiomes, and soil. The microbial community structure within the plant is dynamic and is mediated by abiotic and biotic factors such as species and genotype of host plants, soil conditions, biogeography, and microbe–microbe and plant–microbe interactions [2, 3].

The rhizosphere (i.e., the soil close to the root surface) and endosphere (i.e., all inner root tissues) are critical interfaces for the exchange of nutrients and microbes between soil and plant [2]. The microorganisms mediate nutrient uptake, stress tolerance and disease resistance; and some of them are especially important for plants during harsh and unfavorable growing conditions [4].

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_11,

The rhizobiome, populations of microorganisms in the rhizosphere, can directly affect nutrient availability to the plants benefiting by the root exudates carbon-rich substrates excreted by plants [5], whereas endophytic bacteria colonize the internal tissues of plants, can promote plant growth through mechanisms such as phosphate solubilization, indole-3-acetic acid (IAA) and siderophore production, and/or supplying essential vitamins to plants and can act as biocontrol agents [6].

The characterization of the plant-associated microbiome by microbial culture-dependent approaches is limited due to specific cultivation requirements, and their detection is only possible with the DNA-based techniques allowing sequencing their genetic fingerprint, the so-called metagenome. In recent years, next-generation sequencing technologies have enabled systemic studies of root-associated microbiomes. Microbial marker-gene surveys (e.g., bacterial/ archaeal 16S rRNA or eukaryotic 18S rRNA genes) can be used to identify composition of total microorganisms present in roots, through the massive sequencing.

To carry out this approach, an universal gene is isolated and amplified through the polymerase chain reaction (PCR). The amplified gene product is sequenced and the variation within the gene sequences profile microbiota with varying degrees of taxonomic specificity. The process of going from raw gene sequences to taxonomic profiles or diversity measures involves sequence quality checking, error rates of sequencing by denoising, taxonomic classification, alignment to reference microorganisms, and phylogenetic tree building.

In this chapter, we first present a protocol for dissecting the microbiota from various root compartments. Subsequently, a method for amplifying the 16S rRNA gene. Finally, we present a generic workflow of the Quantitative Insights Into Microbial Ecology version 2 (QIIME2) software to analysis the resulting sequencing data of 16S rRNA than allow to transform raw sequences into taxonomic bar plots, phylogenetic tree, principal coordinates analyses, and diversity measures.

2 Materials

1. Falcon 50 mL conical centrifuge tubes.
2. 1.5 mL microfuge tubes.
3. 0.2 mL PCR tubes.
4. Filtered pipette tips (10, 200, 1000 μL).
5. Nuclease-free water.
6. Ethanol.
7. 2% chloride solution.

8. 0.1% sodium pyrophosphate.
9. Tween 80.
10. 10 mM $MgSO_4$.
11. Dissection tools (scissors and forceps).
12. PowerSoil® DNA isolation kit.
13. Primer 515F (GTGCCAGCMGCCGCGGTAA).
14. Primer 806R (GGACTACHVGGGTWTCTAAT).
15. Phusion high-fidelity DNA polymerase.
16. 5× Phusion high-fidelity (HF) buffer.
17. QIAquick PCR purification kit.
18. Quant-iT PicoGreen dsDNA Assay Kit.
19. Fluostar Omega plate reader.
20. Agilent 2100 Bioanalyzer system.
21. Quant-iT PicoGreen dsDNA Assay Kit.
22. −80 °C freezer.
23. Microcentrifuge.
24. PCR thermal cycler.
25. QIIME version 2.

3 Methods

3.1 Collection of Root Endosphere, Rhizosphere, and Bulk Soil Fractions

The procedure outlined below is generally applicable across a wide array of conditions and has been successfully used to survey bacterial community composition in plant roots of contaminated industrial soils [7].

To collect bulk soil sample, use an ethanol-sterilized soil core collector to obtain soil that is free of plant roots by collecting a core approximately 20 to 30 cm from the base of the plant. Transfer the soil to a plastic bag, homogenize the soil by gentle shaking. Five grams of bulk soil is shaken in 10 mL of 0.1% sodium pyrophosphate. Soil particles are allowed to settle for 1 h. Supernatants are recovered to DNA Extraction.

To obtain rhizosphere fraction, vigorously shake the roots to remove loose soil, leaving only the soil layer firmly attached to the root. This layer constitutes the rhizosphere compartment. Five grams of rhizosphere soils is shaken in 10 mL of 0.1% sodium pyrophosphate. Soil particles are allowed to settle for 1 h. Supernatants are recovered to DNA Extraction.

To obtain the endosphere fraction, roots are rinsed under running tap water. Using flame-sterilized scissors, cut ~5 cm of root immediately below the root-shoot junction. Root samples are surface sterilized for 10 min in 2% active chloride solution

supplemented with one droplet of Tween 80 per 100 mL of solution and subsequently rinsed three times for 1 min in sterile distilled water. After surface sterilization, root samples are macerated in 10 mL of 10 mM $MgSO_4$ with a mortar. The liquid from the macerated was recovered and used for the subsequent extraction of DNA. Preprocessed roots should be stored for no longer than 24 h at 4 °C.

3.2 DNA Extraction of Recovered Fractions

Use the PowerSoil® DNA isolation kit to isolate the genomic DNA from the root-associated communities. For the bulk and rhizosphere soils, add 500 μL of the soil suspension generated in step 1 and 2 to PowerSoil Bead tubes. For the endosphere fraction, 500 μL of liquid from the macerated generated in step 3 is added to PowerSoil Bead tubes. After adding the samples to the PowerSoil Bead tubes, follow the PowerSoil® kit protocol instructions.

3.2.1 16S rRNA Amplification

The V4 region of the 16S rRNA gene is amplified using the primer set 515F (5′-GTGCCAGCMGCCGCGGTAA-3′) and 806R-(5-′-GGACTACHVGGGTWTCTAAT-3′) [8]. PCR amplicons are generated in 50 μL PCR reaction mixtures with the following conditions: 1 μM forward and reverse primers, 10 ng template DNA, 25 μL 5× Phusion high-fidelity (HF) buffer containing 200 μM each deoxynucleotide and 1.5 mM $MgCl_2$ in a master mix, 15 μL molecular biology grade water, and 1 unit (0.5 μL) Phusion high-fidelity DNA polymerase. Cycling conditions include initial denaturation at 94 °C for 3 min, followed by 30 cycles of denaturation at 94 °C for 45 s, annealing at 50 °C for 1 min, and extension at 72 °C for 1.5 min; and a final extension phase was conducted at 72 °C for 10 min. Amplify samples in triplicate. Prior to sequencing, amplicons from different plant samples were multiplexed by incorporating unique molecular identifier tags (MIDs) at the 5′ end of the reverse primer. To incorporate MIDs into the PCR amplicons, secondary PCR is run using similar reagents and PCR cycling conditions as in the primary PCR, with the amplification cycles reduced to 10. Additionally, it is important to run a negative control for each individual reaction to detect any potential contamination. PCR reactions are pooled and purified using the QIAquick PCR purification kit, the quality of the amplicon pools are evaluated using an Agilent 2100 Bioanalyzer system, and DNA concentration of amplicon libraries is determined using Quant-iT PicoGreen dsDNA Assay Kit and a Fluostar Omega plate reader and pooled in equimolar concentrations. Utilize the services of a sequencing facility to sequence the DNA on a NGS platform, 2 × 300 bp paired-end sequencing.

```
@ML-P2-14:9:000H003HG:1:11102:17290:1073 1:N:0:TCCTGAGC+GCGATCTA
TTTGGTAACAGCATGAATTATTCTAGCCACTAAAACTCTATGAACATCTTGTGAAGGTTTCAGATAGAGCCTGAAGTACACAGAGAACAATTCTTAAAAAA
+
AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE<AEEEEEEE
```

Fig. 1 Representation of FASTQ file content

3.3 Bioinformatic Analysis of 16S rRNA Amplicons

3.3.1 Sequence Data

This protocol requires a marker-gene data set generated from a 16S rRNA gene fragment and sequenced as paired-end reads on an Illumina platform. For a paired-end run, one R1 and one Read 2 (R2) FASTQ file is created for each sample for each lane. Sequence data should be in FASTQ format, this is the format in which Illumina (also Ion Torrent) sequencing data is delivered. FASTQ files includes DNA sequence information with quality metadata. FASTQ files are compressed and created with the extension (.fastq.gz). Each entry in a FASTQ files consists of the following 4 lines [9]:

- A sequence identifier with information about the sequencing run and the cluster. The exact contents of this line vary by based on the BCL to FASTQ conversion software used.
- The sequence (the base calls; A, C, T, G, and N).
- A separator, which is simply a plus (+) sign.
- The base call quality scores. These are Phred +33 encoded, using ASCII characters to represent the numerical quality scores (http://www.ascii-code.com/).

The Fig. 1 shows an example of a single entry in a R1 FASTQ file.

FASTQ sequences must be named using the Illumina naming convention. For example, a gzipcompressed FASTQ file may be called.

```
SampleName_S1_L001_R1_001.fastq.gz
SampleName_S1_L001_R2_001.fastq.gz
```

where SampleName → name of the sample, S1 → sample number on the sample sheet, L001 → lane number, R1/R2 → R1 indicates that the file contains the forward reads and R2 indicates that the file contains the reverse reads, and the last three numbers are always 001 by convention.

3.3.2 Quality of the Files

The quality of the files can be visualized by using the "FastQC" program (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/). FastQC aims to provide a simple way to do some quality control checks on raw sequence data coming from high throughput sequencing pipelines. It provides a modular set of analyses which you can use to give a quick impression of whether your data has any problems of which you should be aware before doing any further analysis.

3.3.3 Analysis of 16S rRNA Amplicons with QIIME2

QIIME2 (https://qiime2.org) is a microbiome bioinformatics platform with a completely reengineered and rewritten system that is expected to facilitate reproducible and modular analysis of microbiome data to enable the next generation of microbiome science [10]. QIIME 2 also incorporates a major advance, the use of exact "Sequence Variants" (SV) rather than "Operational Taxonomic Units" (OTU). The data processing is based on the following steps.

- Importing raw sequence (FASTQ) data into QIIME 2.
- Performing quality control of the sequences.
- Grouping of the sequences on the basis of similarity and clustering.
- Taxonomic assignation.
- Analysis of data.

(a) Installing QIIME2.

See the installing QIIME2 (https://docs.qiime2.org/2020.8/install/) using Conda in Mac and Linux environments.

(b) File types in QIIME2.

QIIME2 uses two different file types, *.qza* files are data files while *.qzv* files are graphic visualizations. All of QIIME2 files can be viewed using an online browser that is available at (https://view.qiime2.org).

(c) Sample metadata.

Sample metadata is stored in a tab-separated text file. Each row represents a sample, and each column represents a metadata category. The first line is a header that contains the metadata category names. The first column is used for sample names. QIIME2 host a browser-based metadata validation tool, Keemei (https://keemei.qiime2.org/), Google Sheets add-on for validating sample metadata [11]. The sample metadata is available as a Google Sheet. You can download this file as tab-separated text by selecting File > Download as > Tab-separated values. Alternatively, the following command will download the sample metadata as tab-separated text and save it in the file sample-metadata.tsv. This sample-metadata.tsv file is used throughout the rest of the tutorial (https://docs.qiime2.org/2020.8/tutorials/metadata/).

(d) Set up your working directory.

Initially, set up your working directory and change to that directory.

```
mkdir qiime2-tutorial
cd qiime2-tutorial
```

3.3.4 Importing Raw Sequence (FASTQ) Data into QIIME2

In QIIME2, there are functions to import different types of FASTQ data, for both single or pair-end sequences.

1. FASTQ data with the EMP (Earth Microbiome Project) Protocol format.
2. FASTQ data in the Casava 1.8 demultiplexed format.
3. Multiplexed FASTQ data with barcodes in sequences.
4. Any FASTQ data not represented in the list items above: " FASTQ manifest" formats.

If you have sequencing data with a specific formats (EMP or Casava), you can directly import the folder containing your sequences with the following commands.

(a) **"EMP protocol" multiplexed single-end FASTQ containing the associated barcode reads**

This format should have two fastq.gz files (one forward.fastq.gz file that contains the single-end reads, and one barcodes.fastq.gz file that contains the associated barcode reads).

```
qiime tools import \
--type EMPSingleEndSequences \
--input-path qiime2-tutorial/emp-single-end-sequences \
--output-path qiime2-tutorial/emp-single-end-sequences.qza
```

(b) **"EMP protocol" multiplexed paired-end FASTQ containing the associated barcode reads**

This format should have three fastq.gz files (one forward.fastq.gz file that contains the forward sequence reads, one reverse.fastq.gz file that contains the reverse sequence reads, and one barcodes.fastq.gz file that contains the associated barcode reads).

```
qiime tools import \
--type EMPPairedEndSequences \
--input-path qiime2-tutorial/emp-paired-end-sequences \
--output-path qiime2-tutorial/emp-paired-end-sequences.qza
```

(c) **Casava 1.8 single-end demultiplexed format.**

This format should have one fastq.gz file for each sample in the study. The file name includes the sample identifier and should look like L2S357_15_L001_R1_001.fastq.gz. The underscore-separated fields in this file name are the sample identifier, the barcode sequence or a barcode identifier, the lane number, the direction of the read (i.e., only R1, because these are single-end reads), and the set number.

```
qiime tools import \
--type 'SampleData[SequencesWithQuality]' \
```

```
--input-path qiime2-tutorial/casava-18-single-end-demulti-
plexed \
--input-format CasavaOneEightSingleLanePerSampleDirFmt \
--output-path qiime2-tutorial/demux-single-end.qza
```

(d) **Casava 1.8 paired-end demultiplexed format.**

This format should have two fastq.gz files for each sample in the study, each containing the forward or reverse reads for that sample. The file name includes the sample identifier. The forward and reverse read file names for a single sample might look like L2S357_15_L001_R1_001.fastq.gz and L2S357_15_L001_R2_001.fastq.gz, respectively. The underscore-separated fields in this file name are the sample identifier, the barcode sequence or a barcode identifier, the lane number, the direction of the read (i.e., R1 or R2), and the set number.

```
qiime tools import \
--type 'SampleData[PairedEndSequencesWithQuality]' \
--input-path qiime2-tutorial/casava-18-paired-end-demulti-
plexed \
--input-format CasavaOneEightSingleLanePerSampleDirFmt \
--output-path qiime2-tutorial/demux-paired-end.qza
```

3.3.5 Multiplexed FASTQ Data with Barcodes in Sequences

(a) **Multiplexed single-end FASTQ.**

This format should have one fastq.gz file, containing records from multiple samples and one metadata file with a column of per-sample barcodes for use in FASTQ demultiplexing.

```
qiime tools import \
--type MultiplexedSingleEndBarcodeInSequence \
--input-path qiime2-tutorial/sequences.fastq.gz \
--output-path qiime2-tutorial/multiplexed-seqs.qza
```

(b) **Multiplexed pair-end FASTQ.**

This format should have one forward.fastq.gz file, containing forward reads from multiple samples, one reverse.fastq.gz file, containing reverse reads from the same samples, and one metadata file with a column of per-sample barcodes for use in FASTQ demultiplexing (or two columns of dual-index barcodes).

```
qiime tools import \
--type MultiplexedPairedEndBarcodeInSequence \
--input-path qiime2-tutorial/muxed-pe-barcode-in-seq \
--output-path qiime2-tutorial/multiplexed-seqs.qza
```

Once the sequences are imported, the amplicon primers are removed with *qiime cutadapt trim-paired*.

```
qiime cutadapt trim-paired \
--i-demultiplexed-sequences qiime2-tutorial/multiplexed-seqs.
qza \
# 515f
--p-front-f GTGCCAGCMGCCGCGGTAA \
# 806r
--p-front-r GGACTACHVGGGTWTCTAAT \
--o-trimmed-sequences qiime2-tutorial/multiplexed-trimmed-
seqs.qza \
--verbose
```

3.3.6 FASTQ Manifest Formats

This format is used when your data do not matches the established formats, you will need to import your data into QIIME2 manually by first creating a "manifest file" and then using the qiime tools import command. The "manifest file" is a tab-separated (i.e., .tsv) text file containing the names, locations and orientation of the read files. In the "manifest file" there is no used naming convention for the files. The first column defines the Sample ID, while the second (and optional third) column defines the absolute filepath to the forward (and optional reverse) reads. The fastq.gz absolute filepaths may contain environment variables (e.g., $HOME or $PWD). The following example illustrates a simple fastq manifest file for paired-end read data for three samples.

```
sample-id  forward-absolute-filepath        reverse-absolute-
filepath
sample-1   $PWD/filepath/sample1_R1.fastq.gz   $PWD/some/file-
path/sample1_R2.fastq.gz
sample-2   $PWD/filepath/sample2_R1.fastq.gz   $PWD/some/file-
path/sample2_R2.fastq.gz
sample-3   $PWD/filepath/sample3_R1.fastq.gz   $PWD/some/file-
path/sample3_R2.fastq.gz
```

There are four variants of FASTQ "manifest" format data. Since importing data in these four formats is very similar, we will only provide examples for two of the variants single and pair-end, respectively: *SingleEndFastqManifestPhred33V2* and *PairedEndFastqManifestPhred64V2*.

(a) **FASTQ manifest formats: SingleEndFastqManifestPhred33V2.**

In this variant of the FASTQ manifest format, the read directions must all either be forward or reverse. This format assumes that the phred offset used for the positional quality scores in all of the fastq.gz/fastq files is 33.

```
qiime tools import \
--type 'SampleData[SequencesWithQuality]' \
--input-path qiime2-tutorial/se-33-manifest \
--output-path qiime2-tutorial/single-end-demux.qza \
--input-format SingleEndFastqManifestPhred33V2
```

(b) **FASTQ manifest formats: PairedEndFastqManifestPhred64V2.**

In this variant of the FASTQ manifest format, there must be forward and reverse read fastq.gz/fastq files for each sample ID. This format assumes that the phred offset used for the positional quality scores in all of the fastq.gz/fastq files is 64. During import, QIIME2 will convert the PHRED 64 encoded quality scores to PHRED 33 encoded quality scores.

```
qiime tools import \
--type 'SampleData[PairedEndSequencesWithQuality]' \
--input-path qiime2-tutorial/pe-64-manifest \
--output-path qiime2-tutorial/paired-end-demux.qza \
--input-format PairedEndFastqManifestPhred64V2
```

3.3.7 Demultiplexing Data (i.e. Mapping Each Sequence to the Sample It Came From)

Summarize the reads and examine the quality of the data. We can view the characteristics of the dataset and the quality scores of the data by creating a QIIME2 visualization artifact.

```
qiime demux summarize
--input-data qiime2-tutorial/demux.qza
--output-visualization qiime2-tutorial/demux.qzv
```

When the *demux.qza* file is created from paired-end reads, the visualization will automatically display the quality distributions for a random sample of both the forward and reverse sequences. Viewing the data look for the point in the forward and reverse reads where quality scores decline below 25–30. We will need to trim reads to this point to create high quality sequence variants. Based on this summary, we choose the nucleotide positions to trim at. The point at which the quality begins to decrease should inform the truncation parameter used in the subsequent sequence denoising step. The truncation value is provided separately for forward and reverse sequence reads, so it is important to note where the quality decrease occurs for both sets.

3.3.8 Denoising and Selecting Sequences Variants with DADA2 (Length Trimming, Denoising, and Chimera Removal)

Reads should then be denoised into amplicon sequence variants (ASVs) to achieve goals like: reducing sequence errors and dereplicating sequences. The process of selecting sequence variants is the core processing step in amplicon analysis. Three different methods have been published to select sequence variants: DADA2 uses and statistical error correction model, Deblur takes an information

theoretic approach and UNOISE2 applies a heuristic model. Here, we develop the DADA2 method.

The qiime DADA2 denoise-paired will both merge and denoise paired-end reads. Two parameters are needed: *--p-trunc-len-f,* indicates the position at which the forward sequence will be truncated, and *--p-trunc-len-r* indicates the position at which the reverse read will be truncated. Using quality score visualization obtained in the previous step, you can choose trunc-len-f and trunc-len-r.

```
qiime dada2 denoise-paired \
--i-demultiplexed-seqs qiime2-tutorial/demux.qza \
--p-trim-left-f 20 \
--p-trim-left-r 20 \
--p-trunc-len-f 280 \
--p-trunc-len-r 200 \
--p-n-threads 8 \
--o-table qiime2-tutorial/table.qza \
--o-representative-sequences qiime2-tutorial/rep-seqs.qza \
--o-denoising-stats qiime2-tutorial/denoising-stats.qza \
--verbose
```

This step produces three output files.

- *denoising_stats.qza*, with a summary of the denoising results.
- *representative_sequences.qza*, the sequences of the exact sequence variants (features); they are joined paired-end reads.
- *table.qza*, the feature table (feature counts by samples).

The obtained files can be summarized and visualized by creating QIIME2 visualization artifacts.

```
qiime feature-table summarize \
  --i-table qiime2-tutorial/table.qza \
  --o-visualization qiime2-tutorial/table.qzv \

qiime feature-table tabulate-seqs \
  --i-data qiime2-tutorial/rep-seqs.qza \
  --o-visualization qiime2-tutorial/rep-seqs.qzv

qiime metadata tabulate \
  --m-input-file qiime2-tutorial/denoising-stats.qza \
  --o-visualization qiime2-tutorial/denoising-stats.qzv
```

3.3.9 Taxonomic Classification

Comparing our query sequences to a reference database of sequences with known taxonomic composition, we can identify the organisms that are present in a sample. To identify these sequence variants two things are needed: a reference database and an algorithm for identifying the sequence using the database.

The primary databases most used are Greengenes, a curated database of archaea and bacteria since 2013 (http://greengenes.secondgenome.com); Silva, the most up-to date database of prokaryotes and eukaryotes (https://www.arb-silva.de) and RDP database, a large collection of archaeal, bacterial, and fungal sequences (https://rdp.cme.msu.edu).

In QIIME2 are used taxonomy classifiers to determine the closest taxonomic affiliation with some degree of confidence or consensus, based on alignment, k-mer frequencies, and so on. Taxonomy classifiers are used through the command *q2-feature-classifier* with three possible different classification methods [10].

- *Classify-consensus-blast* and *classify-consensus-vsearch* are both alignment-based methods that find a consensus assignment across N top hits. These methods take reference database FeatureData[Taxonomy] and FeatureData[Sequence] files directly, and do not need to be pretrained.
- Machine-learning-based classification methods are available through *classify-sklearn*, and theoretically can apply any of the classification methods available in scikit-learn. These classifiers must be trained, for example, to learn which features best distinguish each taxonomic group, adding an additional step to the classification process. Classifier training is reference database- and marker-gene-specific and only needs to happen once per marker-gene/reference database combination.

Generally it is best to train the classifier on the exact region of the 16S, 18S, or ITS you sequenced. For this tutorial we will be using a classifier model trained on the Silva 99% database trimmed to the V4 region with 515F/806R primer set (https://data.qiime2.org/2018.4/common/silva-119-99-515-806-nb-classifier.qza). Other pretrained artifacts are available on the QIIME2 website (https://docs.qiime2.org/).

```
qiime feature-classifier classify-sklearn \
--i-classifier  qiime2-tutorial/silva-119-99-515-806-nb-clas-
sifier.qza \
--i-reads qiime2-tutorial/rep-seqs.qza \
--o-classification qiime2-tutorial/taxonomy.qza
```

To tabulate the features, their taxonomy and the confidence of taxonomy assignment.

```
qiime metadata tabulate \
 --m-input-file qiime2-tutorial/taxonomy.qza \
 --o-visualization qiime2-tutorial/taxonomy.qzv
```

The taxonomic profiles of each sample can be visualized using the *qiime taxa barplot* command. It will be necessary the sample metadata file. This generates an interactive bar plot, bars can be aggregated at the desired taxonomic level and sorted by abundance of a specific taxonomic group or by metadata groupings.

```
qiime taxa barplot \
  --i-table qiime2-tutorial/table.qza \
  --i-taxonomy qiime2-tutorial/taxonomy.qza \
  --m-metadata-file qiime2-tutorial/mapping.txt \
  --o-visualization qiime2-tutorial/taxa-bar-plots.qzv
```

3.3.10 Create a Phylogenetic Tree

There are a number of diversity metrics like unweighted and weighted UniFrac or Faith's phylogenetic diversity (PD) that require the construction of a phylogenetic tree diversity. The process is divided into four steps: multiple sequence alignment, masking, tree building, and rooting.

```
qiime phylogeny align-to-tree-mafft-fasttree \
  --i-sequences qiime2-tutorial/rep-seqs.qza \
  --p-n-threads 8 \
  --o-alignment qiime2-tutorial/aligned-rep-seqs.qza \
   --o-masked-alignment  qiime2-tutorial/masked-aligned-rep-
seqs.qza \
  --o-tree qiime2-tutorial/unrooted-tree.qza \
  --o-rooted-tree qiime2-tutorial/rooted-tree.qza
```

3.3.11 Alpha and Beta Diversity Analysis

Amplicon sequencing allow to look at within sample and between sample ecological diversity. Alpha diversity measures the level of diversity within individual samples. Beta diversity measures the level of diversity or dissimilarity between samples.

This information allow statistically test whether alpha diversity is different between groups of samples (indicating, e.g., that those groups have more/less species richness) and whether beta diversity is greater between groups (indicating, e.g., that samples within a group are more similar to each other than those in another group, suggesting that membership within these groups is shaping the microbial composition of those samples) [10].

The *qiime diversity core-metrics-phylogenetic* command will produce alpha-and beta-diversity measures. An important parameter that needs to be provided to this script is *--p-sampling-depth*, which is the even sampling (i.e., rarefaction) depth. This script will randomly subsample the counts from each sample to the value provided for this parameter.

Select sampling depth by reviewing the information presented in the *table.qzv* QIIME2 artifact that was created above, and in particular the Interactive Sample Detail tab in that visualization.

Choose a value that is as high as possible (so you retain more sequences per sample) while excluding as few samples as possible.

```
qiime diversity core-metrics-phylogenetic \
--i-phylogeny qiime2-tutorial/rooted_tree.qza \
--i-table qiime2-tutorial/table.qza \
--p-sampling-depth 20000 \
--m-metadata-file qiime2-tutorial/mapping.txt \
--output-dir qiime2-tutorial/core_metrics \
--p-n-jobs 16 \
--verbose \
```

This step produces the following files in the core-metrics folder.

```
bray_curtis_distance_matrix.qza
bray_curtis_pcoa_results.qza
evenness_vector.qza
faith_pd_vector.qza
jaccard_distance_matrix.qza
jaccard_pcoa_results.qza
observed_otus_vector.qza
shannon_vector.qza
unweighted_unifrac_distance_matrix.qza
unweighted_unifrac_pcoa_results.qza
weighted_unifrac_distance_matrix.qza
weighted_unifrac_pcoa_results.qza
```

The *qiime diversity alpha-group-significance* and *qiime diversity beta-group-significance* commands will explore the microbial composition of the samples in the context of the sample metadata file. It is possible to test for associations between categorical metadata columns and alpha diversity data. We will do that here for the Faith Phylogenetic Diversity (a measure of community richness) and Shannon indices.

```
qiime diversity alpha-group-significance \
--i-alpha-diversity  qiime2-tutorial/core_metrics/faith_pd_-
vector.qza \
--m-metadata-file qiime2-tutorial/mapping.txt
--o-visualization  qiime2-tutorial/diversity/alpha_PD_signifi-
cance
```

```
qiime diversity alpha-group-significance \
--i-alpha-diversity  qiime2-tutorial/core_metrics/shannon_vec-
tor.qza \
--m-metadata-file qiime2-tutorial/mapping.txt
```

```
--o-visualization qiime2-tutorial/diversity/ alpha_shannon_-
significance
```

For beta diversity, using the *qiime diversity beta-group-significance* command will test whether distances between samples within a group, such as the metadata category: type of samples. If you call this command with the *--p-pairwise* parameter, it will also perform pairwise tests that will allow you to determine which specific pairs of groups differ from one another. We will apply this to the unweighted UniFrac distances.

```
qiime diversity beta-group-significance \
--i-distance-matrix qiime2-tutorial/core_metrics/ unweighte-
d_unifrac_distance_matrix.qza \
--m-metadata-file qiime2-tutorial/mapping.txt
--m-metadata-category TypeSample
--o-visualization qiime2-tutorial/diversity/unweighted-uni-
frac-subject-group-significance.qzv
--p-pairwise
```

3.3.12 Ordination

Ordination is a dimensionality reduction technique that enables the visualization of sample differences, ordination is a popular approach for exploring microbial community composition in the context of sample metadata. QIIME2 use the Emperor tool to explore principal coordinates (PCoA) plots. We will generate Emperor plots for unweighted UniFrac so that the resulting plot will contain axes for principal coordinate 1, principal coordinate 2, and type of samples. We will use that last axis to explore how these samples changed over the type of samples.

```
qiime emperor plot \
--i-pcoa qiime2-tutorial/core-metrics /unweighted_unifrac_p-
coa_results.qza \
--m-metadata-file qiime2-tutorial/mapping.txt \
--p-custom-axes TypeSample \
--o-visualization qiime2-tutorial/diversity/unweighted-uni-
frac-emperor-type-sample.qzv
```

3.3.13 Alpha Rarefaction Plotting

The final analysis is to explore the alpha diversity as a function of sampling depth. The *qiime diversity alpha-rarefaction* command will generate rarefaction curves based on the Shannon diversity and observed OTUs measures by default, and will additionally generate phylogenetic diversity-based curves if the phylogenetic tree created above is provided using the *--i-phylogeny* parameter [12]. Average diversity values will be plotted for each sample at each even sampling depth, and samples can be grouped based on metadata in the resulting visualization. The value that you provide

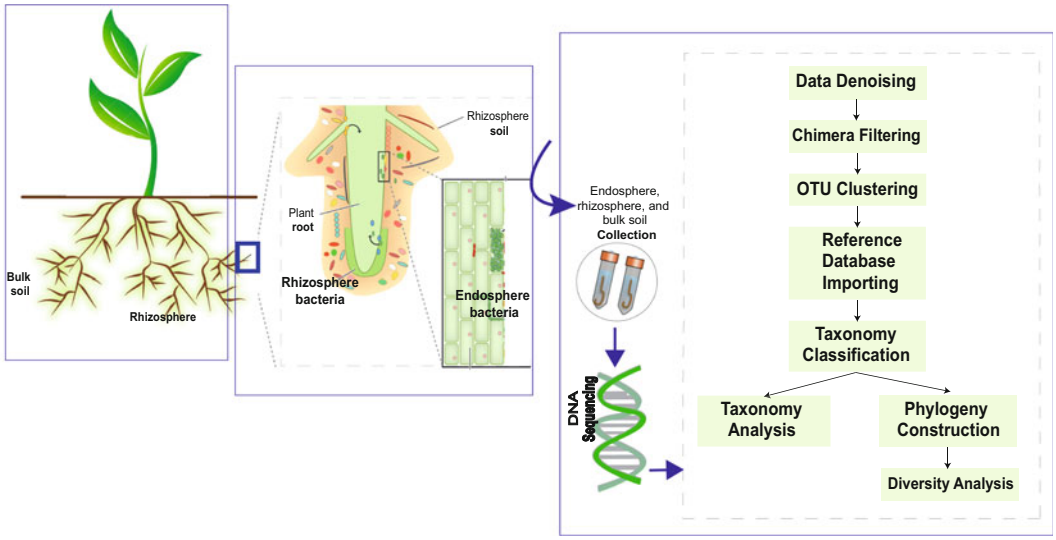


Fig. 2 Flowchart of methods for sampling the bulk soil, rhizosphere, and endosphere fractions and a generic workflow to analysis of 16S rRNA marker-gene

for *--p-max-depth* should be determined by reviewing the "Frequency per sample" information presented in the *table.qzv* file that was previously [10].

```
qiime diversity alpha-rarefaction \
--i-table qiime2-tutorial/table.qza
--i-phylogeny rooted_tree.qza
--p-max-depth 41000
--m-metadata-file qiime2-tutorial/mapping.txt \
--o-visualization diversity_41000/alpha_rarefaction.qzv
```

Figure 2 shows the flow diagram of the series of steps followed for obtention of bulk soil, rhizosphere, and endosphere fractions and a generic workflow to analysis of 16S rRNA marker-gene.

4 Notes

1. Compositional analysis of plant-associated communities is important to develop tools in order to manipulate root-associated microbiota for example could be useful to increase crop productivity.
2. Understanding factors that control the microbial communities and microbial processes is essential to achieve the potential of microbial management in agricultural systems. Evaluation of microbial diversity in designed experiments provides an avenue to generate hypotheses about the mechanisms of treatment effects on host phenotype and performance [13].

3. Microbial marker-gene sequence data can be used to generate comprehensive taxonomic profiles of the microorganisms present in a given community and for other community diversity analyses.
4. This tutorial covered a wide range of analyses of 16S rRNA marker-gene that can be done with QIIME2. Other approaches can complement this workflow such as analysis tools in R-based (Phyloseq and Microbiome packages) and functional analysis attempt to impute function from taxonomy (PiCrust and Tax4fun packages).

References

1. Berendsen RL, Pieterse CMJ, Bakker PAHM (2012) The rhizosphere microbiome and plant health. Trends Plant Sci 17:478–486. https://doi.org/10.1016/j.tplants.2012.04.001
2. Yu P, Hochholdinger F (2018) The role of host genetic signatures on root–microbe interactions in the rhizosphere and endosphere. Front Plant Sci 871:1–5. https://doi.org/10.3389/fpls.2018.01896
3. Hassani MA, Durán P, Hacquard S (2018) Microbial interactions within the plant holobiont. Microbiome 6:58. https://doi.org/10.1186/s40168-018-0445-0
4. Meena KK, Sorty AM, Bitla UM et al (2017) Abiotic stress responses and microbe-mediated mitigation in plants: the omics strategies. Front Plant Sci 8:1–25. https://doi.org/10.3389/fpls.2017.00172
5. Mommer L, Kirkegaard J, van Ruijven J (2016) Root–root interactions: towards a rhizosphere framework. Trends Plant Sci 21:209–217. https://doi.org/10.1016/j.tplants.2016.01.009
6. Ryan RP, Germaine K, Franks A et al (2008) Bacterial endophytes: recent developments and applications. FEMS Microbiol Lett 278:1–9. https://doi.org/10.1111/j.1574-6968.2007.00918.x
7. Mesa V, Navazas A, González-Gil R et al (2017) Use of endophytic and rhizosphere bacteria to improve phytoremediation of arsenic-contaminated industrial soils by autochthonous Betula celtiberica. Appl Environ Microbiol 83:e03411. https://doi.org/10.1128/AEM.03411-16
8. Caporaso JG, Lauber CL, Walters WA et al (2012) Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. ISME J 6:1621–1624. https://doi.org/10.1038/ismej.2012.8
9. FASTQ files explained. https://support.illumina.com/bulletins/2016/04/fastq-files-explained.html. Accessed 20 Oct 2020
10. Bolyen E, Rideout JR, Dillon MR et al (2019) Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. Nat Biotechnol 37:852–857. https://doi.org/10.1038/s41587-019-0209-9
11. Rideout JR, Chase JH, Bolyen E et al (2016) Keemei: cloud-based validation of tabular bioinformatics file formats in Google sheets. Gigascience 5:27. https://doi.org/10.1186/s13742-016-0133-6
12. Hall M, Beiko RG (2018) 16S rRNA gene analysis with QIIME2. Methods Mol Biol 1849:113–129
13. Poudel R, Jumpponen A, Kennelly MM et al (2019) Rootstocks shape the rhizobiome: rhizosphere and endosphere bacterial communities in the grafted tomato system. Appl Environ Microbiol 85:1–16. https://doi.org/10.1128/AEM.01765-18

Chapter 12

Applying Synteny Networks (SynNet) to Study Genomic Arrangements of Protein-Coding Genes in Plants

Samuel David Gamboa-Tuz, Alejandro Pereira-Santana, Tao Zhao, and M. Eric Schranz

Abstract

In comparative genomics, the study of synteny can be a powerful method for exploring genome rearrangements, inferring genomic ancestry, defining orthology relationships, determining gene and genome duplications, and inferring gene positional conservation patterns across taxa. In this chapter, we present a step-by-step protocol for microsynteny network (SynNet) analysis, as an alternative to traditional methods of synteny comparison, where nodes in the network represent protein-coding genes and edges represent the pairwise syntenic relationships. The SynNet pipeline consists of six main steps: (1) pairwise genome comparisons between all the genomes being analyzed, (2) detection of inter- and intrasynteny blocks, (3) generation of an entire synteny database (i.e., edgelist), (4) network clustering, (5) phylogenomic profiling of the gene family of interest, and (6) evolutionary inference. The SynNet approach facilitates the rapid analysis and visualization of synteny relationships (from specific genes, specific gene families up to all genes) across a large number of genomes.

Key words Synteny network, Microsynteny, Gene family, Comparative genomics, TMBIM, Bax Inhibitor

1 Introduction

Rapid advances in both sequencing technology and computational resources have generated a wealth of complete genome sequences from a wide range of organisms, which facilitates our ability to study the trait and genome evolution of diverse branches on the tree of life. Comparative genomics provides specific information about gene and genome dynamics such as gene duplications, polyploidy (whole genome duplications; WGD) rearrangements, gain and loss and adaptive evolution [1–3]. Typically, phylogenomic studies often report on the presence/absence of genes across specific lineages, but neglect the importance of the genomic context of these genes.

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_12,

In comparative genomics, synteny (the analysis of the relative gene order) is a very powerful method to study gene and genome evolution [4]. Syntenic genes are those anchor pairs that share a similar genomic context between two duplicated regions (collinear regions), either within the same genome or between two genomes. These genes are also often referred to as syntelogs or syntenic homologous genes. Analysis of microsynteny facilitates a wide-range of research. For example, it allows researchers to explore genome rearrangements between two or more genomes, define orthology relationships [5–7] and to infer shared ancestry on a set of genes and to determine ancient polyploidy events (paleopolyploidy events; [8]).

Pairwise dot-plots or parallel coordinate plots are commonly used to represent synteny, where each parallel line represents a specific region of a genome. Among these databases are SynFind [9], PGDD [10], and Genomicus [11], among others. These approaches have some limitations including the following: (1) some programs can only handle pairwise comparisons, (2) comparisons are restricted to only a few species (sometimes by no more than three species due the great number of intragenome duplications by WGD) [12], and (3) visualizing large multigene families in several species is very difficult [13].

In this chapter we present a synteny network (SynNet) approach, as an alternative method to analyzing and visualizing syntenic relationships between genomes without restricting the number of species being investigated. SynNet aids in identifying lineage-specific transpositions and positional conservation and ancient tandem duplications. In a SynNet analysis, nodes represent protein-coding genes and edges represent the pairwise synteny relationships. For a complete and in-depth explanation of the method please see [13–15].

The SynNet approach has been successfully applied in several works to understand the evolution of gene families in different taxonomic groups [15–18], and has resulted in a promising complement in the evolutionary analysis of traits of interest. As an example of the SynNet approach, [19] report the analysis of the plant Transmembrane Bax Inhibitor Motif containing (TMBIM) superfamily genes across 46 land plants. TMBIM superfamily is composed of calcium channels with many biological roles in microorganisms, animals, and plants, such as regulation of programmed cell death (PCD). In eukaryotes, the TMBIM superfamily is divided into the Bax Inhibitor (BI) and Lifeguard (LFG) families. Gamboa-Tuz, et al. (2018) found by SynNet approach that at least two major transpositions occurred during the evolution of the BI genes in monocots. On the other hand, the LFG family remained as a single synteny block in most plant species and only one major transposition event occurred in a few legume species [19].

Table 1
Plant species used to build a SynNet in the present protocol

Species	Family	Abbreviation
Medicago truncatula	Fabaceae	mt
Glycine max	Fabaceae	gm
Phaseolus vulgaris	Fabaceae	pv
Cicer arietinum	Fabaceae	ca
Cajanus cajan	Fabaceae	cc
Zea mays	Poaceae	zm
Oryza sativa	Poaceae	os
Brachypodium distachyon	Poaceae	bd
Sorghum bicolor	Poaceae	sb
Setaria italica	Poaceae	si
Arabidopsis thaliana	Brassicaceae	at
Brassica rapa	Brassicaceae	br
Capsella rubella	Brassicaceae	cb
Brassica oleracea	Brassicaceae	bo
Arabidopsis lyrata	Brassicaceae	al

In order to demonstrate more details of the SynNet approach, we give a step-by-step tutorial here by partially reproducing the results described in [19] for TMBIM superfamily genes with a smaller data set of 15 species, including plants from the Brassicaceae, Fabaceae, and Poaceae families (Table 1).

2 Materials

2.1 Hardware and System Requirements

This protocol has been tested in a DELL Inspiron-5559 laptop with the following characteristics.

- Operating system: Ubuntu 20.04.2 LTS x86_64.
- Shell: bash v5.0.17.
- CPU: Intel i7-6500U (4) at 3.100 GHz.
- GPU: AMD ATI Radeon HD 8670A/8670 M/8690 M/Intel Skylake GT2 [HD Graphics 520].
- Memory: 2679MiB/7853MiB.

The time to build a SynNet following this protocol was ~2 h.

2.2 Software

The following programs must be installed in your system (*see* **Note 1**).

- Diamond v2.0.9. Link: https://github.com/bbuchfink/diamond [20].
- MCScanX. Link: https://github.com/wyp1125/MCScanX [21].
- CFinider 2.0.6. Link: http://www.cfinder.org [22].
- R v4.0.4 Link: https://www.r-project.org [23].
  - Tidyverse 1.3.0. Link: https://www.tidyverse.org [24].
  - ggtree v2.5.1.9001. Link: https://github.com/YuLab-SMU/ggtree [25] (*see* **Note 2**).
- Cytoscape v3.8.0. Link: https://cytoscape.org [26].
- Git v2.28.0. Link: https://git-scm.com (*see* **Note 3**).

Please follow the specific installation instructions for each program indicated on their respective websites.

As an alternative to programs installation, you can create a CONDA environment containing Diamond, R (r-base, tidyverse, ggtree), Cytoscape, and Git, and just installing MCScanX and CFinder in your system (*see* **Note 4**). You must activate the created environment before running the subsequent pipeline.

Furthermore, for this protocol, we will use the iTOL web tool [27] for visualizing phylogenetic trees with cluster information. A temporary iTOL tree can be created at https://itol.embl.de/upload.cgi. You can permanently save your tree with a personal iTOL account.

2.3 General Guideline to Run the Codes

- Commands that must be typed (or pasted) in the Linux terminal are indicated by a shell prompt, that is, a dollar symbol in bold ("$"), followed by the code in a monospaced font. The prompt must not be typed.
- Names of files and directories in the text are indicated with a cursive font. For example, *build_synnet.sh* refers to a script file with that name.
- Code is written in a monospace font. The text between "<" and ">" symbols must be substituted with your information (e.g., a full path to a specific file).

2.4 Protocol Starter

For this protocol, we will build a SynNet of 15 plant genomes (Table 1). A protocol starter including an example work directory, scripts, and database files is available at https://github.com/sdgamboa/synnet_protocol_starter.

1. To download the protocol starter, open a terminal in your home directory (or the directory of your choice) and execute the following code (*see* **Note 3**).

```
$ git clone https://github.com/sdgamboa/synnet_protocol_starter.git
```

We recommend creating a specific folder into your home directory to contain all your programs and scripts. As an example you can create the "apps" directory into your home (i.e., `$ mkdir $HOME/apps`).

2. Set up the working directory profile by sourcing the *set_profile.sh* file (*see* **Notes 5** and **6**).

```
$ cd synnet_protocol_starter
$ source set_profile.sh
```

3. Go to the database directory (*db*), and download and decompress the data set files.

```
$ cd $db
$ curl
https://zenodo.org/record/5546148/files/synnet_dataset.tar.gz?download=1 --output synnet_dataset.tar.gz
$ tar -xvzf synnet_dataset.tar.gz
```

All data sets in the correct format must be now in the database directory (*db*). The database files consist of 15 proteomes in FASTA format (one for each plant species) and the genomic regions of their respective protein-coding genes in BED format. Figures 1 and 2 show examples of FASTA and BED files in the correct format for the construction of a SynNet. Notice that a prefix consisting of an abbreviation of the species name and an underscore ("_") has been added to the sequence identifiers in the FASTA files (Fig. 1) and the chromosome and gene identifiers in the BED files (Fig. 2). This helps to parse the output files during downstream analyses and visualization of results (*see* **Note 7**).

3 Methods

In this protocol, the main steps for building a SynNet are contained in the *build_synnet.sh* file script (Fig. 3). Downstream analyses such as network clustering of the TMBIM superfamily and visualization of results are also included in this protocol (Fig. 3).

3.1 Build a SynNet

To build a SynNet for the 15 plant species for this example, go to the *db* directory and execute the *build_synnet.sh* script (*see* **Notes 8** and **9**).

```
$ cd $db
$ build_synnet.sh species_list.txt
```

Fig. 1 Example input FASTA file. First 10 sequences of the *at.pep* file (located in the *db* directory), which contains all predicted proteins from the Arabidopsis genome in FASTA format. The "at_" prefix has been appended to each protein identifier (marked with a red rectangle). Notice that the sequence identifiers match the sequence identifiers in their corresponding BED file (*at.bed*; column 2, Fig. 2)

Two directories will be created:

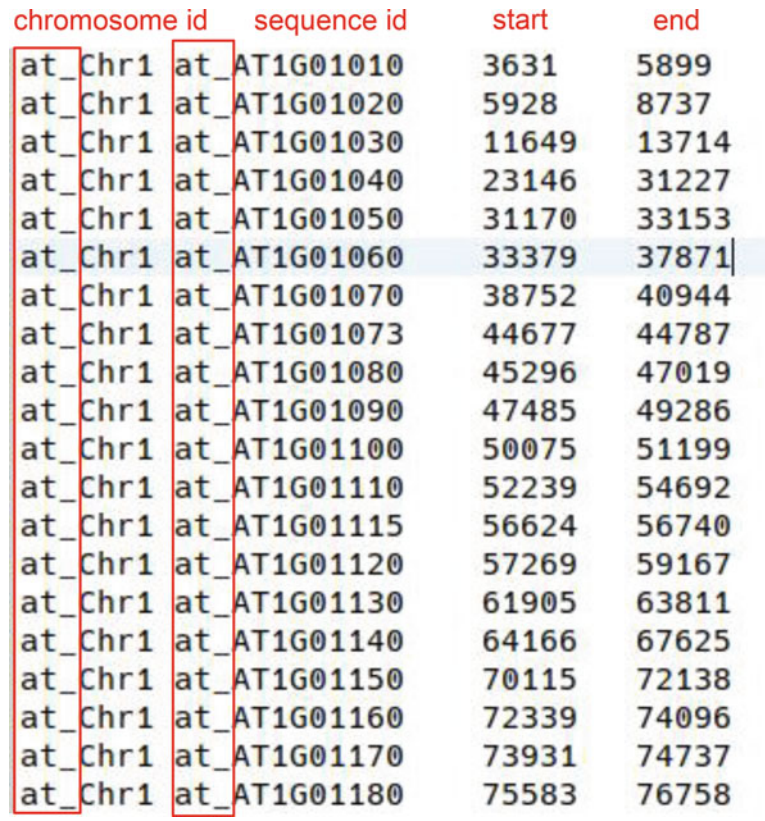
- The *DiamondDB* directory contains the diamond database files necessary to run homology searches.
- The *synnetOutput.[date-time].[options]* directory contains several output files including the *SynNet_k*s*m*.csv* final file (*see* **Note 10**). This is a tab-delimited file with five columns without headers: column 1, MCScanX synteny blocks; column 2, MCScanX score; column 3, gene 1 (syntelog to gene 2); column 4, gene 2 (syntelog to gene 1); column 5, location of the genes on a syntenic block. For this protocol, we are only interested in columns 3 and 4, which together make an edge list or network of syntenic genes.

3.2 Retrieve the TMBIM Family Network from the SynNet

The step above creates a SynNet including all gene families present in the genomes of the 15 species used in this protocol. To study a specific gene family you must provide a list of ids in a text file, in this case, the *TMBIM_id_list.txt* file within the *files* directory (*see* **Note 11**).

1. Go to the *analyses* directory and execute the *extract_subnetwork.sh* script.

```
$ cd $wdir/analyses
$ extract_subnetwork.sh $wdir/files/TMBIM_id_list.txt \
$db/synnetOutput*/SynNet_*.csv > TMBIM_Network.csv
```



chromosome id	sequence id	start	end
at_Chr1	at_AT1G01010	3631	5899
at_Chr1	at_AT1G01020	5928	8737
at_Chr1	at_AT1G01030	11649	13714
at_Chr1	at_AT1G01040	23146	31227
at_Chr1	at_AT1G01050	31170	33153
at_Chr1	at_AT1G01060	33379	37871
at_Chr1	at_AT1G01070	38752	40944
at_Chr1	at_AT1G01073	44677	44787
at_Chr1	at_AT1G01080	45296	47019
at_Chr1	at_AT1G01090	47485	49286
at_Chr1	at_AT1G01100	50075	51199
at_Chr1	at_AT1G01110	52239	54692
at_Chr1	at_AT1G01115	56624	56740
at_Chr1	at_AT1G01120	57269	59167
at_Chr1	at_AT1G01130	61905	63811
at_Chr1	at_AT1G01140	64166	67625
at_Chr1	at_AT1G01150	70115	72138
at_Chr1	at_AT1G01160	72339	74096
at_Chr1	at_AT1G01170	73931	74737
at_Chr1	at_AT1G01180	75583	76758

Fig. 2 Example input BED file. First 10 features of the *at.bed* file (located in the *db* directory), which contains the genomic ranges of the protein-coding genes of the Arabidopsis genome in BED format. The meaning of each column is written in red at the top: column 1, chromosome identifier; column 2, sequence identifier; column 3, start position in the chromosome; column 4, end position in the chromosome. The "at_" prefix has been appended to each chromosome (column 1) and sequence (column 2) identifiers (marked with a red rectangle). Notice that the sequence identifiers (column 2) match the sequence identifiers in their corresponding FASTA file (*at.pep*; Fig. 1)

The *TMBIM_Network.csv* file now contains all and only the detected synteny relationships of the TMBIM family in the 15 analyzed plant species (*see* **Note 12**).

3.3 Network Clustering

Now that we have the TMBIM SynNet, we would like to find discrete synteny blocks to make evolutionary inferences. We can accomplish this by clustering the TMBIM network into communities. There are several network clustering methods available, such as the clique percolation [28] and Infomap [29, 30]) methods [14]. In the current protocol, we will use the clique percolation method (k-clique = 4) implemented in CFinder to detect SynNet communities or subnetworks. You can specify another "k-clique" as a convenience as this is not a set value in all studies.

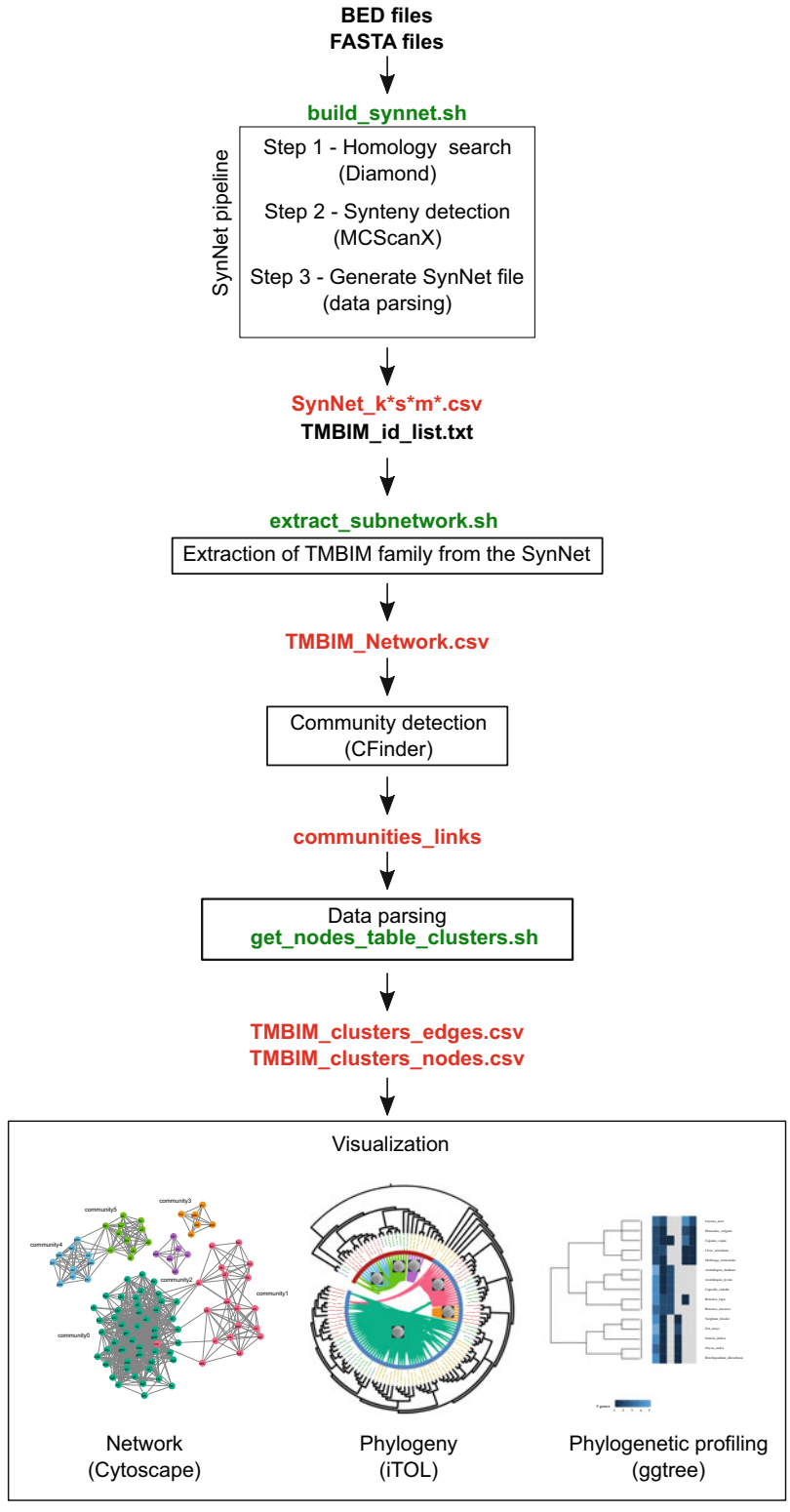


Fig. 3 SynNet pipeline implemented in the present protocol. The main steps of the protocol are in rectangles. Filenames are in bold: files downloaded from the SynNet starter protocol are in black, output files are in red, scripts are in green

1. Go to the *analyses* directory and execute the following commands (*see* **Note 13**):

```
$ cd $wdir/analyses
$ cut --output-delimiter=" " -f 3,4 \
TMBIM_Network.csv > TMBIM_Network_EdgeList.txt
$ CFinder_commandline64 -k 4 \
-l <full/path/to/Cfinder/license.txt> \
-i TMBIM_Network_EdgeList.txt -o CFinder_output_k4
```

   The results must be now inside the *CFinder_output_k4/k = 4* directory. The edges (pairwise synteny relationships between genes) are in the *communities_links* file and the nodes (genes) are in the *communities* file.

2. Extract an edge list (network) per community with the following.

```
$ sed -e '/\(#\|^$\)/d' CFinder_output_k4/k=4/communities_links
| \
csplit --suppress-matched -z -f community -b %01d - '/:/' '{*}'
$ ls community*
```

   A total of six community files should have been created (*community0* to *community5*).

3. Extract the node list for each community with the following.

```
$ for i in community*; do get_node_table_clusters.sh \
$wdir/files/nodes_metadata.csv $i > $i.nodes.csv; done
$ for i in community{0..5}; do \
sed -i -e "s/$/\t$i/;1s/community[0-9]$/community/" \
$i.nodes.csv; done
```

3.4 Visualization

3.4.1 Network Visualization

The network (and clusters) can be visualized with different available tools, for example, Cytoscape [26] or Gephi [31]. For this protocol, we use Cytoscape here.

1. Concatenate all clusters into a single .csv file.

```
$ cd $wdir/analyses
$ cat community{0..5} > TMBIM_clusters_edges.csv
```

2. Get a list of nodes with associated metadata with the following command (*see* **Note 14**).

```
$cat *nodes.csv | sed -e '2,${/species\tfamily/d}' \
> TMBIM_clusters_nodes.csv
```

3. Import the network clusters in Cytoscape by clicking on the "File > Import > Network from File" menu button and selecting the *TMBIM_clusters_edges.csv* file. In the advanced options make sure to use the space character as field delimiter and leave the "use first line as column names" box unchecked. Columns 1 and 2 must be set as "source" and "target" nodes even when the network is treated as undirected (as in this case).
4. Import the nodes' metadata into Cytoscape by clicking on the "File > Import > Table from File" button and selecting the *TMBIM_clusters_nodes.csv* file (*see* **Note 14**). Make sure to import data to "Selected networks only" specifying the current network file (*TMBIM_clusters_edges.csv)*. In the advanced options select the TAB character as field delimiter and leave the "use first line as column names'"box checked.
5. You can edit several features of the network to make it more informative. For example, you could edit labels and color in the "Style" tab, or the layout in the "Layout" menu. In the example in Fig. 4, the network clusters are depicted with the "yFiles Organic" layout (the "yFiles" layout algorithm must be installed in Cytoscape), and the nodes labeled according to species abbreviations and colored according to the community (*see* **Note 15**).

3.4.2 Visualizing Clusters in a Phylogenetic Tree with iTOL

Once the clusters have been obtained, they can be visualized in a phylogenetic tree using a connections template in iTOL. For this example, we have built a phylogenetic tree named as *TMBIM_proteins.trimmed.afa.treefile*. Go to https://itol.embl.de/upload.cgi and load the phylogenetic tree located in the *files* directory: *TMBIM_proteins.trimmed.afa.treefile* (*see* **Note 11**). You can import the annotation templates as well (which are also in the files directory); just drag the *iTOL_template_connections.txt, iTOL_template_labels.txt,* and *iTOL_template_strips.txt* files to the iTOL tree in your browser. You can root the tree in the middle of the two TMBIM families (BI and LFG; the longest branch), invert the tree, and ignore branch lengths to obtain a phylogenetic tree similar to the one depicted in Fig. 5.

3.4.3 Phylogenomic Profiling

There are several tools to do phylogenetic profiling. In this case, we will use the ggtree package implemented in the R language.

1. Copy the *phylogenetic_profiling.R* script to the *analyses* directory and execute it.

```
$ cd $wdir/analyses
$ cp $wdir/scripts/phylogenetic_profiling.R .
$ Rscript --vanilla phylogenetic_profiling.R
```

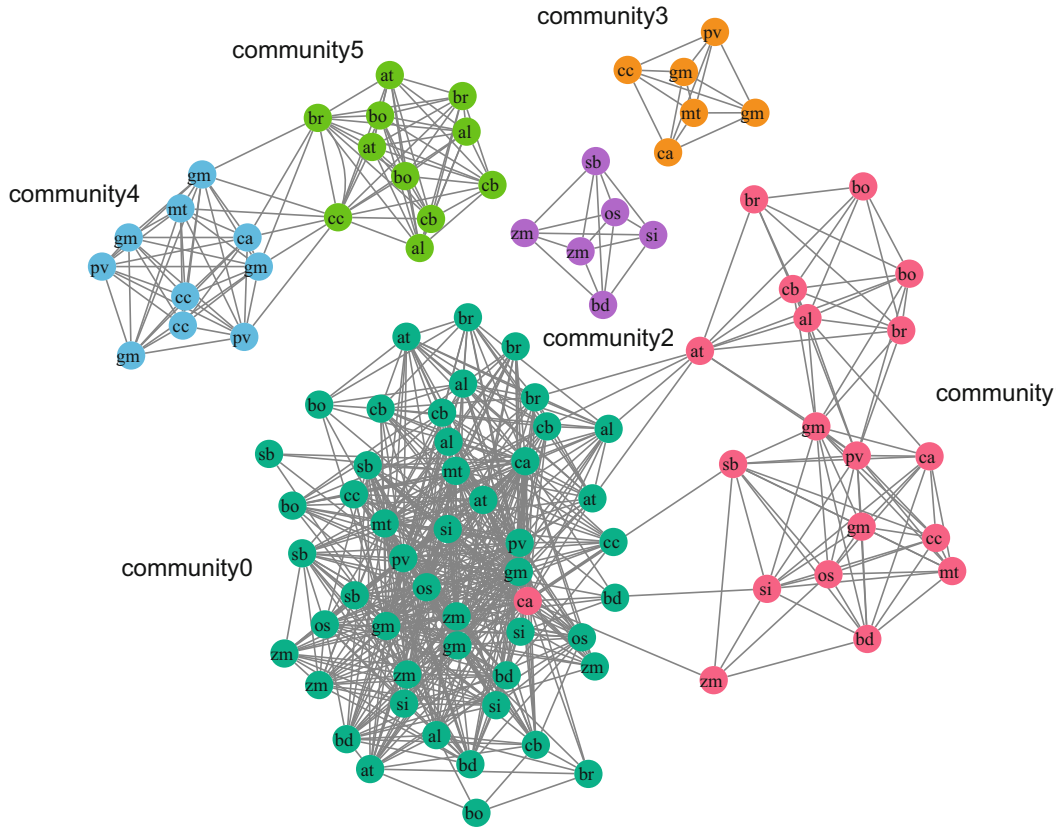


Fig. 4 Visualization of the TMBIM SynNet communities. Communities 0 to 5 detected for the TMBIM superfamily in 15 plant species. The figure was generated with Cytoscape [26]. Labels were added with Inkscape v1.0.2 (1.0.2 + r75 + 1)

The *phylogenetic_profiling.R* script takes the *species_tree.newick* (located in the *files* directory) and the *TMBIM_clusters_nodes.csv* (created during this session in the *analyses* directory) files as inputs. You should obtain a phylogenetic tree of the 15 species and a profiling figure depicting the number of genes per community (Fig. 6).

4 Notes

1. *Diamond*, *MCScanX*, and *CFinder* must be added to the *PATH* environment variable (bash shell). You can do so by adding the following lines to your *.bashrc* file in your home directory (before opening a new terminal).

```
export PATH=$PATH:<full/path/to/diamond_scripts>
export PATH=$PATH:<full/path/to/MCScanX_scripts>
export PATH=$PATH:<full/path/to/MCScanX/downstream_analyses>
export PATH=$PATH:<full/path/to/CFinder_scripts>
```

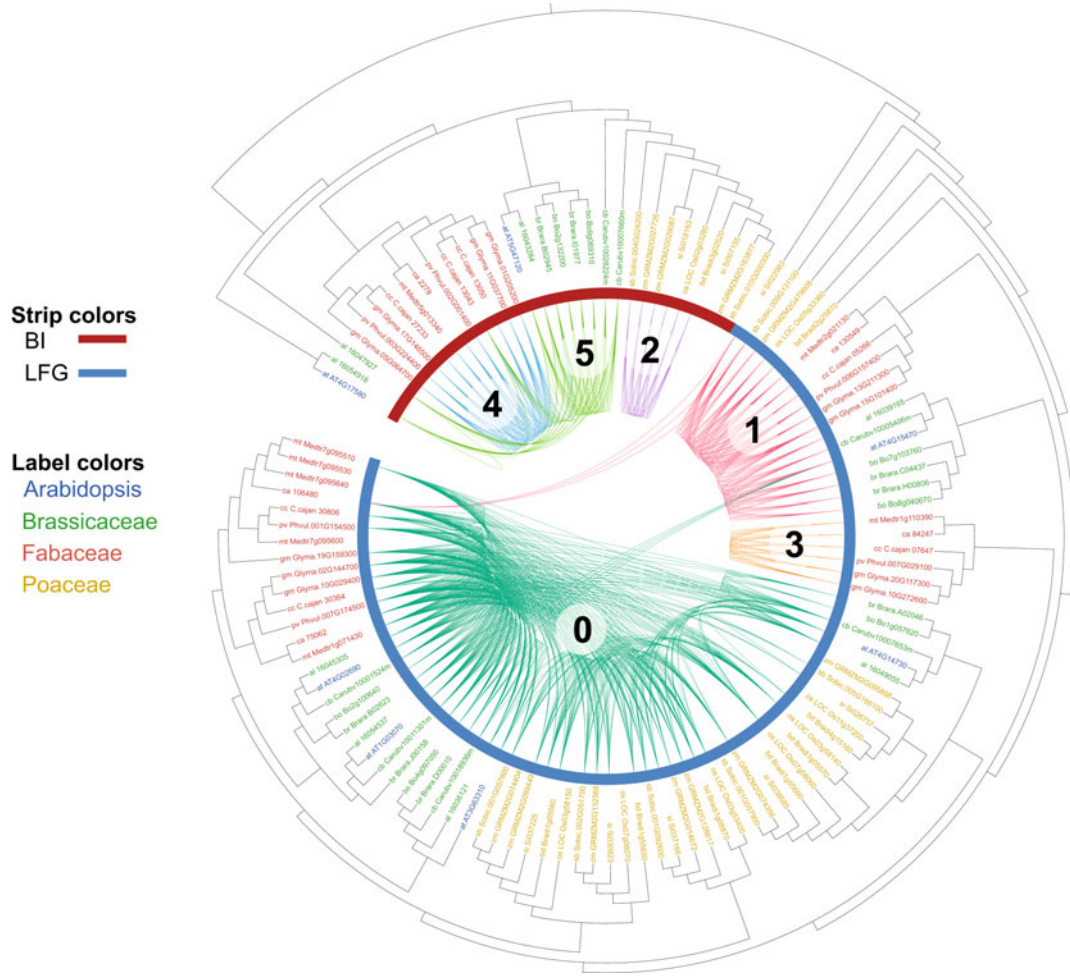


Fig. 5 Visualization of the TMBIM SynNet communities in a phylogenetic tree. Phylogenetic tree of 117 TMBIM proteins found in 15 plant species in the present study. The numbers of the communities and the legend were added with Inkscape v1.0.2 (1.0.2 + r75 + 1)

***** Skip this step for Diamond if you decide to use the CONDA environment, *see* **Note 4**.

2. Installing the develop version is recommended (you need devtools installed in R). Within an R session, install ggtree with (Skip this step is you decide to use the CONDA environment, *see* **Note 4**).

```
> devtools::install_github("YuLab-SMU/ggtree")
```

3. Git is only necessary to clone the repository in your computer with a single step. You can avoid using Git by downloading the repository with the following:

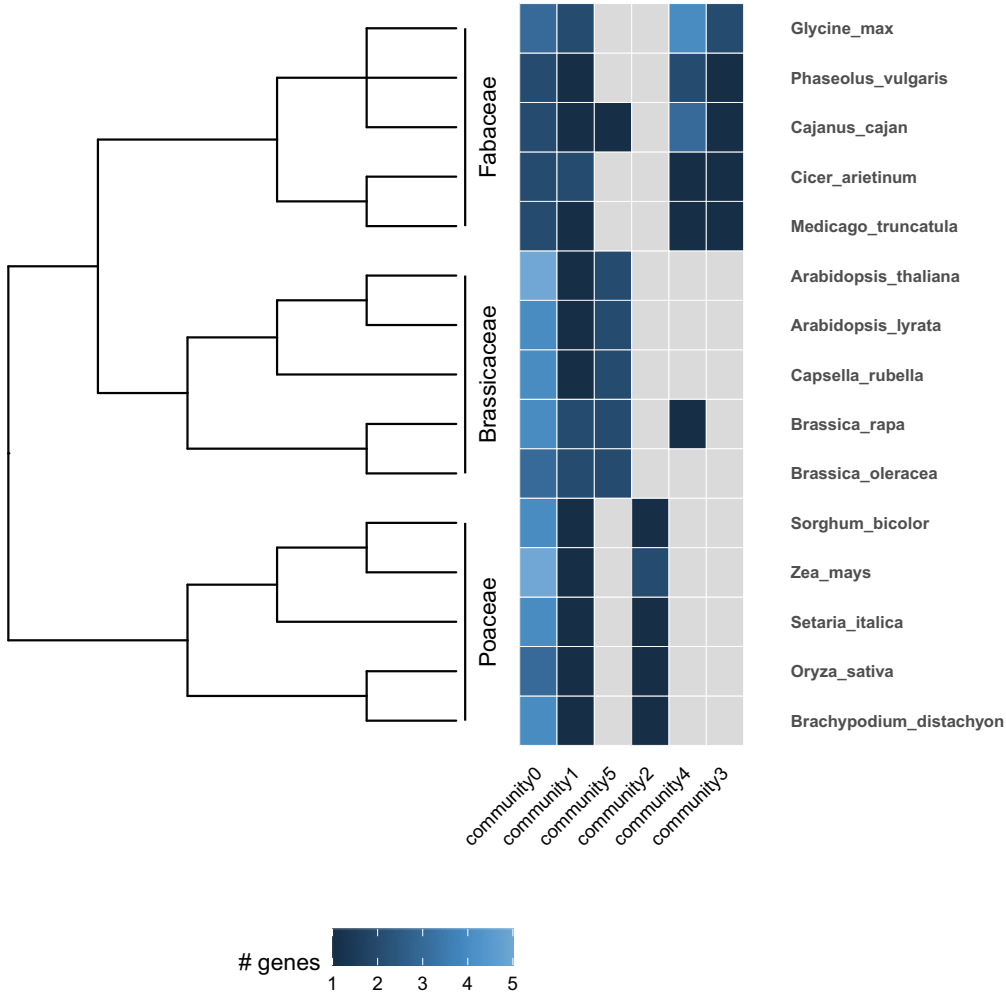


Fig. 6 Phylogenetic profiling of the TMBIM SynNet communities. The tree was generated with ggtree [25]

```
$ wget \
https://github.com/sdgamboa/synnet_protocol_starter/archive/
main.zip
$ unzip main.zip
$ cd synnet_protocol_starter-main
```

Then, you'll need to download the data set of FASTA and GFF files:

```
$ cd db
$ wget \
```

```
https://zenodo.org/record/5546148/files/synnet_dataset.tar.
gz?download=1
$ tar -xvzf synnet_dataset.tar.gz
```

4. Installing Diamond, Git, R (r-base, Tidyverse, ggtree), and Cytoscape via CONDA environment.
   (a) Install CONDA on Linux.

```
$ cd $HOME
$ wget https://repo.continuum.io/archive/Anaconda2-2019.10-
Linux-x86_64.sh
$ bash Anaconda2-2019.10-Linux-x86_64.sh
```

Follow the prompts on the installer screens.

   (b) Set conda environment.

```
$ echo ". ~/anaconda2/etc/profile.d/conda.sh" >> ~/.bashrc
$ source ~/.bashrc
```

Conda documentation (https://docs.conda.io/projects/conda/en/latest/user-guide/install/linux.html).

   (c) Create a text file called "synnet.yml" and type the text shown below "OUTPUT".

```
$ cat synnet.yml
OUTPUT
name: SYNNET_PIPELINE
channels:
 - bioconda
 - conda-forge
 - r
dependencies:
 - diamond=2.0.*
 - cytoscape=3.7
 - git
 - r-base=4.0.*
 - r-tidyverse
 - bioconductor-ggtree
```

   (d) Create a SYNNET_PIPELINE conda environment using the synnet.yml file.

```
$ conda env create -f synnet.yml
```

   (e) Activate the SYNNET_PIPELINE environment before run the pipeline.

```
$ conda activate SYNNET_PIPELINE
```

- To run cytoscape just type: `$ cytoscape.sh`.
- To deactivate conda environment just type: `$ conda deactivate`.

5. You might need to grant execution permission for each of the script files inside the *scripts* directory before sourcing the *set_profile.sh* file. You can do so with the chmod command. For example, inside the *scripts* directory you can execute the following.

```
$ for script in *sh; do chmod +x $script; done
```

6. By sourcing the *set_profile.sh* file, the scripts in the scripts directory will be made available without the need of specifying the full path every time a script is executed. This must be done only once every time a terminal is opened.
7. The BED files were obtained by parsing GFF/GFF3 files. The FASTA and GFF/GFF3 files must belong to the same annotation version. Consider that the correct parsing of both types of files depends on the format used by each database website.
8. The *build_synnet.sh* script is a modified version of the original pipeline available at https://github.com/zhaotao1987/SynNet-Pipeline. The *build_synnet.sh* script implements the main features of the pipeline as of March 13, 2021. Please refer to the source to see what new features are added to the pipeline in the future.
9. You can get help about the main functionalities of the *build_synnet.sh* script by executing:

```
$ build_synnet.sh -h
```

10. The k, s, and m letters in the SynNet output filename indicate the options that were used to run the *synnet_build.sh* script. Please use synnet_build.sh -h to know the meaning of each option (*see* **Note 9**).
11. Homology search and phylogenetic inference are out of the scope of the present protocol. Example outputs of those analyses are provided in the *files* directory. You can access how those files were obtained through the *README.md* file within the files directory.
12. You would probably like to get some metadata for the nodes. Execute the following.

```
$ cd $wdir/analyses
$ get_node_table.sh $wdir/files/nodes_metadata.csv \
TMBIM_Network.csv > TMBIM_node_metadata.csv
```

13. Alternatively to the command line, you can have access to a graphical version of CFinder by going to the CFinder main directory and executing the *start.sh* script there.
14. You can visualize the network without this file, but it's extremely helpful to have metadata associated with each node. This way you can change color and labels easily within Cytoscape and make informative figures.
15. Colors used for these communities were selected manually within Cytoscape. The colors are in the *community_colors.txt* file within the *files* directory. Depending on the number of clusters, it could be more convenient to create a column with the colors and labels of the clusters. In this case, the nodes can also be colored according to taxonomy if required.

References

1. Jiao Y, Wickett NJ, Ayyampalayam S et al (2011) Ancestral polyploidy in seed plants and angiosperms. Nature 473(7345):97–100. https://doi.org/10.1038/nature09916
2. Soltis PS, Soltis DE (2016) Ancient WGD events as drivers of key innovations in angiosperms. Curr Opin Plant Biol 30:159–165. https://doi.org/10.1016/j.pbi.2016.03.015
3. Ohno S (1970) Evolution by gene duplication. Springer, Berlin, Heidelberg
4. Zhao T, Zwaenepoel A, Xue JY et al (2021) Whole-genome microsynteny-based phylogeny of angiosperms. Nat Commun 12:3498. https://doi.org/10.1038/s41467-021-23665-0
5. Parey E, Louis A, Cabau C et al (2021) Synteny-guided resolution of gene trees clarifies the functional impact of whole-genome duplications. Mol Biol Evol 37(11): 3324–3337. https://doi.org/10.1093/molbev/msaa149
6. Van Bel M, Proost S et al (2012) Dissecting plant genomes with the PLAZA comparative genomics platform. Plant Physiol 158: 590–600. https://doi.org/10.1104/pp.111.189514
7. Pevzner P, Tesler G (2003) Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. Genome Res 13: 37–45. https://doi.org/10.1101/gr.757503
8. Wan T, Liu Z, Leitch L et al (2021) The Welwitschia genome reveals a unique biology underpinning extreme longevity in deserts. Nat Commun 12:4247. https://doi.org/10.1038/s41467-021-24528-4
9. Tang H, Bomhoff MD, Briones E et al (2015) SynFind: compiling syntenic regions across any set of genomes on demand. Genome Biol Evol 7(12):3286–3298. https://doi.org/10.1093/gbe/evv219
10. Lee TH, Tang H, Wang X et al (2013) PGDD: a database of gene and genome duplication in plants. Nucleic Acids Res 41(Database issue): D1152–D1158. https://doi.org/10.1093/nar/gks1104
11. Muffato M, Louis A, Poisnel CE et al (2010) Genomicus: a database and a browser to study gene synteny in modern and ancestral genomes. Bioinformatics 26(8):1119–1121. https://doi.org/10.1093/bioinformatics/btq079
12. Ibarra-Laclette E, Lyons E, Hernández-Guzmán G et al (2013) Architecture and evolution of a minute plant genome. Nature 498:94–98. https://doi.org/10.1038/nature12132
13. Zhao T, Schranz ME (2017) Network approaches for plant phylogenomic synteny analysis. Curr Opin Plant Biol 36:129–134. https://doi.org/10.1016/j.pbi.2017.03.001
14. Zhao T, Eric Schranz M (2019) Network-based microsynteny analysis identifies major differences and genomic outliers in mammalian and angiosperm genomes. PNAS 116(6): 2165–2174. https://doi.org/10.1073/pnas.1801757116

15. Zhao T, Holmer R, de Bruijn S et al (2017) Phylogenomic Synteny Network analysis of MADS-Box transcription factor genes reveals lineage-specific transpositions, ancient tandem duplications, and deep positional conservation. Plant Cell 29:1278–1292. https://doi.org/10.1105/tpc.17.00312
16. Pereira-Santana A, Gamboa-Tuz SD, Zhao T et al (2020) Fibrillarin evolution through the Tree of Life: comparative genomics and microsynteny network analyses provide new insights into the evolutionary history of Fibrillarin. PLoS Comput Biol 16:e1008318. https://doi.org/10.1371/journal.pcbi.1008318
17. Gao B, Wang L, Oliver M et al (2020) Phylogenomic synteny network analyses reveal ancestral transpositions of auxin response factor genes in plants. Plant Methods 16:70. https://doi.org/10.1186/s13007-020-00609-1
18. Kerstens MHL, Schranz ME, Bouwmeester K (2020) Phylogenomic analysis of the APETALA2 transcription factor subfamily across angiosperms reveals both deep conservation and lineage-specific patterns. Plant J 103: 1516–1524. https://doi.org/10.1111/tpj.14843
19. Gamboa-Tuz SD, Pereira-Santana A, Zhao T et al (2018) New insights into the phylogeny of the TMBIM superfamily across the tree of life: Comparative genomics and synteny networks reveal independent evolution of the BI and LFG families in plants. Mol Phylogenet Evol 126:266–278
20. Buchfink B, Xie C, Huson DH (2015) Fast and sensitive protein alignment using DIAMOND. Nat Methods 12:59–60
21. Wang Y, Tang H, DeBarry JD et al (2012) MCScanX: a toolkit for detection and evolutionary analysis of gene synteny and collinearity. Nucleic Acids Res 40:e49–e49
22. Adamcsek B, Palla G, Farkas IJ et al (2006) CFinder: locating cliques and overlapping modules in biological networks. Bioinformatics 22:1021–1023
23. R Core Team (2021) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria
24. Wickham H, Averick M, Bryan J et al (2019) Welcome to the tidyverse. J Open Source Softw 4:1686
25. Yu G, Smith DK, Zhu H et al (2017) Ggtree: an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. Meth Ecol Evol 8:28–36
26. Shannon P, Markiel A, Ozier O et al (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13:2498–2504
27. Letunic I, Bork P (2007) Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation. Bioinformatics 23:127–128
28. Palla G, Derényi I, Farkas I et al (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435:814–818
29. Rosvall M, Esquivel AV, Lancichinetti A et al (2014) Memory in network flows and its effects on spreading dynamics and community detection. Nat Commun 5:4630
30. Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. PNAS 105:1118–1123
31. Bastian M, Heymann S, Jacomy M (2009) Gephi: an open source software for exploring and manipulating networks. In: Proceedings of the International AAAI Conference on web and social media, vol 3, pp 361–362

Chapter 13

Plant In Situ Hi-C Experimental Protocol and Bioinformatic Analysis

Francisco J. Pérez-de los Santos, Jesús Emiliano Sotelo-Fonseca, América Ramírez-Colmenero, Hans-Wilhelm Nützmann, Selene L. Fernandez-Valverde, and Katarzyna Oktaba

Abstract

Hi-C enables the characterization of the 0conformation of the genome in the three-dimensional nuclear space. This technique has revolutionized our ability to detect interactions between linearly distant genomic sites on a genome-wide scale. Here, we detail a protocol to carry out in situ Hi-C in plants and describe a straightforward bioinformatics pipeline for the analysis of such data, in particular for comparing samples from different organs or conditions.

Key words Hi-C, Chromosome conformation, Arabidopsis, Sequencing, Bioinformatics, Differential interactions

1 Introduction

Eukaryotic chromosomes repeatedly fold in the three-dimensional space of the cell nucleus. This organization is known as genome or chromosome topology and has several hierarchical levels [1]. At the most basic level, DNA is wrapped around histone proteins and arranged into loosely or tightly packed nucleosomal arrays (reviewed in [2]). Above the nucleosomal level, chromatin folds into loops that bring linearly distant regions of the genome into close spatial proximity, creating a multitude of functional interactions, for example enabling distal enhancers to interact with target promoters [3, 4]. Topologically associating domains (TADs), characterized by an increased interaction count between loci located in the same domain, and less frequent interactions with neighboring loci, comprise the next organization level [5, 6]. TADs themselves are organized into active (A) and inactive (B) compartments that share similar gene expression and epigenetic profiles [7]. Finally,

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_13,

individual chromosomes occupy distinct regions in the nucleus, known as chromosome territories [8]. This genomic architecture exhibits conserved folding patterns across species and cell types in both animals and plants [9].

The analysis of genome topology by chromosome conformation capture methods has drastically improved our ability to identify interactions between DNA domains that are seemingly distant in a linear representation of the genome [10]. The recent widespread use of chromatin conformation capture (3C) followed by unbiased high-throughput sequencing techniques (Hi-C) has provided comprehensive knowledge of chromatin interactions at a genome-wide level in a variety of organisms [11].

Here, we present a detailed in situ Hi-C protocol for plants along with a complete bioinformatic analysis workflow (Fig. 1), based on previously described wet-lab and bioinformatic methods [12–20]. This protocol was implemented in cotyledons and roots of *Arabidopsis thaliana*. The experimental protocol and the bioinformatics pipeline can be adapted for analysis in other organs and plant species.

Briefly, the experimental protocol involves formaldehyde fixation of cells, isolation of nuclei and digestion of chromatin with a restriction enzyme. The overhangs left by the restriction enzyme are then filled in with biotin-conjugated nucleotides, the fragments are religated, size-selected, and purified using streptavidin-coated magnetic beads. Sequencing libraries are amplified, undergo quality control and the reads are paired-end sequenced. The bioinformatics workflow details the steps required for filtering and aligning reads, generating contact matrices, and annotating compartments, TADs, interaction peaks, and differential interactions using publicly available bioinformatics packages.

2 Materials

2.1 Reagents

1. 2 M glycine.
2. DNase/RNase-free distilled water (Invitrogen).
3. Liquid nitrogen.
4. Vectashield Antifade Mounting Medium (Vector).
5. 4′,6-diamidino-2-phenylindole (DAPI).
6. DpnII restriction enzyme and DpnII buffer (NEB).
7. 0.4 mM biotin-14-dATP (Invitrogen).
8. DNA Polymerase I, Large (Klenow) Fragment (NEB).
9. T4 DNA Ligase and 10x T4 DNA Ligase buffer (Thermo Scientific).
10. 10 mg/ml bovine serum albumin (BSA; NEB).

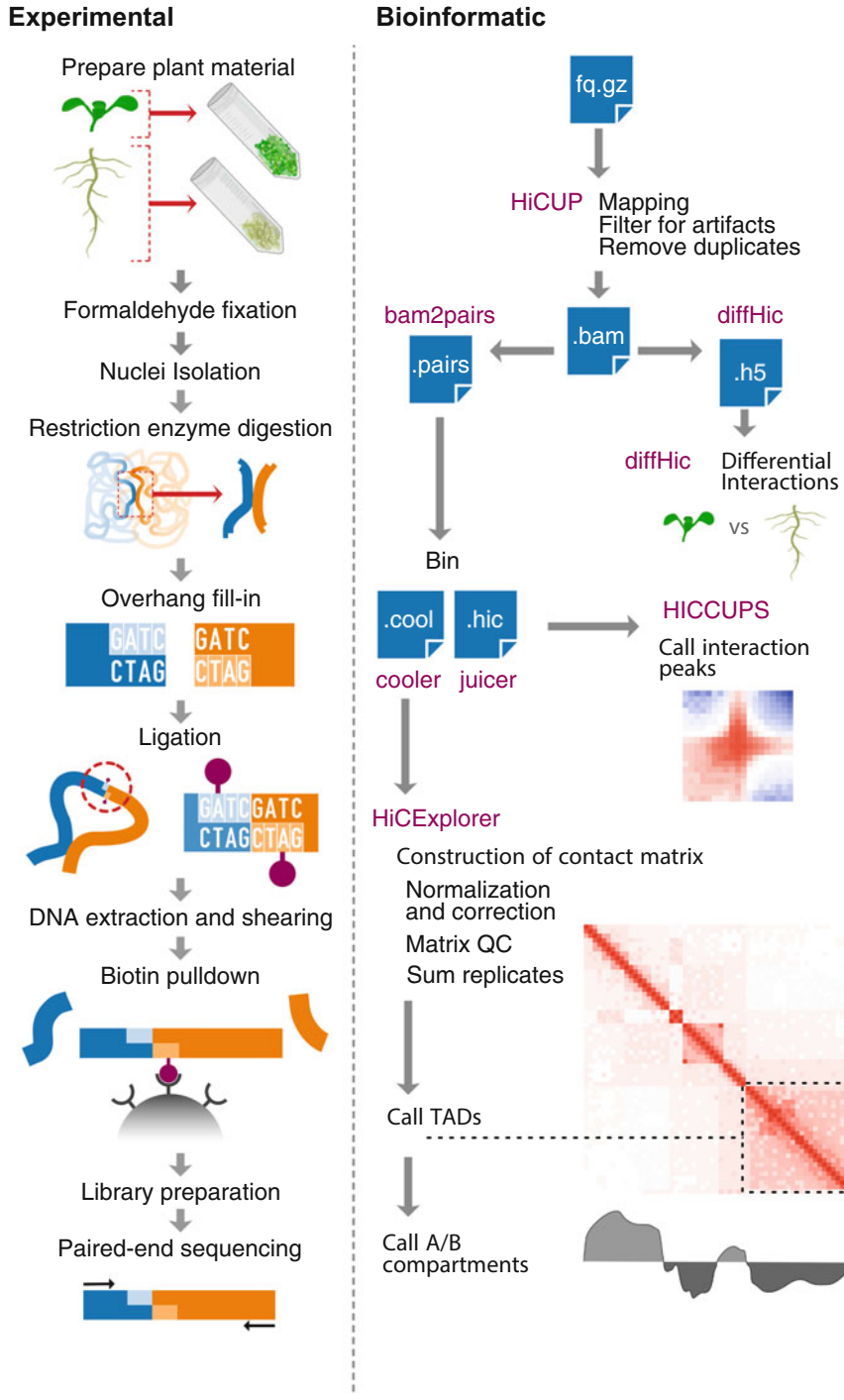


Fig. 1 Overview of experimental in situ Hi-C protocol and bioinformatic analysis of sequencing data. Left panel, simplified schematic representation of the Hi-C protocol. After obtaining plant material, nucleic acid–protein interactions are preserved by cross-linking with a formaldehyde solution, followed by nuclei extraction. DNA is subsequently digested with a restriction enzyme (DpnII), generating 5′ overhangs, which are filled with regular dNTPs and biotin-14-dATP, followed by blunt-end ligation. Next, DNA is purified and sonicated, producing small-sized DNA fragments that can be captured using streptavidin-coated beads. These fragments are used

11. 20 mg/ml Proteinase K (Thermo Scientific).
12. 5 M sodium chloride (NaCl).
13. 25:24:1 (v/v/v) phenol–chloroform–isoamyl alcohol.
14. Chloroform.
15. 20 mg/ml glycogen (Roche).
16. Ethanol.
17. 20 mg/ml RNase A (Invitrogen).
18. 1 Kb Plus DNA Ladder (Invitrogen).
19. Agarose.
20. T4 DNA polymerase and 10× NEB 2.1 buffer (NEB).
21. 0.5 M ethylenediaminetetraacetic acid (EDTA; Invitrogen).
22. Dynabeads MyOne Streptavidin C1 (Invitrogen).
23. NEBNext Ultra II DNA Library Prep Kit for Illumina (NEB).
24. NEBNext Multiplex Oligos for Illumina (NEB).
25. SPRIselect beads (Beckman Coulter).
26. 2× NEBNext Ultra II Q5 Master Mix (NEB).
27. 100× SYBR Green I Nucleic Acid Gel Stain (Invitrogen).
28. Qubit dsDNA HS Assay Kit (Invitrogen).
29. Bioanalyzer High Sensitivity DNA Kit (Agilent).

2.2 Labware and Equipment

1. 1.5 ml microcentrifuge tubes.
2. Greiner dishes (Sigma-Aldrich).
3. 50 ml centrifuge tubes.
4. Scalpel.
5. Nylon filters.
6. 10 ml serological pipettes.
7. Miracloth (Millipore).
8. Paper towels.
9. Mortar and pestle.

Fig. 1 (continued) to generate libraries for high-throughput paired-end sequencing. Right panel, the bioinformatics analysis begins with mapping sequencing read pairs to the reference genome and filtering out uninformative read pairs. The resulting bam file is converted to file formats compatible with HiC analysis tools. This information is used to generate a count matrix, which is then processed using HiCExplorer tools, to account for visibility biases. If the analysis aims to compare several matrices, a normalization step is necessary to account for differences in sequencing depth between samples. Once the matrices are corrected, topologically associating domains (TADs), compartments and interaction peaks can be determined. diffHiC can also be used to account for biases between libraries and to detect regions that interact differently in each experimental condition

10. Cell counting chamber.
11. Maxymum Recovery pipette filter tips (Axygen).
12. Phase Lock Gel Heavy tubes (Quantabio).
13. 1.5 ml LoBind microcentrifuge tubes (Eppendorf).
14. MicroTube AFA Fiber Pre-Slit Snap-Cap (Covaris).
15. 0.2 ml PCR tubes.
16. Plant growth incubator.
17. Desiccator connected to a vacuum pump with manometer.
18. Refrigerated centrifuge for 50 ml conical tubes.
19. Refrigerated centrifuge for 1.5 ml microcentrifuge tubes.
20. Epifluorescent microscope.
21. Thermomixer (Eppendorf).
22. NanoDrop spectrophotometer (Thermo Scientific).
23. Agarose gel casting tray and combs.
24. DNA electrophoresis chamber and power supply.
25. S2 Focused-Ultrasonicator (Covaris).
26. Magnetic rack for 1.5 ml microcentrifuge tubes.
27. PCR thermocycler.
28. Real-time PCR instrument.
29. Qubit Fluorometer (Thermo Scientific).
30. Bioanalyzer Instrument (Agilent).

2.3 Reagent Setup

1. MS medium: 4.3 g/l Murashige and Skoog basal salt mixture (MS; Sigma-Aldrich), 10 g/l sucrose, 6 g/l Phytagel (Sigma-Aldrich). Adjust pH to 5.8 with 1 M NaOH before adding Phytagel, store at 4 °C.
2. Nuclei isolation buffer (NIB): 20 mM HEPES pH 8.0, 250 mM sucrose, 1 mM $MgCl_2$, 5 mM KCl, 40% (v/v) glycerol, 0.25% (v/v) Triton X-100, 0.1 mM PMSF, and 0.1% (v/v) 2-mercaptoethanol (*see* **Note 1**) [12].
3. Nuclei isolation buffer with formaldehyde (NIB-FA): 20 mM HEPES pH 8.0, 250 mM Sucrose, 1 mM $MgCl_2$, 5 mM KCl, 40% (v/v) glycerol, 0.25% (v/v) Triton X-100, 4% (v/v) formaldehyde (ultrapure, methanol free; Polysciences), 0.1 mM PMSF, and 0.1% (v/v) 2-mercaptoethanol [12].
4. Nuclease isolation buffer with protease inhibitor cocktail (NIB-P): 20 mM HEPES pH 8.0, 250 mM Sucrose, 1 mM $MgCl_2$, 5 mM KCl, 40% (v/v) glycerol, 0.25% (v/v) Triton X-100, 1× cOmplete ULTRA protease inhibitor cocktail (Roche), 0.1 mM PMSF, and 0.1% (v/v) 2-mercaptoethanol [12].

5. 1× PBS: Diluted from 10× phosphate buffered saline (Gibco).
6. 0.5% (v/v) SDS: Prepare from 10% (v/v) ultrapure sodium dodecyl sulfate solution (Invitrogen).
7. 10% (v/v) Triton X-100: Prepare from Triton X-100 solution (Sigma-Aldrich).
8. 3.3 mM dCTP–dGTP–dTTP mix: Prepare from 100 mM dNTPs set (Invitrogen).
9. Extraction buffer: 50 mM Tris–HCl, pH 8.0, 10 mM EDTA, and 1% (v/v) SDS.
10. 3 M sodium acetate pH 5.2 ($C_2H_3NaO_2$).
11. 70% (v/v) ethanol.
12. EB buffer: 10 mM Tris–HCl pH 8.0.
13. 1 mM dATP: Prepare from 100 mM dNTPs set (Invitrogen).
14. 1 mM dGTP: Prepare from 100 mM dNTPs set (Invitrogen).
15. 2× B&W buffer: 10 mM Tris–HCl pH 7.5, 1 mM EDTA, 2 M NaCl, prepare fresh.
16. 1× B&W buffer +0.1% (v/v) Triton X-100, prepare fresh.
17. 80% (v/v) ethanol, prepare fresh.

3 Methods

The protocol below describes the preparation of Hi-C sequencing libraries from two plant organs: cotyledons and roots of *Arabidopsis thaliana* 8-day-old seedlings. All steps are described for processing a single sample (either from cotyledons or roots). If using other plant material or species, growth conditions, harvest, cross-linking, and nuclei isolation steps may have to be adapted and optimized.

3.1 Preparation of Plant Material

1. Using a 1.5 ml microcentrifuge tube, measure a volume of 300 μl of *Arabidopsis thaliana* seeds (approximately 6900 seeds). This amount of seeds is sufficient to collect 2–3 g of each organ, cotyledon or root, from 8-day-old seedlings.
2. Grow seedlings vertically on MS medium in Greiner dishes for 8 days in a plant growth incubator under long-day conditions (16 h light at 22 °C and 8 h darkness at 16 °C).

3.2 Formaldehyde Cross-Linking of Plant Material

1. Separate cotyledons from roots with a scalpel and transfer 2–3 g of each type of organ immediately into a 50 ml centrifuge tube containing 15 ml NIB. Store on ice (*see* **Note 2**).
2. To cross-link, add 15 ml NIB-FA (2% formaldehyde final concentration). Mix by swirling with a 1 ml pipette tip. Completely immerse the plant material in the solution using nylon filters. Vacuum-infiltrate in a desiccator for 1 h at room temperature

(RT). Release vacuum every 15 min to enhance the penetration of fixative (*see* **Note 3**).

3. To stop cross-linking reaction, release vacuum, remove the nylon filter and add 2 ml 2 M glycine. Mix by pipetting up and down with a 10 ml serological pipette. Reimmerse the sample and vacuum-infiltrate for 5 min.
4. Decant liquid and wash the sample three times with ice-cold 1× PBS or ultrapure water.
5. Carefully wrap the plant material in Miracloth and dry by pressing with paper towels.
6. Transfer the dried sample to a 50 ml centrifuge tube and flash-freeze in liquid nitrogen. Proceed to nuclei isolation.

3.3 Nuclei Isolation

1. Use mortar and pestle (precooled with liquid nitrogen) to grind the cross-linked plant material to a fine powder in liquid nitrogen. Transfer powder to a liquid nitrogen-cooled 50 ml centrifuge tube.
2. Resuspend the sample in 10 ml NIB-P. Mix by gentle agitation with a 1 ml pipette tip until the solution becomes homogeneous. Place the tube on ice (*see* **Note 4**).
3. Separate nuclei by filtering samples twice through a double layer of Miracloth. Collect the filtrate in a 50 ml centrifuge tube and wash Miracloth with an extra 5 ml of NIB-P to collect the remaining material. To avoid contamination with cell debris, do not squeeze the Miracloth.
4. Centrifuge nuclei suspension at 3000 × *g* for 15 min at 4 °C. Carefully remove and discard the supernatant using a 10 ml serological pipette.
5. To resuspend the nuclei pellet, add 1 ml NIB-P and mix by gently swirling with a pipette tip. To avoid mechanical damage of nuclei, continue to resuspend by pipetting gently using a cut-off pipette tip. Do not vortex.
6. Transfer sample to a 1.5 ml microcentrifuge tube using a cut-off pipette tip. Centrifuge at 1900 × *g* for 5 min at 4 °C. Discard supernatant.
7. Wash nuclei pellet twice with 1 ml NIB-P. Resuspend nuclei pellet by pipetting gently using a cut-off pipette tip. Centrifuge at 1900 × *g* for 5 min at 4 °C. Discard the supernatant.
8. Resuspend the nuclei in 100 µl NIB-P by pipetting gently using a cut-off pipette tip.
9. Assess nuclei quality by staining 1 µl nuclei suspension with Vectashield mounting medium with DAPI. Analyze the nuclei using epifluorescence microscopy [14]. Intact nuclei show sharp contours as described by [12]. Estimate nuclei quantity

by staining 3 μl nuclei suspension with DAPI and pipette sample onto a counting chamber. Count the individual nuclei using epifluorescence microscopy. Determine the concentration of nuclei and proceed with $>10^7$ nuclei [14].

10. Centrifuge the nuclei suspension at 1900 × *g* for 5 min at 4 °C. Discard the supernatant.
11. Remove NIB-P traces by washing the nuclei pellet twice with 300 μl 1× DpnII buffer. Resuspend nuclei pellet by pipetting gently using a cut-off pipette tip. Centrifuge at 1900 × *g* for 5 min at 4 °C. Discard the supernatant. For all subsequent steps use pipette filter tips. In order to minimize sample retention and enhance yield, use Maxymum Recovery pipette filter tips or equivalent.
12. Permeabilize nuclei by gently resuspending the pellet in 100 μl 0.5% SDS using a cut-off pipette tip, be careful to avoid froth or bubbles. Incubate for 10 min at 65 °C (*see* **Note 5**).
13. Quench SDS by adding 70 μl ultrapure water and 50 μl 10% Triton X-100. Mix by pipetting up and down, be careful to avoid froth or bubbles. Incubate for 15 min at 37 °C while shaking at 450 rpm (*see* **Note 6**).

3.4 Restriction Enzyme Digestion

1. Add 25 μl 10× DpnII buffer and mix by pipetting. Collect 10 μl as undigested chromatin control, store at −20 °C until further processing. Digest chromatin by adding 100 U of DpnII enzyme. Mix by pipetting up and down. Incubate the digestion reaction for 3 h at 37 °C while shaking at 450 rpm. This digestion step may be done overnight (*see* **Note 7**).
2. Inactivate DpnII enzyme by incubating for 20 min at 62 °C. Collect 10 μl as digested chromatin control, store at −20 °C until further processing. Transfer the sample to ice (*see* **Note 8**).
3. Assess the quantity and quality of undigested and digested chromatin control samples by performing a cross-linking reversal, followed by DNA extraction, spectrophotometric quantification, and gel electrophoresis. Alternatively, perform rapid reversal of chromatin cross-linking (*see* **Note 9**).

3.5 Overhang Fill-in with a Biotinylated Nucleotide

1. The overhangs left by DpnII are filled-in by adding 19 μl 0.4 mM biotin-14-dATP (0.03 mM final concentration), 2.3 μl 3.3 mM dCTP/dGTP/dTTP (0.03 mM final concentration) and 50 U DNA Polymerase I Large (Klenow) Fragment. Mix by pipetting up and down. Incubate for 90 min to 2 h at 37 °C while shaking at 450 rpm.

3.6 In-Situ Ligation of Proximal Ends

1. Filled-in DNA fragments are ligated by adding 719 μl ultrapure water, 120 μl 10× T4 DNA ligase buffer, 100 μl 10% Triton

X-100, 50 Weiss U T4 DNA ligase and 12 μl 10 mg/ml BSA. Mix by inverting the tube 5 times. Incubate overnight at 16 °C with gentle rotation (*see* **Note 10**).

3.7 Cross-Linking Reversal

1. Centrifuge the nuclei suspension at 2500 × *g* for 10 min at 4 °C. Carefully remove and discard the supernatant.
2. Resuspend the nuclei in 380 μl extraction buffer.
3. To digest proteins, add 20 μl 20 mg/ml proteinase K and mix by pipetting. Incubate for 30 min at 55 °C while shaking at 1000 rpm.
4. Add 100 μl 5 M NaCl and mix by pipetting. Incubate at least 8 h or overnight at 68 °C.

3.8 DNA Extraction

1. Before use, spin Phase Lock Gel (Quantabio) tube at 12,000 × *g* for 30 s at RT. Transfer decross-linked sample to prespun Phase Lock Gel tube. Add 500 μl phenol–chloroform–isoamyl alcohol (25:24:1). Mix thoroughly by vigorous shaking for 2 min to form a transiently homogenous suspension, do not vortex. Centrifuge at 12,000 × *g* for 5 min at RT.
2. Perform a second extraction, by adding 500 μl chloroform to the same Phase Lock Gel tube. Mix thoroughly by vigorous shaking for 2 min, do not vortex. Centrifuge at 12,000 × *g* for 5 min at RT.
3. Transfer DNA-containing aqueous upper phase to a 1.5 ml LoBind microcentrifuge tube.
4. Precipitate DNA by adding 1/10 volume 3 M sodium acetate pH 5.2, 1 μl glycogen and 1200 μl ice-cold 100% ethanol.
5. Mix by inverting 5 times and incubate at −80 °C for at least 1 h. Centrifuge at 20,000 × *g* for 1 h at 4 °C. Remove the supernatant.
6. Wash the pellet twice with 1 ml 70% ethanol. Centrifuge at 20,000 × *g* for 5 min at RT. Remove the supernatant and air-dry pellet.
7. Dissolve the ligated Hi-C sample in 50 μl EB buffer.
8. Digest RNA by adding 1 μl 20 mg/ml RNase A. Mix by pipetting. Incubate at 37 °C for 30 min.
9. To assess ligation efficiency, measure DNA concentration using a NanoDrop spectrophotometer, and load 500 ng of sample and a 1 Kb Plus DNA Ladder on a 1.8% agarose gel for electrophoresis. Run the gel for 40 min at 90 V. Ligated Hi-C sample will appear as a smear of intermediate sizes between undigested and digested chromatin controls (assessed in Subheading 3.4).

3.9 Biotin Removal from Unligated DNA Ends

1. Remove biotin from unligated DNA ends, by adding to the 50 μl ligated Hi-C sample 12 μl 10× NEB 2.1 buffer, 3 μl 1 mM dATP, 3 μl 1 mM dGTP, 1.2 μl 10 mg/ml BSA,15 U T4 DNA polymerase, and ultrapure water up to 120 μl. Incubate for 30 min at 20 °C (*see* **Note 11**).
2. To stop the reaction, add 3 μl 0.5 M EDTA. Mix by pipetting up and down.

3.10 DNA Shearing

1. Transfer sample to a Covaris AFA fiber microtube; be careful to avoid bubbles.
2. Shear the DNA to ~200–400 bp (depends on sequencing read length). For Covaris S2 Focused-Ultrasonicator use the following program: 2 cycles of 50 s, 10% duty, intensity 5, 200 cycles per burst (*see* **Note 12**).
3. To assess the fragment size distribution of the sheared sample, load 300 ng and a 100 bp DNA Ladder on a 1.8% agarose gel for electrophoresis. Run the gel for 40 min at 90 V. The sheared sample will appear as a smear from 200 to 400 bp approximately.

3.11 Biotin Pulldown

1. To prepare Dynabeads MyOne Streptavidin C1 for biotin pulldown, vortex 10 mg/ml beads suspension for >30 s and transfer 60 μl to a 1.5 ml LoBind microcentrifuge tube. Wash twice by adding 1 ml 2× B&W buffer and resuspending by pipetting. Place the tube on a magnetic rack for 2 min. Carefully remove and discard the supernatant. Resuspend the beads in 120 μl 2× B&W.
2. To capture biotinylated DNA fragments, adjust the volume of the sheared Hi-C sample to 120 μl with ultrapure water, and add to the 120 μl 2× B&W bead suspension. Incubate the tube using gentle rotation for 20 min at RT.
3. Place the tube on a magnetic rack for 2 min. Carefully remove and discard the supernatant.
4. Wash beads twice with 600 μl 1× B&W buffer +0.1% Triton X-100. Mix by pipetting and incubate for 2 min at 55 °C while shaking at 1000 rpm. Place the tube on a magnetic rack for 2 min. Carefully remove and discard the supernatant.
5. Wash beads with 600 μl EB buffer. Mix by pipetting. Place the tube on a magnetic rack for 2 min. Carefully remove and discard the supernatant.
6. Resuspend bead-Hi-C sample in 50 μl EB buffer (*see* **Note 13**).

3.12 Sequencing Library Preparation

The following steps describe the Hi-C sequencing library preparation using NEBNext Ultra II DNA Library Prep Kit for Illumina (NEB) and NEBNext Multiplex Oligos for Illumina (NEB).

3.13 End Repair and dA-Tailing

1. Transfer the 50 μl bead-Hi-C sample to a 0.2 ml PCR tube.
2. Add 7 μl NEBNext Ultra II End Prep Reaction Buffer and 3 μl NEBNext Ultra II End Prep Enzyme Mix. Pipette up and down at least 10 times to mix thoroughly, be careful to avoid bubbles. Perform a quick spin to collect all liquid from the sides of the tube.
3. Place in a PCR thermocycler with the heated lid set to ≥75 °C and run the following program: 30 min 20 °C, 30 min 65 °C, hold at 10 °C.

3.14 Adaptor Ligation

1. To ligate the adaptor, add in the following order: 2.5 μl 15 μM NEBNext Adaptor for Illumina, 1 μl NEBNext Ligation Enhancer, and 30 μl NEBNext Ultra II Ligation Master Mix. Since the NEBNext Ultra II Ligation Master Mix is very viscous, ensure adequate mixing by pipetting up and down at least 10 times. Perform a quick spin to collect all liquid from the sides of the tube. Incubate for 15 min at 20 °C in a PCR thermocycler with the heated lid off (*see* **Note 14**).
2. Add 3 μl USER enzyme to the ligation mixture and mix by pipetting. Incubate 15 min at 37 °C in a PCR thermocycler with the heated lid set to ≥47 °C.
3. Transfer sample to 1.5 ml LoBind microcentrifuge tube. Place the tube on a magnetic rack for 2 min. Carefully remove and discard the supernatant.
4. Wash beads twice with 600 μl 1× B&W + 0.1% Triton X-100. Mix by pipetting. Place the tube on a magnetic rack for 2 min. Carefully remove and discard the supernatant.
5. Wash beads with 600 μl EB buffer. Mix by pipetting. Place the tube on a magnetic rack for 2 min. Carefully remove and discard the supernatant.
6. Resuspend beads in 40 μl EB buffer.

3.15 PCR Amplification (PCR 1)

1. Amplify the Hi-C library by setting up a first PCR reaction (PCR 1). Transfer 20 μl bead-Hi-C library suspension to a 0.2 ml PCR tube, add 25 μl 2× NEBNext Ultra II Q5 Master Mix, 2.5 μl 10 μM NEBNext Universal PCR Primer, and 2.5 μl 10 μM NEBNExt Index Primer (sample-specific). Mix by pipetting up and down at least 10 times. Perform a quick spin to collect all liquid from the sides of the tube. Store the remaining 20 μl Hi-C library bead suspension at −20 °C, for potential troubleshooting or later amplification.
2. Place in a PCR thermocycler and run the following program: 1 cycle: 3 min 98 °C (initial denaturation); 3–5 cycles: 30 s 98 °C (denaturation), 30 s 63 °C (annealing/extension), 40 s 72 °C (final extension); hold at 10 °C. The number of PCR

cycles should be chosen based on input amount and thus may need to be optimized.

3. Transfer the sample to a 1.5 ml LoBind microcentrifuge tube. Place the tube on a magnetic rack for 5 min. Transfer the supernatant containing the amplified Hi-C library to a new 1.5 ml Lo-Bind microcentrifuge tube (*see* **Note 15**).
4. Adjust the volume of the amplified Hi-C library to 50 μl with EB.

3.16 Removal of Adapter Dimers

1. To remove adapter dimers, add 45 μl well resuspended SPRI-select beads to 50 μl of amplified Hi-C library (0.9× ratio bead/sample). Pipette up and down at least 10 times to mix thoroughly. Incubate for 5 min at RT. Place the tube on a magnetic rack for 2 min. Carefully remove and discard the supernatant (*see* **Note 16**).
2. Wash the beads twice with 200 μl freshly prepared 80% ethanol while on the magnetic rack, do not resuspend. Incubate for 2 min at RT. Carefully remove and discard the supernatant.
3. After removal of all traces of ethanol with a 10 μl pipette, air-dry beads for 3–4 min while on the magnetic rack. Do not over-dry the beads, as this may result in a lower Hi-C library recovery.
4. Elute the amplified Hi-C library from the beads by adding 20 μl EB buffer. Mix thoroughly by vortex. Incubate for 5 min at 37 °C. Place the tube on a magnetic rack for 2 min. Transfer the supernatant to a 1.5 ml LoBind microcentrifuge tube.

3.17 Side qPCR

1. To reduce PCR-related artifacts during further amplification of the Hi-C library, the appropriate number of cycles for PCR 2 is determined by qPCR. Set up a qPCR reaction in a well of a qPCR plate by adding 5 μl 2× NEBNext Ultra II Q 5 Master Mix, 0.5 μl 10 μM NEBNext Universal Primer, 0.5 μl 10 μM NEBNext Index Primer (sample-specific, same as for PCR 1), 0.1 μl 100× SYBR Green I, 1.5 μl amplified (PCR 1) Hi-C library, and 2.4 μl ultrapure water.
2. Place in a Real-time PCR instrument and run using the following program: 1 cycle: 3 min 98 °C; 30 cycles: 30 s 98 °C, 30 s 63 °C, 40 s 72 °C.
3. Calculate the additional number of cycles for PCR 2 by plotting the linear Rn versus cycle number. The cycle number that corresponds to one-third of the maximum fluorescent intensity is the desired number of cycles for PCR 2. The number of cycles may vary between samples and some may not need additional amplification (*see* **Note 17**).

3.18 PCR Amplification (PCR 2)

1. If further amplification of the Hi-C library is needed, set up a second PCR reaction (PCR 2) in a 0.2 ml PCR tube by adding 25 µl 2× NEBNext Ultra II Q5 Master Mix, 2.5 µl 10 µM NEBNext Universal PCR Primer, 2.5 µl 10 µM NEBNExt Index Primer (sample-specific, same as for PCR 1), 18.5 µl amplified (PCR 1) Hi-C library and 1.5 µl ultrapure water. Mix by pipetting.
2. Place in a PCR thermocycler and run the following program: 1 cycle: 3 min 98; N cycles: 30 s 98 °C, 30 s 63 °C, 40 s 72 °C; hold at 10 °C, where N is the number of cycles calculated from the side qPCR (*see* **Note 18**).
3. Adjust the volume of the amplified Hi-C library to 50 µl with EB.

3.19 Size Selection of Amplified Hi-C Library

1. To remove adapter dimers and narrow the amplified Hi-C library size range (depends on sequencing read length), add 35 µl well resuspended SPRIselect beads to 50 µl amplified Hi-C library (0.7× ratio bead/sample, to remove <250 bp. Pipette up and down at least 10 times to mix thoroughly. Incubate for 5 min at RT. Place the tube on a magnetic rack for 2 min. Carefully remove and discard the supernatant (*see* **Note 19**).
2. Wash the beads twice with 200 µl freshly prepared 80% ethanol while on the magnetic rack, do not resuspend. Incubate for 2 min at RT. Carefully remove and discard the supernatant.
3. After removal of all traces of ethanol with a 10 µl pipette, air-dry the beads for 3–4 min while on the magnetic rack. Do not over-dry the beads, as this may result in a lower Hi-C library recovery.
4. Elute the final Hi-C library from the beads by adding 20 µl EB buffer. Mix thoroughly by vortex. Incubate for 5 min at 37 °C. Place the tube on a magnetic rack for 2 min. Transfer the supernatant to a 1.5 ml LoBind microcentrifuge tube.
5. Store the final Hi-C library at −20 °C.

3.20 Library Quality Assessment

1. Quantify the concentration of the final Hi-C library by Qubit or qPCR.
2. Analyze fragment size and molarity by Bioanalyzer using a High Sensitivity DNA Kit.

3.21 Sequencing

The final Hi-C library is sequenced using sequencing by synthesis on a standard Illumina platform. Paired-end sequencing enables both ends of the DNA fragment to be sequenced and reads of at least 100 bp are recommended. For *A. thaliana*, a minimum of 200 million paired-end sequenced reads is needed to obtain sufficient data for subsequent bioinformatic analysis steps. Higher

sequencing depth will generally result in higher resolution Hi-C interaction maps (*see* **Note 20**).

3.22 Bioinformatic Analysis

The following steps are described as if processing only one sequencing file but should be applied to each Hi-C sequenced library. For steps such as matrix quality control and correction, or differential interactions identification, sequencing of two biological replicates (independent plant material collections and Hi-C library preparations) for each sample (either from cotyledons or roots) are required.

3.22.1 Software Requirements for Data Analysis

Given the large size of Hi-C sequenced data, these analyses should be carried out in a high-performance computer suitable for genomic analysis. The first step is to install the following software on this computer. If the computer is centrally managed, there may be a protocol in place to request software installs. The person carrying out the analysis should have basic knowledge of the Unix shell Bash and the programming language R.

1. Bowtie2 (http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml#obtaining-bowtie-2).
2. HiCUP (https://www.bioinformatics.babraham.ac.uk/projects/hicup/read_the_docs/html/index.html#installation).
3. HiCExplorer (https://hicexplorer.readthedocs.io/en/latest/content/installation.html).
4. Juicer_tools (https://github.com/aidenlab/juicer/wiki/Download).
5. Cooler (https://cooler.readthedocs.io/en/latest/quickstart.html#installation).
6. Bam2pairs (https://github.com/4dn-dcic/pairix/tree/master/util/bam2pairs).
7. diffHic (https://www.bioconductor.org/packages/release/bioc/html/diffHic.html).
8. edgeR (http://bioconductor.org/packages/release/bioc/html/edgeR.html).
9. csaw (https://bioconductor.org/packages/release/bioc/html/csaw.html).
10. GenomicRanges (https://bioconductor.org/packages/release/bioc/html/GenomicRanges.html).
11. samtools (https://github.com/samtools/samtools).
12. statmod (https://cran.r-project.org/web/packages/statmod/index.html).

3.22.2 Read Alignment and Filtering

The first step in Hi-C data analysis is to align the sequencing reads to the reference genome. HiCUP is an automatic pipeline that maps and filters paired-end reads derived from Hi-C ligation products [15]. Two files must be generated to map and filter Hi-C reads with HiCUP: (1) a mapping index and (2) a digested genome file. HiCUP can use either bowtie or bowtie2 to map reads. Here, we align the reads using bowtie2 [21].

In this protocol, we use generic file names for the commands which should be substituted with the names of actual files. For example, we use *your_genome.fa*, which should be substituted by the name of the file containing the genome of interest. These generic file names are in italics in the commands below.

1. Generate a bowtie2 mapping index for the genome.

```
$ bowtie2-build --threads 1 your_genome.fa your_genome_bt2idx
```

   Output files: Six bowtie2 index files whose names start with the prefix your_genome_bt2idx and end with the suffix bt2.

2. Generate an in silico digested genome file. Indicate with the character "^" the restriction enzyme recognition site and the restriction enzyme name separated by a comma.

```
$ hicup_digester --re1 ^GATC,DpnII --genome your_genome your_-
genome.fa
```

   Output file: A text file containing the digested genome, *Digest_your_genome_DpnII.txt*.

3. Run the complete pipeline with the `hicup` command to align and filter read pairs. The index files, the digested genome and the samples files should be in the working directory, otherwise indicate the path on the command line.

```
$ hicup --bowtie2 /your/path/to/bowtie2 --index your_gen-
ome_bt2idx --digest Digest_your_genome_DpnII.txt --longest
800 --shortest 100 --threads 1 --zip cotyledon_rep1_1.fq.gz
cotyledon_rep1_2.fq.gz
```

   Output files: A bam file ready for downstream analysis, *cotyledon_rep1_1_2.hicup.bam* and an html file summarizing the pipeline results, *cotyledon_rep1_1_2.HiCUP_summary_report.html*.

   Repeat this step for every library. Note that each library is made up of two fastq files (one per read pair).

3.22.3 Obtain Pairs File from the Bam Generated with HiCUP

Convert the bam containing filtered Hi-C pairs to the pairs format used by downstream Hi-C analysis tools. The pairs file is a standard format proposed by the 4DNucleome consortium [22]. We will use the bam2pairs command to obtain a .pairs file.

1. Generate a chromosome file containing the chromosome names and lengths, separated by a tab. An example file:

```
chr1    1000000
chr2    2000000
```

   This information is generally included in the header of the alignment bam files in the @SQ fields and can be visualized using the following command.

```
$ samtools view -H cotyledon_rep1_1_2.hicup.bam | grep @SQ
```

   Output file: *chr_file.txt*

2. Convert the bam file to a .pairs file.

```
$ bam2pairs -l -c chr_file.txt cotyledon_rep1_1_2.hicup.bam
cotyledon_rep1_1_2.hicup
```

   Output file: cotyledon_rep1_1_2.hicup.bsorted.pairs.

3.22.4 Bin Read Pairs to Obtain a Contact Matrix

The next step in the workflow is to aggregate the read level pairs into bins. Once the data is binned, other formats are used to store the matrix data. We will bin and store matrices using two common tools: juicer_tools, which stores the resulting matrix in the .hic format, and cooler, which uses the .cool format. Both formats are binary containers for Hi-C data [16, 18]. When binning the matrix, a high-resolution bin size (1–10 kb) is recommended because a lower resolution (>10 kb) can be easily obtained by summing adjacent bins. The .cool files store a single matrix at a particular resolution.

1. Convert the .pairs file to a .cool file binned at 10 kb resolution.

```
$ cooler cload pairs -c1 2 -p1 3 -c2 4 -p2 5 chr_file.txt:10000
cotyledon_rep1_1_2.hicup.bsorted.pairs  cotyledon_rep1_10k.
cool
```

   Output file: *cotyledon_rep1_10k.cool*.

3.22.5 Normalize Matrices to Account for Differences in Sequencing Depth Between Samples

When it is of interest to compare multiple matrices, differences in sequencing depth between experiments must be considered. To do this, we will use the hicNormalize function from HiCExplorer to adjust the matrices, so the total sum is equal to the matrix with lower sequencing depth [17].

In the following commands, we use 10 kb binned cool files for all the samples. In this case we have cotyledon_rep1, cotyledon_rep2, root_rep1, and root_rep2.

1. Normalize matrices to make them comparable in terms of sequencing depth.

```
$ hicNormalize --matrices cotyledon_rep1_10k.cool cotyledon_rep2_10k.cool root_rep1_10k.cool root_rep2_10k.cool --normalize smallest -o cotyledon_rep1_10k_norm.cool cotyledon_rep2_10k_norm.cool root_rep1_10k_norm.cool root_rep2_10k_norm.cool
```

Output files: One cool file with the _norm.cool suffix for each sample.

3.22.6 Correct Matrices to Account for Underlying Biases

Matrix correction is necessary to account for biases such as GC content or mappability. A typical matrix transformation is the iterative correction strategy [23]. It works under the assumption that, if an experiment was unbiased, all bins should have equal visibility of contacts. Iterative correction results in a matrix where the sum of each column and each row is equal.

Bin level filtering is necessary to remove low count bins before correction. To decide filtering values for this filter, we can run a diagnostic plot of a histogram of counts per bin. It is important to remove bins with a low number of contacts.

1. Generate a histogram of counts per bin.

```
$ hicCorrectMatrix diagnostic_plot --matrix cotyledon_rep1_10k_norm.cool -o cotyledon_rep1_10k_diagnostic.png
```

Output file: a histogram of counts per bin, *cotyledon_rep1_10k_diagnostic.png*.

Inspect the diagnostic plot and choose the cutoff values (*see* **Note 21**).

2. After deciding on minimum and maximum values, proceed to correct the matrix.

```
$ hicCorrectMatrix correct --matrix cotyledon_rep1_10k_norm.cool --correctionMethod ICE --outFileName cotyledon_rep1_10k_corrected.cool --filterThreshold -2.5 5
```

Output file: *cotyledon_rep1_10k_corrected.cool*.

Repeat **steps 1** and **2** for the rest of the samples.

3.22.7 Matrix QC

When working with an experimental design with multiple conditions and replicates, it is useful to assess how similar the replicates are and how different the conditions are (*see* **Note 22**).

1. Calculate the correlation of counts between replicates and conditions.

```
$ hicCorrelate --log1p --matrices cotyledon_rep1_10k.cool cotyledon_rep2_10k.cool root_rep1_10k.cool root_rep2_10k.cool --range 20000:500000 -oh between_matrix_cor_h.png -os between_matrix_cor_s.png
```

Output files: correlation heatmap, *between_matrix_cor_h.png* and correlation scatter plot *between_matrix_cor_s.png*.

3.22.8 Sum Replicate Matrices to Increase the Resolution

A common practice in Hi-C data analysis is to sum biological replicate matrices in order to increase sequencing depth and thus matrix resolution.

1. Sum matrices.

```
$ hicSumMatrices -m cotyledon_rep1_10k.cool cotyledon_rep2_10k.cool -o cotyledon_merge_10k.cool
$ hicSumMatrices -m root_rep1_10k.cool root_rep2_10k.cool -o root_merge_10k.cool
```

Output files: *cotyledon_merge_10k.cool* and *root_merge_10k.cool* contain the merged replicates for each sample.

For comparative analyses, the normalization and correction steps from Subheadings 3.22.5 and 3.22.6 should be applied to the merged matrices. Up to this point, we obtained matrices merged, binned and normalized at 10 kb. Repeat Subheadings 3.22.4 through 3.22.7, changing the resolution value to obtain matrices in other resolutions. In the following sections, we will be working with matrices with 50 kb and 500 kb resolution.

3.22.9 Build a .hic Matrix

Another Hi-C storage format is the .hic format, used by juicer, juicer_tools and juicebox [18]. It is a binary format that stores a Hi-C matrix with multiple bin sizes and corrections in a single file. The `juicer_tools pre` command bins and corrects the Hi-C matrix at several resolutions.

1. Generate the .hic matrix.

```
$ java -Xmx1G -jar /path/to/your/juicer_tools_1.13.02.jar pre cotyledon_rep1_1_2.hicup.bsorted.pairs cotyledon_rep1.hic chr_file.txt
```

Output file: The *cotyledon_rep1.hic* file stores the matrix at various resolutions, as well as different corrections. This file can be directly uploaded to juicebox for visualization and is ready to use with juicer_tools, which we will do in Subheading 3.22.13.

3.22.10 Visualize the Hi-C Matrices

Use the HiCExplorer command-line tools to generate visualizations of Hi-C matrices [17]. In this section, we are using matrices merged, binned, and corrected at 50 and 500 kb resolution.

1. Plot a large region.

```
$ hicPlotMatrix --perChromosome --log1p --matrix cotyledon_-
merge_500kb_corrected.cool --outFileName cotyledon_mer-
ge_500kb_corrected.png
```

Output file: *cotyledon_merge_500kb_corrected.png* is a plot of the corrected matrix per chromosome.

2. Plot a small region.

```
$ hicPlotMatrix --log1p --region 2:125000000-130000000 --ma-
trix cotyledon_merge_50kb_corrected.cool --outFileName coty-
ledon_merge_50kb_corrected.png
```

Output file: *cotyledon_merge_50kb_corrected.png* is a plot showing the region 2:125000000-130000000.

3. Obtain and plot a matrix containing the differences between conditions.

```
$ hicCompareMatrices --operation log2ratio --matrices coty-
ledon_merge_500kb_corrected.cool root_merge_500kb_corrected.
cool --outFileName root_cotyledon_500kb_log2.cool
```

Output file: *root_cotyledon_500kb_log2.cool* is a cool file containing the difference between root and cotyledon.

```
$ hicPlotMatrix --perChromosome --matrix root_cotyledon_500k-
b_log2.cool --outFileName root_cotyledon_500kb_log2.png
```

Output file: *root_cotyledon_500kb_log2.png* is a plot per chromosome of the difference matrix.

3.22.11 Identify A/B Compartments

Intrachromosomal contacts are segregated into transcriptionally active (A compartment) and inactive (B compartment) regions [7]. This is generally done using principal component analysis (PCA), a method that reduces the global interaction patterns to a single vector, or principal component, that captures most of the

variability between compartments. Each bin is labeled as either A or B compartment based on the sign of the principal component value for that bin (*see* **Note 23**).

1. Obtain the principal component to define compartments.

```
$ hicPCA -noe 1 --matrix cotyledon_merge_500kb_corrected.cool
--format bigwig -o cotyledon_merge_500kb_pca1.bw
```

Output file: *cotyledon_merge_500kb_pca1.bw* is a bigWig file containing the PC1 of the matrix.

2. Plot the first principal component along with the matrix.

```
$ hicPlotMatrix -m cotyledon_merge_500kb_corrected.cool -o
cotyledon_merge_compartments.png --log1p --bigwig cotyledon_-
merge_500kb_pca1.bw --perChromosome
```

Output file: *cotyledon_merge_compartments.png* is a plot of the whole genome matrix per chromosome along with the PC1.

3. If histone modification information is available, for example, H3K4me3, which is generally associated with active transcription (A compartment), this information can be displayed alongside the compartments.

```
$ hicPlotMatrix --log1p -m cotyledon_merge_500kb_corrected.
cool -o cotyledon_merge_500kb_histonemod.png --perChromosome
--bigwig cotyledon_H3K4me3.bw
```

Output file: *cotyledon_merge_500kb_histonemod.png* is a plot showing the whole genome matrix per chromosome together with the H3K4me3 signal.

3.22.12 Identify TADs

Topologically Associating Domains are defined as regions of increased self-interaction in Hi-C maps [6]. Several computational approaches have been developed to identify them. Here, we use hicFindTADs from HiCExplorer to identify TADs [17].

1. Identify TADs with hicFindTADs.

```
$ hicFindTADs -m cotyledon_merge_50kb_corrected.cool --out-
Prefix cotyledon_merge_50kb_tads --correctForMultipleTesting
fdr
```

Output files: *cotyledon_merge_50kb_tads* is a directory containing a list of domain boundaries in bed and gff formats, a list of domains in bed format, the TAD separation score in

bedgraph and a z score matrix calculated during the TAD calling procedure.

2. The hicPlotTADs function requires a file with the track information. The extension of this track configuration file is ".ini". Generate a .ini configuration file for TAD visualization. Copy the information below, between the hashes (#) to a text file called hic_tads.ini.

```
#######
[x-axis]
where=top
[hic matrix]
file = cotyledon_merge_50kb_tads
title = Hi-C data
depth = 1000000
transform = log1p
file_type = hic_matrix
[tads]
file = cotyledon_merge_50kb_tads_domains.bed
file_type = domains
border color = black
overlay previous = share-y
[spacer]
[tad score]
file = cotyledon_merge_50kb_tads_tad_score.bedgraph
title = "TAD separation score"
file_type = bedgraph
#######
```

3. Visualize TADs alongside the matrix.

```
$ hicPlotTADs --tracks hic_tads.ini -o cotyledon_50k_tads.png
--region 2:122000000-126000000
```

Output file: *cotyledon_50k_tads.png* is a plot showing the TADs and TAD separation score along the matrix focused in the region from position 122,000,000 to 126,000,000 of chromosome 2.

3.22.13 Identify Interaction Peaks

Interaction peaks are regions of high interaction frequency between two distant genomic regions. A standard tool for peak calling is HiCCUPs, available from the juicer_tools toolkit [18]. To use this tool, we need the .hic matrix generated previously in Subheading 3.22.9.

1. Identify peaks using HiCCUP.

```
$ java -jar /usr/local/src/juicer/juicer_tools_1.13.02.jar
hiccups --cpu --threads 2 -r 10000 cotyledon_rep1.hic -k KR
cotyledon_hiccups_loops
```

Output files: *cotyledon_hiccups_loops* is a directory containing the merged_loops file containing the final list of identified loops. Intermediate processing files will also be saved in this output directory.

2. Generate an aggregated peak plot. This plot is useful to get an overview of all the peaks at once.

```
$ java -jar /usr/local/src/juicer/juicer_tools_1.13.02.jar apa
-r 10000 cotyledon_rep1.hic cotyledon_hiccups_loops cotyle-
don_hiccups_apa
```

Output files: the *cotyledon_hiccups_apa* directory will contain an APA.png file with the aggregate signal across all loops.

3.22.14 Identifying Statistically Significant Differential Interactions

An additional strategy when analyzing chromatin conformation data is to identify changes in interaction intensity that are statistically significant between two or more biological conditions. Various publicly available tools identify these Differential Interactions (Dis) from Hi-C data, such as FIND [24], HOMER [25], and HiBrowse [26]. We will identify differential interactions using the diffHic package [19].

1. Sort bam files by read name. Do this for each bam file before going to the next step.

```
$ samtools sort -n my_bam_file.bam > my_bam_file.hicup.sorted.
bam ; done
```

In the following steps, use the R console.

2. Load required libraries.

```
> Packages <- c("diffHic", "GenomicRanges", "edgeR", "csaw")

> lapply(Packages, library, character.only=T)
```

3. Import HiCUP file with digested genome into R (generated in Subheading 3.22.2).

```
> digest <- read.csv("Digest_your_genome_DpnII.txt.", head-
er=T, sep="\t", skip=1)
```

4. Generate the object hic_experiment.frag with the digested genome in the format required by diffHic.

```
> hic_experiment.frag <- with(digest,GRanges(Chromosome, IR-
anges(Fragment_Start_Position,Fragment_End_Position)))
```

5. Generate a pairParam object to store the fragments and other parameters.

```
> hic_experiment.param <- pairParam(hic_experiment.frag)
```

6. Create h5 files to count Hi-C reads into bins. This process matches the mapping location of each read to a restriction fragment in the reference genome.

```
# Cotyledon samples.

> preparePairs("cotyledon_rep1_1_2.hicup.sorted.bam", hic_experiment.param, file="cotyledon1.h5")
> preparePairs("cotyledon_rep2_1_2.hicup.sorted.bam", hic_experiment.param, file="cotyledon2.h5")

# Root samples.

> preparePairs("root_rep1_1_2.hicup.sorted.bam", hic_experiment.param, file="root1.h5")
> preparePairs("root_rep2_1_2.hicup.sorted.bam", hic_experiment.param, file="root2.h5")

# Generate input object.

> input <- c("cotyledon1.h5","cotyledon2.h5","root1.h5","root2.h5")
```

7. Count reads that fall within each genomic bin. Choose a bin size and count read pairs between paired bins for the four libraries with squareCounts using input, contained in the hic_experiment_data object.

```
> bin.size <- 50000
> hic_experiment_data <- squareCounts(input, hic_experiment.param , width=bin.size, filter=1)
```

The following steps are necessary to filter noninformative bin pairs.

8. Plot the log-NBmean-per-million (average abundance).

```
> ave.ab <- aveLogCPM(asDGEList(hic_experiment_data))
> hist(ave.ab, xlab="Average abundance", col="powderblue")
```

9. One filtering strategy is to keep only bin pairs with abundances x-times higher (3 in this example) than the median abundance across interchromosomal bin pairs, as the majority of these represent false interactions.

```
> direct <- filterDirect(hic_experiment_data)
> direct.keep <- direct$abundances > log2(3) + direct$thresh-
old
> summary(direct.keep)
> log2(3) + direct$threshold
```

10. Apply filter to data object. This will eliminate all rows that are not named in the object `direct.keep`.

```
> hic_experiment_data <- hic_experiment_data[direct.keep, ]
```

11. Visualize filtered data.

```
> ave.ab <- aveLogCPM(asDGEList(hic_experiment_data))
> hist(ave.ab, xlab="Average abundance", col="blue")
```

diffHic allows for various library normalization strategies, the simplest being library size normalization. In the next steps, we will use nonlinear normalization (LOESS) to account for trended biases between libraries.

12. Compare one library of each sample group using an MA plot (in this case 1 and 4). The fitted line on the plot shows that there is an abundance-dependent trend.

```
> ab <- aveLogCPM(asDGEList(hic_experiment_data))
> o <- order(ab)
> adj.counts <- cpm(asDGEList(hic_experiment_data), log=TRUE)
> mval <- adj.counts[,1]-adj.counts[,4]
> smoothScatter(ab, mval, xlab="A", ylab="M", main="Cotyledon
vs Root")
> fit <- loessFit(x=ab, y=mval)
> lines(ab[o], fit$fitted[o], col="red")
```

13. Apply normalization.

```
> hic_experiment_data <- normOffsets(hic_experiment_data)
```

14. Store the matrix of offsets in a separate object.

```
> nb.off <- assay(hic_experiment_data, "offset")
```

15. Adjust the log counts with the offsets and generate another MA plot to evaluate the normalization. We should see that the trend is removed.

```
> ab <- aveLogCPM(asDGEList(hic_experiment_data))
> o <- order(ab)
> adj.counts <- log2(assay(data) + 0.5) - nb.off/log(2)
> mval <- adj.counts[,1]-adj.counts[,4]
> smoothScatter(ab, mval, xlab="A", ylab="M", main="Cotyledon
vs Root after NLN")
> fit <- loessFit(x=ab, y=mval)
> lines(ab[o], fit$fitted[o], col="red")
```

16. Create a design matrix that describes the experimental setup. In this case we have two conditions (cotyledon and root) with two replicates each.

```
> design <- model.matrix(~factor(c("cotyledon", "cotyledon",
"root", "root")))
> colnames(design) <- c("Intercept", "root")
```

17. Convert the `hic_experiment_data` object to a DGEList object to analyze it with edgeR.

```
> y <- asDGEList(hic_experiment_data)
```

18. The variability between replicates of the same condition is estimated using the dispersion parameter of the Negative Binomial (NB) distribution. Estimate the dispersion.

```
> y <- estimateDisp(y, design)
> plotBCV(y)
```

19. Estimate the Quasi-likelihood dispersion.

```
> fit <- glmQLFit(y, design, robust=TRUE)
> plotQLDisp(fit)
```

20. Identify differential interactions with the quasi-likelihood *F*-test. This test will evaluate the statistical significance of each differential interaction and provide a *p*-value and an adjusted *p*-value, or false discovery rate (FDR) for each of them.

```
> result <- glmQLFTest(fit, coef=2)
> topTags(result)
```

21. Save significance statistics in the variable rowData of the InteractionSet object.

```
> rowData(hic_experiment_data) <- cbind(rowData(hic_experi-
ment_data), result$table)
```

22. Plot the total of differential interactions in a smear MA plot.

```
> de <- decideTestsDGE(result, p.value=0.05, adjust.method="BH")
> debins <- rownames(result)[as.logical(de)]
> plotSmear(result, de.tags=debins)
```

23. We can cluster those bin pairs that are adjacent and significant to avoid redundancy so that each cluster contains only statistically significant bins that correspond to a differential interaction. Here, we are clustering bins that are right next to one another (tol = 1).

```
> clustered.sig <- diClusters(hic_experiment_data, result$table, target=0.05, cluster.args=list(tol=1))
> length(clustered.sig$interactions)
> head(clustered.sig$interactions)
> clustered.sig$FDR
```

24. Using the indices of the bin pairs we can use the combineTests function to calculate the combined p-value for the cluster.

```
> tabcomdata <- combineTests(clustered.sig$indices[[1]], result$table)
> head(tabcomdata)
```

25. Using the same indices, we can also use getBesetTest to identify the bin pair with the most significant p-value within a cluster.

```
> tabbestdata <- getBestTest(clustered.sig$indices[[1]], result$table)
> head(tabbestdata)
```

26. Save the coordinates and statistics for each differential interaction.

```
> tabstat <- data.frame(tabcomdata[,,], logFC=tabbestdata$logFC, FDR=clustered.sig$FDR)
> result.d <- as.data.frame(clustered.sig$interactions)[,c("seqnames1","start1","end1","seqnames2","start2","end2")]
> result.d <- cbind(result.d, tabstat)
> o.d <- order(result.d$PValue)
> write.table(result.d[o.d,], file="DI_ClustersData.tsv", sep="\t", quote=FALSE, row.names=FALSE)
```

4 Notes

1. When preparing NIB, NIB-FA and NIB-P, add all the components in the indicated order. PMSF, 2-mercaptoethanol, and formaldehyde should be added prior to use, under a fume hood. Prepare the protease inhibitor cocktail using cOmplete ULTRA Tablets (Roche) and add prior to use.
2. It is recommended to collect plant material quickly and cross-link with formaldehyde immediately. In this case, harvesting and separating organs of approximately 6900 seedlings is done in maximum 20 min, involving two persons.
3. Formaldehyde is oxidized to formic acid under normal atmospheric oxygen concentrations. Therefore, preferably use pure, methanol-free, ampule-sealed formaldehyde solution. Opened ampules should be resealed using Parafilm and stored at 4 °C for no longer than a week. Poor-quality formaldehyde will adversely affect the experiment.
4. To avoid air bubbles, add NIB-P slowly to the sample. To prevent chromatin degradation, precool 50 ml centrifuge tubes, 1.5 ml microcentrifuge tubes, and NIB-P buffer on ice. Always keep the samples on ice.
5. Incubation with SDS will increase chromatin accessibility for better restriction digestion and inactivation of endogenous nucleases. The duration of incubation may need to be optimized in a sample dependent manner. Shorter incubation time may result in inefficient or partial digestion due to the chromatin being inaccessible to the restriction enzyme. Longer incubation time may lead to excessive digestion, alteration of chromatin territories and may even reverse cross-links [14].
6. It is important to maintain a 6–10× ratio Triton X-100/SDS, as nonadequate SDS quenching can inhibit the enzymatic activities of the enzymes used in downstream steps.
7. Hi-C experiments can be done using 6-cutter (i.e., HindIII) or 4-cutter (i.e., DpnII) restriction enzymes, the latter generating genome-wide chromosomal contact maps with higher resolution [27]. The MboI 4-cutter enzyme, that recognizes the same sequence as DpnII, has been used in plants such as rice, foxtail millet, sorghum, tomato, and maize [28]. When using different enzymes, concentrations and incubation times may need to be optimized.
8. The method of enzyme inactivation is specific for DpnII. Use appropriate inactivation conditions if using other enzymes.
9. Thaw 10 μl control sample, add 82 μl EB buffer and 1 μl 20 mg/ml RNase A. Incubate for 30 min at 37 °C. Add 5 μl 10% SDS and 2 μl 20 mg/ml proteinase K. Incubate overnight

at 37 °C and 6 h at 65 °C. Before use, spin Phase Lock Gel (Quantabio) tube at 12,000 × *g* for 30 s at RT. Transfer sample to prespun Phase Lock Gel tube. Add 100 μl EB buffer and 200 μl phenol–chloroform–isoamyl alcohol (25:24:1). Mix thoroughly by vigorous shaking for 2 min to form a transiently homogenous suspension, do not vortex. Centrifuge at 12,000 × *g* for 5 min at RT. Perform a second extraction, by adding 200 μl chloroform to the same Phase Lock Gel tube. Mix thoroughly by vigorous shaking for 2 min, do not vortex. Centrifuge at 12,000 × *g* for 5 min at RT. Transfer DNA-containing aqueous upper phase to a 1.5 ml microcentrifuge tube. Precipitate the DNA by adding 1/10 volume 3 M sodium acetate pH 5.2, 1 μl glycogen, and 2.5 volumes ice-cold 100% ethanol. Mix by inverting 5 times and incubate at −80 °C for 1 h. Centrifuge at 20,000 × *g* for 20 min at 4 °C. Remove the supernatant. Wash the pellet twice with 1 ml 70% ethanol. Centrifuge at 20,000 × *g* for 5 min at RT. Remove the supernatant and air-dry pellet. Dissolve DNA pellet in 10 μl EB buffer. Measure DNA concentration using a NanoDrop spectrophotometer. Load 500 ng of sample and a 1 Kb Plus DNA Ladder on a 1.8% agarose gel for electrophoresis. Run the gel for 40 min at 90 V. Undigested (or intact) chromatin will be seen as a tight high molecular weight band (>10 kb), while digested chromatin will appear as a smear from 100 bp to 3 kb approximately. For rapid reversal of chromatin cross-linking, thaw 10 μl of control sample, add 83 μl EB buffer, 4 μl 5 M NaCl, 2 μl 20 mg/ml proteinase K, and 1 μl 20 mg/ml RNase A. Incubate for 1 h at 65 °C. Proceed as above with DNA extraction using Phase Lock Gel tubes.

10. The temperature and incubation time of the ligation reaction can be optimized.
11. Inefficient removal of biotin from unligated ends can lead to sequencing of reads from unwanted dangling-end products and not from real interactions [14].
12. DNA shearing target length depends on sequencing strategy and sequencing read length. The shearing settings depend on the equipment used and can be optimized.
13. Samples can be stored at −20 °C.
14. The appropriate adaptor concentration may need to be optimized depending on sample input amount.
15. The original Hi-C library is bound to the magnetic beads as a single biotinylated strand of the hybrid molecule. After PCR 1 the beads can be resuspended in 200 μl of EB buffer and stored at 4 °C for later troubleshooting. If needed, the Hi-C library can be amplified again setting up a new PCR reaction on these beads.

16. It is recommended to remove primers (<85 bp) and adaptor dimers (~127 bp) from the amplified Hi-C library. Due to their short size, the latter may be preferentially amplified during the following qPCR based library quantification, impairing the accurate determination of the library concentration.
17. This step helps minimize PCR-related artifacts, such as over-amplification and reduced library complexity, or GC and size bias, during further Hi-C library amplification [29].
18. The Hi-C library amplification should provide sufficient fragments for high-throughput sequencing, minimizing PCR-related artifacts.
19. The amplified Hi-C library size range depends on the sequencing strategy. For 2x150 bp (paired-end) Illumina sequencing the optimal library fragment size (insert +120 bp adaptors) is ~320–520 bp. A double size selection with SPRIselect beads may be performed if removing larger fragments (>600 bp) is needed for narrowing the library size. Importantly, it is recommended to remove primers (<85 bp) and adaptor dimers (~127 bp) from the library. Primers cannot cluster or be sequenced but can bind to the flow cell and reduce cluster density. On the other hand, adapter dimers will cluster and be sequenced if present in the library. The beads from all steps can be stored in 80% ethanol at 4 °C for later troubleshooting.
20. If other plant species are used, the sequencing depth has to be adjusted according to the genome size. It is worth noting that the resolution obtained in Hi-C does not vary linearly with genome size (our observations). Thus, it is recommended to use a higher sequencing depth than inferred by extrapolating the number of recommended reads for the *A. thaliana* genome.
21. The histogram of counts per bin should show two modes in the distribution, the first one around zero and the second one around the mean number of contacts per bin. To filter bins with low counts, the lower threshold value selected should be in the valley between the zero and the mean. The upper threshold should be selected based on the upper bound of the counts distribution. Note that the diagnostic plot will include suggested values for each threshold.
22. Replicates should have a higher correlation than conditions. It can be useful to make this analysis with different bin sizes to see if the correlation holds, as higher resolutions may be noisier. Correlation analysis of Hi-C counts is challenging because, as the distance between interacting bins increases, the average counts decrease and are more variable. For this reason, hicCorrelate has the --range option, to limit the distance range of the comparison. Other correlation strategies explicitly designed for

Hi-C data that take into account the distance effect are discussed in [30].

23. An alternative approach for compartment identification in *A. thaliana* is to analyze each chromosomal arm separately [31]. This is because performing principal component analysis in the whole chromosome generally identifies only three compartments, separating the euchromatic arms from the heterochromatin. Excluding the pericentromeric region improves the identification of informative subcompartments on the chromosome arms.

Acknowledgments

FJP-d, JES-F, and AR-C were funded by fellowships from the Consejo Nacional de Ciencia y Tecnología (CONACYT). N-HW, SF-V, and KO are funded by the Newton Advanced Fellowship (No. NAF\R1\180303) awarded to SF-V. KO is supported by the Consejo Nacional de Ciencia y Tecnología (CONACYT, CB-2016-01/285847).

References

1. Gibcus JH, Dekker J (2013) The hierarchy of the 3D genome. Mol Cell 49:773–782. https://doi.org/10.1016/j.molcel.2013.02.011
2. Felsenfeld G, Groudine M (2003) Controlling the double helix. Nature 421:448–453. https://doi.org/10.1038/nature01411
3. Deng W, Lee J, Wang H et al (2012) Controlling long-range genomic interactions at a native locus by targeted tethering of a looping factor. Cell 149:1233–1244. https://doi.org/10.1016/j.cell.2012.03.051
4. Kleinjan DA, van Heyningen V (2005) Long-range control of gene expression: emerging mechanisms and disruption in disease. Am J Hum Genet 76:8–32. https://doi.org/10.1086/426833
5. Dixon JR, Selvaraj S, Yue F et al (2012) Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature 485:376–380. https://doi.org/10.1038/nature11082
6. Nora EP, Lajoie BR, Schulz EG et al (2012) Spatial partitioning of the regulatory landscape of the X-inactivation centre. Nature 485:381–385. https://doi.org/10.1038/nature11049
7. Lieberman-Aiden E, van Berkum NL, Williams L et al (2009) Comprehensive mapping of long-range interactions reveals folding principles of the human genome. Science 326:289–293. https://doi.org/10.1126/science.1181369
8. Cremer T, Cremer C (2006) Rise, fall and resurrection of chromosome territories: a historical perspective. Part I. The rise of chromosome territories. Eur J Histochem 50:161–176. https://www.ncbi.nlm.nih.gov/pubmed/16920639
9. Doğan ES, Liu C (2018) Three-dimensional chromatin packing and positioning of plant genomes. Nat Plants 4:521–529. https://doi.org/10.1038/s41477-018-0199-5
10. Dekker J, Marti-Renom MA, Mirny LA (2013) Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. Nat Rev Genet 14:390–403. https://doi.org/10.1038/nrg3454
11. Eagen KP (2018) Principles of chromosome architecture revealed by Hi-C. Trends Biochem Sci 43:469–478. https://doi.org/10.1016/j.tibs.2018.03.006
12. Hövel I, Louwers M, Stam M (2012) 3C technologies in plants. Methods 58:204–211. https://doi.org/10.1016/j.ymeth.2012.06.010
13. Liu C (2017) In situ hi-C library preparation for plants to study their three-dimensional

chromatin interactions on a genome-wide scale. Methods Mol Biol 1629:155–166. https://doi.org/10.1007/978-1-4939-7125-1_11

14. Padmarasu S, Himmelbach A, Mascher M et al (2019) In situ hi-C for plants: an improved method to detect long-range chromatin interactions. Methods Mol Biol 1933:441–472. https://doi.org/10.1007/978-1-4939-9045-0_28
15. Wingett S, Ewels P, Furlan-Magaril M et al (2015) HiCUP: pipeline for mapping and processing Hi-C data. F1000Res 4:1310. https://doi.org/10.12688/f1000research.7334.1
16. Abdennur N, Mirny LA (2020) Cooler: scalable storage for Hi-C data and other genomically labeled arrays. Bioinformatics 36: 311–316. https://doi.org/10.1093/bioinformatics/btz540
17. Ramírez F, Bhardwaj V, Arrigoni L et al (2018) High-resolution TADs reveal DNA sequences underlying genome organization in flies. Nat Commun 9:189. https://doi.org/10.1038/s41467-017-02525-w
18. Durand NC, Shamim MS, Machol I et al (2016) Juicer provides a one-click system for analyzing loop-resolution Hi-C experiments. Cell Syst 3:95–98. https://doi.org/10.1016/j.cels.2016.07.002
19. Lun ATL, Smyth GK (2015) diffHic: a Bioconductor package to detect differential genomic interactions in Hi-C data. BMC Bioinformatics 16:258. https://doi.org/10.1186/s12859-015-0683-0
20. Dong P, Zhong S (2020) Characterization of plant 3D chromatin architecture, in situ Hi-C library preparation, and data analysis. Methods Mol Biol 2093:147–167. https://doi.org/10.1007/978-1-0716-0179-2_11
21. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with bowtie 2. Nat Methods 9: 357–359. https://doi.org/10.1038/nmeth.1923
22. Lee S, Bakker CR, Vitzthum C et al (2022) Pairs and Pairix: a file format and a tool for efficient storage and retrieval for Hi-C read pairs. Bioinformatics, 38:1729–1731
23. Imakaev M, Fudenberg G, McCord RP et al (2012) Iterative correction of Hi-C data reveals hallmarks of chromosome organization. Nat Methods 9:999–1003. https://doi.org/10.1038/nmeth.2148
24. Djekidel MN, Chen Y, Zhang MQ (2018) FIND: difFerential chromatin INteractions detection using a spatial Poisson process. Genome Res 28:412–422. https://doi.org/10.1101/gr.212241.116
25. Heinz S, Benner C, Spann N et al (2010) Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. Mol Cel 38:576–589. https://doi.org/10.1016/j.molcel.2010.05.004
26. Paulsen J, Sandve GK, Gundersen S et al (2014) HiBrowse: multi-purpose statistical analysis of genome-wide chromatin 3D organization. Bioinformatics 30:1620–1622. https://doi.org/10.1093/bioinformatics/btu082
27. Wang C, Liu C, Roqueiro D et al (2015) Genome-wide analysis of local chromatin packing in *Arabidopsis thaliana*. Genome Res 25: 246–256. https://doi.org/10.1101/gr.170332.113
28. Dong P, Tu X, Chu P-Y et al (2017) 3D chromatin architecture of large plant genomes determined by local a/B compartments. Mol Plant 10:1497–1509. https://doi.org/10.1016/j.molp.2017.11.005
29. Buenrostro JD, Wu B, Chang HY et al (2015) ATAC-seq: a method for assaying chromatin accessibility genome-wide. Curr Protoc Mol Biol 109:21.29.1–21.29.9. https://doi.org/10.1002/0471142727.mb2129s109
30. Yardımcı GG, Ozadam H, Sauria MEG et al (2019) Measuring the reproducibility and quality of Hi-C data. Genome Biol 20:57. https://doi.org/10.1186/s13059-019-1658-7
31. Grob S, Schmid MW, Grossniklaus U (2014) Hi-C analysis in Arabidopsis identifies the KNOT, a structure with similarities to the flamenco locus of Drosophila. Mol Cell 55: 678–693. https://doi.org/10.1016/j.molcel.2014.07.009

Chapter 14

Isolation of *Boechera stricta* Developing Embryos for Hi-C

Mariana Tiscareño-Andrade, Katarzyna Oktaba, and Jean-Philippe Vielle-Calzada

Abstract

The possibility of analyzing chromatin topology in developing plant embryos is hampered by inaccessibility of the embryo sac, deeply embedded in the maternal seed tissue, following double fertilization. Here we describe a protocol to isolate, purify, and prepare developing *Boechera stricta* embryos for chromosome conformation capture-based methods as in situ Hi-C experiments. Early globular embryos can be isolated by air-pressure microaspiration, and subsequently washed to eliminate residual cells from the endosperm and maternal seed coat, allowing for pure sampling of selected stages of embryogenesis. This protocol allows for the possibility of comparing genome topology during plant embryonic differentiation since early until late embryo development stages.

Key words Flowering plants embryo development, Embryo isolation, Apomixis, *Boechera stricta*, Chromosome conformation capture, Genome topology, Hi-C

1 Introduction

The three-dimensional organization of chromatin in its native nuclear state has a role in modulating gene expression regulation [1]. Methods for analyzing chromatin interactions, such as the high-throughput chromosome conformation capture-based assay Hi-C, offer new opportunities to characterize three-dimensional organization of the genome in the nucleus in homogenous or heterogenous cellular constituents. Using next-generation sequencing, Hi-C identifies chromatin interactions in an unbiased way at a genome-wide scale [2, 3]. Analyses of these interactions reveal features of genome topology at different levels of organization, including chromatin loops, topologically associating domains, active or inactive compartments, and chromosome territories [4–7].

Little is known about changes in genome topology during plant embryo development. This is mostly due to the difficulty involved in the isolation of tissues that are deeply embedded within

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_14,

the developing seed. In flowering plants, the embryo is covered surrounded by the endosperm and progressively covered by maternal tissues that constitute the seed coat, which are genetically different from the embryo [8]. In *Arabidopsis thaliana* and *Boechera stricta*, both diploid and phylogenetically close members of the *Brassicaceae* family [9, 10], embryogenesis entails cellular divisions that initiate in the zygote and finish in the mature embryo, reaching approximately 20,000 cells at the end of development [9]. Whereas the most common Hi-C protocols require large amounts of tissue as starting material [7, 11], new approaches have adapted procedures to allow Hi-C library construction based on a reduced number of cells. Recent Hi-C protocols have taken advantage of fluorescence-activated cell sorting (FACS) and isolation of nuclei tagged in specific cell types (INTAC) to develop procedures that attempt exquisite preservation of DNA integrity in minute amounts of cells [1, 3].

Here we describe a recently established protocol that allows the isolation, purification, and preparation of *B. stricta* embryos to perform chromatin conformation capture experiments as low-input in situ Hi-C. Whereas early embryo isolation largely follows *Arabidopsis* procedures described in Raissig et al. [12], the embryo tissue fixation is based on the procedure established by Chang et al. [13], reaching an isolation rate of approximately 50 globular embryos per hour. The protocol has been successfully used to isolate and prepare samples comprising early globular to mature stages (average of 32 to 20,000 cells, respectively), to generate Hi-C libraries from approximately 39,000 cells at different stages of embryo development (see chapter Pérez-de los Santos et al. "Plant in-situ Hi-C experimental protocol and bioinformatic analysis" in this volume).

2 Materials

2.1 Embryo Fixation, Isolation, and Purification

2.1.1 Seeds Collection for Isolation of Early-, Middle-, and Late-Stage Embryos

1. TE buffer: 10 mM Tris–HCl pH 8.0, 1 mM EDTA pH 8.0 (*see* **Notes 1** and **2**).
2. Hypodermic insulin needles.
3. Forceps.
4. Double-sided adhesive tape.
5. Slides.
6. Stereomicroscope.
7. Eppendorf LoBind 1.5 mL microcentrifuge tubes.

2.1.2 Formaldehyde Cross-Linking

1. MC buffer: 10 mM potassium phosphate pH 7.0, 50 mM sodium chloride, 0.1 M sucrose, 0.5% Triton X-100 (*see* **Note 3**).
2. 37% formaldehyde solution (*see* **Note 4**).
3. Vacuum chamber.

2.1.3 Stopping of Cross-Linking

1. MC buffer with 0.15 M glycine (*see* **Note 5**).
2. Vacuum chamber.

2.1.4 Seed and Embryo Washing

1. Ultrapure water.
2. TE buffer.
3. Liquid nitrogen.

2.1.5 Early-Stage Embryo Isolation

1. Micropestle (*see* **Note 6**).
2. Multiwell glass microscopic slides (*see* **Note 7**).
3. Glass microcapillaries (*see* **Note 8**).
4. Puller to generate the desired dimeter of the microcapillary tip. We recommend Narishige Dual-Stage Glass Micropipette Puller.
5. Inverted microscope. We recommend one with 10–40× magnification objectives.
6. Micromanipulator with integrated joysticks. We recommend the Eppendorf TransferMan® 4r and InjectMan® 4 micromanipulators with joysticks for precise movement control in X, Y, Z, and X/Z axes.
7. Pneumatic microinjector. We recommend the pneumatic microinjector Narishige IM-11-2.
8. TE buffer.
9. Eppendorf LoBind 1.5 mL microcentrifuge tubes.
10. Liquid nitrogen.

3 Methods

3.1 Embryo Collection, Fixation, Isolation, and Purification

3.1.1 Seeds Collection for Isolation of Early-, Middle-, and Late-Stage Embryos

1. Collect siliques at the desired developmental stage, place them on slides with the double-slide tape under the stereomicroscope (*see* **Note 9**).
2. Using forceps and hypodermic insulin needles, gently open the siliques from the replum and separate the valves. Also separate the seeds manually from the funiculus with the needles.
3. For early-stage embryos, gently make an incision at the chalazal zone of the seed and collect the rest of the micropylar seed zone where the early embryo is placed. For middle- and late-stage

embryos, as heart, torpedo, and mature embryos, it is necessary to remove the endosperm and seed coat manually, using forceps, and then collect the embryo (*see* **Note 10**).

4. Place the embryos in Eppendorf LoBind microcentrifuge tubes and resuspend each sample in 20–40 μL of TE buffer (*see* **Note 11**).

3.1.2 Formaldehyde Cross-Linking

1. Add 150–300 μL of MC buffer with formaldehyde (1% final concentration), immersing the tissue completely in the solution (*see* **Note 12**).
2. For early-stage embryos, incubate the seeds for 10 min at room temperature (RT) without vacuum, gently and frequently inverting the tube. Make sure that the sepcimens are always immersed in the buffer. For middle- and late-stage embryos, including heart, torpedo, and mature embryos, incubate for 15 min at RT with vacuum infiltration, and subsequently release the vacuum at an extremely low rate.
3. Centrifuge at 5000 × *g* for 10 s at RT. Discard supernatant.

3.1.3 Stopping of Cross-Linking

1. Add 150–300 μL of MC buffer with 0.15 M glycine.
2. For early-stage embryos, incubate the seeds for 10 min at RT without vacuum, gently and frequently inverting the tube. Make sure that the specimens are always immersed in the buffer. For middle- and late-stage embryos, including heart, torpedo, and mature embryos, incubate for 15 min at RT with vacuum infiltration, and subsequently release the vacuum at an extremely low rate.
3. Centrifuge at 5000 × *g* for 10 s at RT. Discard supernatant.

3.1.4 Sample Washing

1. To wash the sample, add 150–300 μL of ultrapure water and incubate for 10 min at RT, gently and frequently inverting the tube.
2. Centrifuge at 5000 × *g* for 10 s at RT. Discard the supernatant and resuspend in 10 μL of TE buffer.
3. Early-stage seeds can be stored at 4 °C until the desired amount is collected. Flash-freeze middle- and late-stage embryos in liquid nitrogen, and store at −80 °C until further processing (*see* **Note 13**).

3.1.5 Early-Stage Embryo Isolation and Purification

1. Early-stage seeds stored in TE buffer are gently crushed with a previously sterilized micropestle. Rinse the micropestle with 150 μL of TE buffer to wash off sticky tissue as much as possible (*see* **Note 14**).
2. Resuspend the sample by gently pipetting twice up and down. Using an inverted microscope, (*see* Fig. 1a) carefully place 10–25 μL of the seed extract in one well of the multiwell glass microscopic slide (*see* Fig. 1b).

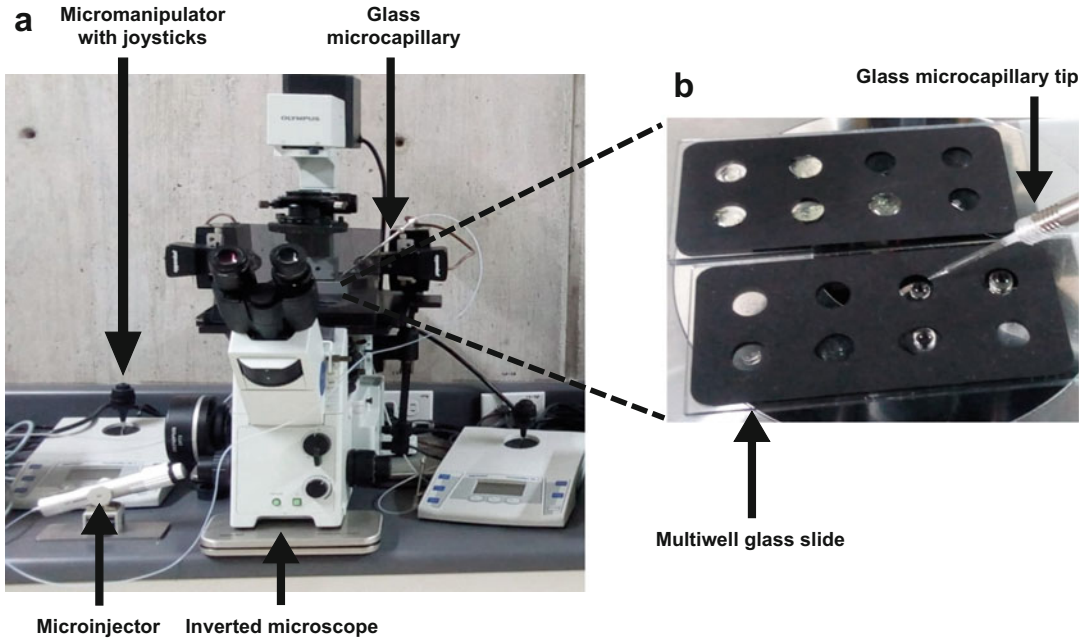


Fig. 1 Workstation for early-stage embryo isolation. (**a**) Inverted microscope to visualize embryos, coupled with a micromanipulator with integrated joysticks, a pneumatic microinjector with a glass microcapillary to precisely manipulate the microcapillary tip and succinate early-stage embryos. (**b**) Multiwell glass microscopic slide for placing samples and glass microcapillary tip

3. Estimate the embryo morphological stage using a 10× or 20× magnification objective. Adjust the angle of the pneumatic microinjector tip and move it with the micromanipulator joysticks coupled to the inverted microscope (*see* **Note 15**) (*see* Fig. 2a).
4. Collect approximately 20 embryos at once using the glass microcapillary tip coupled to the microinjector, and with the micromanipulator place them in 10 μL of TE buffer in a clean well of the multiwell glass microscopic slide (*see* Fig. 2b). Collect embryos with as little solution and surrounding tissue as possible. To progressively wash the embryos from endosperm and seed coat remnants, repeat the procedure by aspirating the embryos and placing them in another clean well of the multiwell slide with 10 μL of TE buffer, for a second wash. If there are still remnants of undesirable tissue, repeat the procedure for a third wash (*see* **Note 16**).
5. Collect the washed embryos in 5 μL of TE buffer and place them in an Eppendorf LoBind microcentrifuge tube previously frozen in liquid nitrogen (*see* **Note 17**). Flash-freeze in liquid nitrogen and store at −80 °C until further processing.

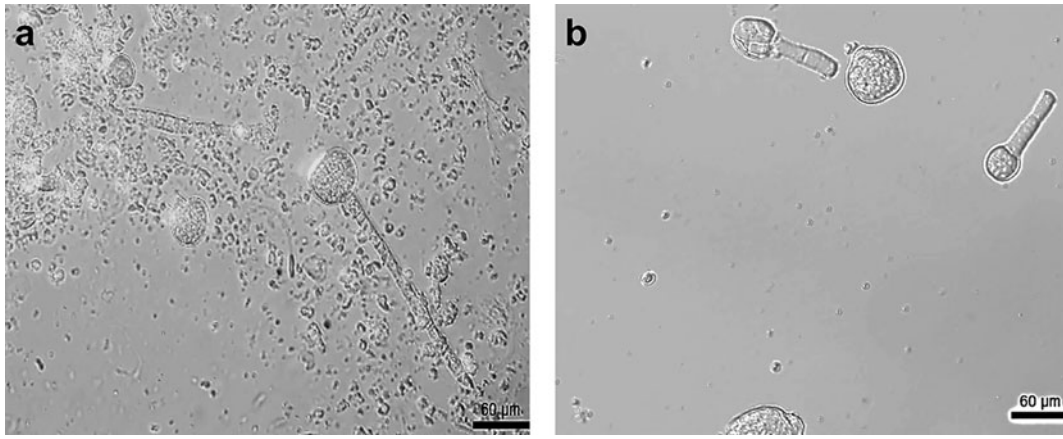


Fig. 2 Isolation of *Boechera stricta* early-stage embryos. (**a**) Seed extract. (**b**) Isolated embryos after first wash with TE buffer. All scale bars = 60 µm. Black arrows indicate globular stage embryos. Micrographs obtained with inverted microscope, 20× magnification objective

6. Isolated embryos can be used to isolate nuclei for in situ Hi-C experiments (see chapter Pérez-de los Santos et al. "Plant in situ Hi-C experimental protocol and bioinformatic analysis" in this volume), with volume adjustments based on the Low-C protocol by Díaz et al. [1] (*see* Fig. 3).

4 Notes

1. Prepare all reagents and solutions with ultrapure DNase and RNase-free distilled water.
2. During all the steps of the protocol use Eppendorf LoBind 1.5 mL microcentrifuge tubes free from DNA, DNase and RNase.
3. Resuspend Triton X-100 in MC buffer by pipetting gently to avoid foaming.
4. Formaldehyde should be added prior to use, under a fume hood. Methanol-free formaldehyde is recommended.
5. Dissolve the glycine powder in ultrapure water at a desired concentration prior to adding to the MC buffer.
6. To reuse micropestles and multiwell glass microscopic slides, wash them every time after use as follows. Wash for 10 min in 10% SDS. Rinse for 2 min in nuclease-free ultrapure water. Immerse for 2 min in 70% ethanol, then immerse for 2 min in 100% ethanol, and air-dry. At each wash step, use sterilized autoclaved Coplin jars.
7. Siliconize all sterilized glassware, including micropestles, microcapillaries, and multiwell glass microscopic slides, to prevent tissue from adhering to the glass surface. Cover the

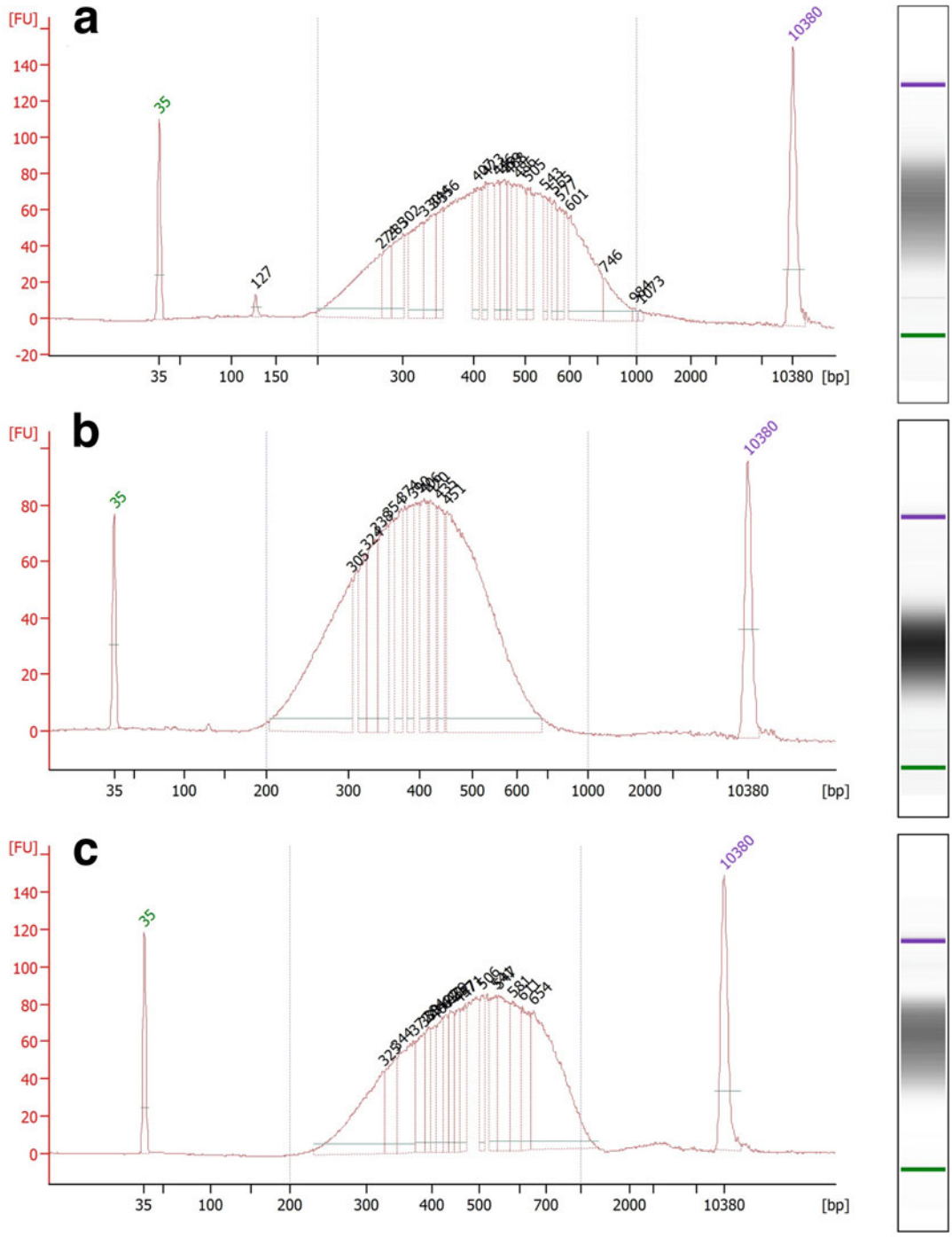


Fig. 3 Bioanalyzer electropherograms of in situ Hi-C libraries generated from isolated *Boechera stricta* embryos. Hi-C libraries generated from (**a**) early-, (**b**) middle-, and (**c**) late-stage isolated embryos

glassware for 15 min with a siliconizing reagent, we recommend undiluted Sigmacote. Remove the remnant solution and air-dry. The remnant solution can be reused. Rinse with ultrapure water.

8. Use a glass microcapillary with an outer diameter according to the microinjector inner diameter.
9. Store siliques in a humid chamber. While other siliques are dissected, place the chamber on ice. Prevent the siliques from getting wet.
10. To easily release the embryo for the next isolation steps, gently make a light cut at the chalazal zone, away from the embryo, cutting the seed coat. Avoid damaging the embryo by cutting far from the micropylar region.
11. For approximately 600 seeds, resuspend in 50 μL of TE buffer. Keep all the sample immerse in the buffer. Always keep microcentrifuge tubes on ice.
12. Do not resuspend by pipetting.
13. Seeds at 4 °C cannot be stored for more than 48 h after their collection. It is always better to process them immediately.
14. To avoid damage, be gentle when applying force to release the embryo. Do not pipette up and down. We recommend continuing with next steps as soon as possible.
15. To achieve the desired diameter of the glass microcapillary tip end, set the heating values, weight, and number of steps following the puller guide settings. We recommend setting values to achieve a tip diameter of 50–100 μm.
16. To prevent samples from drying, do not spend more than 10 min at each step and constantly add 5–10 μL of TE buffer. Usually, two washes are necessary to optimally clean the embryos from surrounding seed remnants before storage.
17. Flash-freeze and store the sample in liquid nitrogen. Add embryos to the same microcentrifuge tube until all embryos are collected. During the collection, keep the sample always frozen in liquid nitrogen. Store samples at −80 °C until further processing.

Acknowledgments

MT-A was funded by a fellowship from the Consejo Nacional de Ciencia y Tecnología (CONACYT). KO is supported by the Consejo Nacional de Ciencia y Tecnología (CONACYT, CB-2016-01/285847). We are grateful to Alfredo Herrera-Estrella, Alfredo Cruz-Ramírez, and Rafael Montiel-Duarte for providing equipment and relevant infrastructure for the establishment of this

protocol, and to Lina López and Joanna Serwatowska for technical help during embryo isolation. Research supported by a grant from the CONACYT Fronteras program assigned to J´Ph.V-C.(#449).

References

1. Díaz N, Kruse K, Erdmann T et al (2018) Chromatin conformation analysis of primary patient tissue using a low input Hi-C method. Nat Commun 9:4938. https://doi.org/10.1038/s41467-018-06961-0
2. Rao SSP, Huntley MH, Durand NC et al (2014) A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. Cell 159:1665–1680. https://doi.org/10.1016/j.cell.2014.11.021
3. Wang N, Liu C (2020) Study of cell-type-specific chromatin organization: in situ Hi-C library preparation for low-input plant materials. Methods Mol Biol 2093:115–127. https://doi.org/10.1007/978-1-0716-0179-2_9
4. Lieberman-Aiden E, van Berkum N, Williams L et al (2009) Comprehensive mapping of long-range interactions reveals folding principles of the human genome. Science 326(5950):289–229. https://doi.org/10.1126/science.1181369
5. Dixon J, Selvaraj S, Yue F et al (2012) Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature 485:376–380. https://doi.org/10.1038/nature11082
6. Dekker J, Marti-Renom M, Mirnz L (2013) Exploring the three dimensional organization of genomes: interpreting chromatin interaction data. Nat Rev Genet 14:390–403. https://doi.org/10.1038/nrg3454
7. Rowley J, Corces V (2018) Organizational principles of 3D genome architecture. Nat Rev Genet 19:789–800. https://doi.org/10.1038/s41576-0180060-8
8. Figueiredo D, Köhler C (2018) Auxin: a molecular trigger of seed development. Genes Dev 32:479–490. https://doi.org/10.1101/gad.312546.118
9. Jurgens G, Mayer U (1994) Arabidopsis. In: Jonathan B (ed) EMBRYOS: color atlas od development. Mosby-Year Book Limited, London
10. Lee CR, Wang B, Mojica JP et al (2017) Young inversion with multiple linked QTLs under selection in a hybrid zone. Nat Ecol Evol 1: 0119. https://doi.org/10.1038/s41559-0170119
11. Belaghzal H, Dekker J, Gibcus JH (2017) Hi-C 2.0: an optimized Hi-C procedure for high-resolution genome-wide mapping of chromosome conformation. Methods 123: 56–65. https://doi.org/10.1016/j.ymeth.2017.04.004
12. Raissig M, Gagliardini V, Jaenisch J et al (2013) Efficient and rapid isolation of early-stage embryos from *Arabidopsis thaliana* seeds. J Vis Exp 76:e50371. https://doi.org/10.3791/50371
13. Liu C (2017) In situ hi-C library preparation for plants to study their three-dimensional chromatin interactions on a genome-wide scale. In: Kaufmann K, Mueller-Roeber B (eds) Plant gene regulatory networks. Methods in molecular biology, vol 1629. Humana Press, New York, NY. https://doi.org/10.1007/978-1-4939-7125-1_11

Part III

Experimental Procedures for Trait Characterization

Chapter 15

Discovering the Secrets of Ancient Plants: Recovery of DNA from Museum and Archaeological Plant Specimens

Oscar Estrada, Stephen M. Richards, and James Breen

Abstract

Plant DNA preserved in ancient specimens has recently gained importance as a tool in comparative genomics, allowing the investigation of evolutionary processes in plant genomes through time. However, recovering the genomic information contained in such specimens is challenging owing to the presence of secondary substances that limit DNA retrieval. In this chapter, we provide a DNA extraction protocol optimized for the recovery of DNA from degraded plant materials. The protocol is based on a commercially available DNA extraction kit that does not require handling of hazardous reagents.

Key words Ancient plant DNA, DNA extraction, Seed DNA extraction, Paleogenomics, Plant genetics, Herbarium

1 Introduction

Analyzing ancient DNA (aDNA) recovered from paleontological, archaeological, and museum specimens has become an increasingly important research tool, unlocking a window into past organisms, environments, and evolutionary processes [1, 2]. Recent technological advances, especially the application of high-throughput sequencing (HTS), have allowed the characterization of the genome sequences of plant specimens dating back to several thousand years [1, 3]. Comparative studies between present-day and ancient plant genomes have provided key insights into the evolution and domestication of species such as maize, barley, wheat, and oaks [3–11]. Future research in ancient plant DNA will significantly enrich our knowledge of plant evolution and adaptation, providing valuable information for agriculture, food security, and conservation.

Researchers have recovered ancient plant DNA from various subfossil and museum specimens, including leaves, fruits, seeds, wood, and even sediments and coprolites [3, 12–15]. In particular,

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_15,

seeds are a common botanical material in archaeological contexts and have been reported as a good source of ancient DNA molecules [4, 9, 10, 16]. However, the extraction of DNA from archaeological samples can often be challenging. First, ancient specimens contain a mixture of endogenous and exogenous contaminant DNA, and these molecules are often degraded and highly fragmented [17–19]. Most of the current plant DNA extraction methods are not optimized to recover fragmented small DNA molecules, and thus modifications to standard methods are needed to isolate DNA from degraded ancient plant specimens. Second, contaminating exogenous DNA also pose a limitation when working with ancient samples. Modern or less degraded DNA is more likely to be recovered and processed in chemical reactions, outcompeting the ancient molecules [17–19]. Third, plant materials contain secondary compounds such as polysaccharides, polyphenols, and proteins [20]. Ancient samples can also have organic and inorganic substances such as humic acids and salts. These secondary compounds can be coextracted with nucleic acids limiting DNA yield and interfering with enzymatic reactions and fluorometric quantifications [20]. Therefore, ancient plant DNA extraction methods should ensure DNA recovery while removing contaminating compounds that may inhibit downstream applications such as PCR amplification and DNA sequencing.

Initially designed for fresh plant tissues, several DNA extraction protocols have been reported to retrieve DNA from ancient plant specimens [21–23]. However, most of these protocols use toxic chemicals like phenol, chloroform, β-mercaptoethanol, and N-phenacylthiazolium bromide (PTB) to remove contaminating compounds. The manipulation of such chemicals makes DNA extraction time-consuming and complicate routine use by researchers. Therefore, commercial kits customized to isolate DNA from plant and soil samples have also been applied to degraded plant samples [24, 25]. Here, we describe a DNA extraction protocol based on the commercially available DNeasy PowerPlant Pro kit (Qiagen, USA), optimized to isolate DNA from complex plant tissues without applying hazardous chemicals. We present a procedure for initial decontamination of specimens and describe the DNA extraction protocol step by step, including modifications such as temperature and time of tissue lysis to maximize the recovery of DNA from ancient plant samples. We have observed that this protocol efficiently isolates high-quality DNA from museum and archaeological seeds suitable for qPCR analysis and DNA sequencing.

2 Materials

All reagents used in this protocol should be sterile, molecular biology grade, and free of DNA and DNases. Equipment and plasticware should be decontaminated before use by UV irradiation, acid treatment (2.5 M HCl), or 5% sodium hypochlorite (bleach) [26–28].

1. Qiagen DNeasy PowerPlant Pro Kit, which includes PowerBead tubes prefilled with 2.38 mm stainless steel beads for tissue disruption, MB Spin Columns (one per sample), bead solution, phenolic separation solution, solution SL, RNAse A, inhibitor removal solution IR, solution PB, ethanol, solution CB, and elution buffer EB (plus 0.05% Tween 20) (*see* **Note 1**).
2. Ultrapure HPLC-grade water.
3. Paraffin film (e.g., Parafilm).
4. Masking tape.
5. Sterile forceps, scalpel, and scalpel blades (one per sample).
6. Sterile Petri dishes (one per sample). If individual petri dishes are not available, several layers of aluminum foil can be used.
7. 1.5 ml and 2 ml safe-lock microcentrifuge tubes (we recommend Eppendorf DNA LoBind tubes or similar for maximum recovery).
8. Aerosol resistant or filtered tips and pipettor set (3–600 μl).
9. Thermomixer or rotator for microcentrifuge tubes (e.g., Eppendorf ThermoMixer or Stuart SB3 rotator).
10. Vortex mixer or bead homogenizer (e.g., Thermo Savant FastPrep 120 instrument or Retsch MM300 Mixer Mill).
11. Microcentrifuge (up to 16,000 × *g*).

3 Methods

Ancient plant specimens are susceptible to contamination from exogenous DNA that can outcompete the extraction, amplification, and analysis of target ancient DNA. Therefore, all sample preparation steps preceding PCR amplification should be performed in a dedicated ancient DNA clean laboratory, physically isolated from any location where large quantities of DNA, such as PCR products, are handled [18, 29, 30]. Ideally, samples should be manipulated in a fume hood. Also, extraction blank controls must be included throughout all procedures.

Sample Preparation

1. Prepare the workspace by cleaning all the surfaces thoroughly with bleach and ethanol.
2. Place an individual seed in a petri dish and remove seed coats using sterile forceps and a scalpel. If seeds do not have coats, clean the external surface with a scalpel (*see* **Note 2**).
3. Transfer the seed into a 2 ml tube, add 1.5 ml 0.5% bleach. Place tubes in rotator and rotate for 5 min at room temperature at 20 rpm.
4. Pulse spin tubes in a centrifuge at 1000 × *g* for 1 min and discard bleach.
5. Wash three times with 1.5 ml ultrapure water.
6. Transfer the seed into a 2 ml PowerBead tube (prefilled with metal beads), add 410 μl of bead solution, 40 μl of phenolic separation solution, 50 μl of solution SL, and 3 μl of RNAse A.
7. Homogenize for 10 min at high vibrational frequency in a tissue homogenizer instrument (6.5 m/sec in FastPrep 120 or 30 Hz in Retsch Mixer Mill) (*see* **Note 3**).
8. Seal tubes with Parafilm to prevent leakage.
9. Incubate the mixture at 42 °C for 24 h with constant agitation (300 rpm in Thermomixer or 20 rpm in rotator).
10. Centrifuge at 13,000 × *g* for 2 min. Recover and transfer supernatant to a 2 ml tube (*see* **Note 4**).
11. Add 175 μl of solution IR, vortex and incubate on ice for 5 min.
12. Centrifuge at 13,000 × *g* for 2 min. Recover and transfer supernatant to a 2 ml tube.
13. Add 680 μl of solution PB and 680 μl of ethanol. Vortex to homogenize.
14. Load 600 μl of the mixture onto the MB Spin column, centrifuge at 8000 × *g* for 30 s, and discard the flow-through. Repeat this step until all the mixture has been filtered with the MB Spin column.
15. Add 500 μl of solution CB to the MB Spin column, centrifuge at 8000 × *g* for 30 s, and discard the flow-through.
16. Add 500 μl of ethanol to the MB Spin column, centrifuge at 8000 × *g* for 30 s, and discard the flow-through.
17. Centrifuge at 13,000 × *g* for 2 min to remove any trace of ethanol. Carefully transfer the MB Spin column to a 1.5 ml LoBind tube.
18. Add 20 μl of elution buffer EB (plus 0.05% Tween 20) directly to the MB Spin filter and incubate for 10 min at 37 °C (*see* **Note 5**).

19. Centrifuge at 10,000 × *g* for 1 min to recover DNA. Extracted DNA is now ready for downstream applications such as qPCR or NGS library construction. DNA extracts must be stored at −20 °C to −80 °C to prevent degradation.

4 Notes

1. Some reagents of the Qiagen DNeasy PowerPlant Pro Kit (e.g., PowerBead solution and solution PB) contain guanidine salts and must not be mixed with bleach (sodium hypochlorite). Also, these chemicals and their containers must be disposed of as hazardous waste. In addition, solution CB and ethanol are flammable. Please review the manufacturer's safety data sheet and consult with institutional authorities to ensure proper handling and disposal.
2. We have optimized this protocol for individual seeds less than 9 mm in length and less than 50 mg weight. However, when bigger seeds or other plant specimens such as leaves or stems are being processed, we recommend using up to 50 mg of tissue with subsequent decontamination with bleach and water.
3. If homogenizer instruments such as FastPrep or Retsch are not available, secure PowerBead tubes on a vortex with masking tape. Vortex at maximum speed for 10 min.
4. Ancient specimens are unique, and sometimes it is necessary to reextract samples with low DNA yields. Therefore, we recommend keeping the pellet that contains all undigested plant material for future reextractions. The pellet can be stored at −20 °C.
5. To maximize DNA recovery, load an additional 20 μl of elution buffer directly to the MB Spin filter, incubate for 10 min at 37 °C, and centrifuge at 10,000 × *g* for 1 min. A total of 40 μl of extracted DNA is recovered.

Acknowledgments

O.E. was supported during his PhD by the Administrative Department of Science, Technology and Innovation of Colombia (COLCIENCIAS), grants CF14-0461/CF15-0672/CF16-0551: Recovery of lost genetic diversity in barley from the Carlsberg Foundation; and grant LP130100648: Identifying the diversity and evolution of loci associated with adaptation to aridity/heat and salinity in ancient cereal crops from the Australian Research Council.

References

1. Orlando L, Allaby R, Skoglund P et al (2021) Ancient DNA analysis. Nat Rev Methods Prim 1:15. https://doi.org/10.1038/s43586-020-00011-0
2. Mitchell KJ, Rawlence NJ (2021) Examining natural history through the lens of Palaeogenomics. Trends Ecol Evol 36:258–267. https://doi.org/10.1016/j.tree.2020.10.005
3. Kistler L, Bieker VC, Martin MD et al (2020) Ancient plant genomics in archaeology, herbaria, and the environment. Annu Rev Plant Biol 71:605–629. https://doi.org/10.1146/annurev-arplant-081519-035837
4. Mascher M, Schuenemann VJ, Davidovich U et al (2016) Genomic analysis of 6,000-year-old cultivated grain illuminates the domestication history of barley. Nat Genet 48(9): 1089–1093. https://doi.org/10.1038/ng.3611
5. Vallebueno-Estrada M, Rodríguez-Arévalo I, Rougon-Cardoso A et al (2016) The earliest maize from San Marcos Tehuacán is a partial domesticate with genomic evidence of inbreeding. Proc Natl Acad Sci U S A 113: 14151–14156. https://doi.org/10.1073/pnas.1609701113
6. Ramos-Madrigal J, Smith BD, Moreno-Mayar JV et al (2016) Genome sequence of a 5,310-year-old maize cob provides insights into the early stages of maize domestication. Curr Biol 26:3195–3201. https://doi.org/10.1016/j.cub.2016.09.036
7. Swarts K, Gutaker RM, Benz B et al (2017) Genomic estimation of complex traits reveals ancient maize adaptation to temperate North America. Science 357:512–515. https://doi.org/10.1126/science.aam9425
8. Scott MF, Botigué LR, Brace S et al (2019) A 3,000-year-old Egyptian emmer wheat genome reveals dispersal and domestication history. Nat Plants 5:1120–1128. https://doi.org/10.1038/s41477-019-0534-5
9. Smith O, Nicholson WV, Kistler L et al (2019) A domestication history of dynamic adaptation and genomic deterioration in sorghum. Nat Plants 5:369–379. https://doi.org/10.1038/s41477-019-0397-9
10. Ramos-Madrigal J, Runge AKW, Bouby L et al (2019) Palaeogenomic insights into the origins of French grapevine diversity. Nat Plants 5: 595–603. https://doi.org/10.1038/s41477-019-0437-5
11. Wagner S, Lagane F, Seguin-Orlando A et al (2018) High-throughput DNA sequencing of ancient wood. Mol Ecol 27:1138–1154. https://doi.org/10.1111/mec.14514
12. Gugerli F, Parducci L, Petit RJ (2005) Ancient plant DNA: review and prospects. New Phytol 166:409–418. https://doi.org/10.1111/j.1469-8137.2005.01360.x
13. Palmer SA, Smith O, Allaby RG (2012) The blossoming of plant archaeogenetics. Ann Anat 194:146–156. https://doi.org/10.1016/j.aanat.2011.03.012
14. Birks HJB, Birks HH (2016) How have studies of ancient DNA from sediments contributed to the reconstruction of quaternary floras? New Phytol 209:499–506. https://doi.org/10.1111/nph.13657
15. Parducci L, Bennett KD, Ficetola GF et al (2017) Ancient plant DNA in lake sediments. New Phytol 214:924–942. https://doi.org/10.1111/nph.14470
16. Winkel T, Aguirre MG, Arizio CM et al (2018) Discontinuities in quinoa biodiversity in the dry Andes: an 18-century perspective based on allelic genotyping. PLoS One 13: e0207519. https://doi.org/10.1371/journal.pone.0207519
17. Hofreiter M, Serre D, Poinar HN et al (2001) Ancient DNA. Nat Rev Genet 2:353–359. https://doi.org/10.1038/35072071
18. Paabo S, Poinar H, Serre D et al (2004) Genetic analyses from ancient DNA. Annu Rev Genet 38:645–679. https://doi.org/10.1146/annurev.genet.37.110801.143214
19. Dabney J, Meyer M, Pa S (2013) Ancient DNA damage. Cold Spring Harb Perspect Biol 5:1–7. https://doi.org/10.1101/cshperspect.a012567
20. Schrader C, Schielke A, Ellerbroek L, Johne R (2012) PCR inhibitors - occurrence, properties and removal. J Appl Microbiol 113:1014–1026. https://doi.org/10.1111/j.1365-2672.2012.05384.x
21. Cappellini E, Gilbert MTP, Geuna F et al (2010) A multidisciplinary study of archaeological grape seeds. Naturwissenschaften 97: 205–217. https://doi.org/10.1007/s00114-009-0629-3
22. Wales N, Andersen K, Cappellini E et al (2014) Optimization of DNA recovery and amplification from non-carbonized archaeobotanical remains. PLoS One 9:e86827. https://doi.org/10.1371/journal.pone.0086827
23. Palmer SA, Moore JD, Clapham AJ et al (2009) Archaeogenetic evidence of ancient nubian barley evolution from six to two-row

indicates local adaptation. PLoS One 4:2–8. https://doi.org/10.1371/journal.pone.0006301

24. Kistler L (2012) Ancient DNA extraction from plants. In: Shapiro B, Hofreiter M (eds) Ancient DNA: methods and protocols. Humana Press, Totowa, NJ, pp 71–79
25. Heenan PB, Wood JR, Cole TL (2018) A partial cpDNA trnL sequence from the extinct legume Streblorrhiza speciosa confirms its placement in the tribe Coluteae (Fabaceae). Phytotaxa 374:87–91. https://doi.org/10.11646/phytotaxa.374.1.8
26. Champlot S, Berthelot C, Pruvost M et al (2010) An efficient multistrategy DNA decontamination procedure of PCR reagents for hypersensitive PCR applications. PLoS One 5: e13042. https://doi.org/10.1371/journal.pone.0013042
27. Tamariz J, Voynarovska K, Prinz M, Caragine T (2006) The application of ultraviolet irradiation to exogenous sources of DNA in plasticware and water for the amplification of low copy number DNA. J Forensic Sci 51:790–794. https://doi.org/10.1111/j.1556-4029.2006.00172.x
28. Shapiro B, Barlow A, Heintzman PD et al (2019) Ancient DNA methods and protocols, 2nd edn. Springer, New York, NY
29. Cooper A, Poinar H (2000) Ancient DNA: do it right or not at all. Science 289:1139. https://doi.org/10.1126/science.289.5482.1139b
30. Llamas B, Valverde G, Fehren-Schmitz L et al (2017) From the field to the laboratory: controlling DNA contamination in human ancient DNA research in the high-throughput sequencing era. STAR Sci Technol Archaeol Res 3:1–14. https://doi.org/10.1080/20548923.2016.1258824

Chapter 16

Use of Allele-Specific Amplification for Rapid Identification of Aromatic and Non-aromatic Rice Germplasms

Debarati Chakraborty

Abstract

In the current context of global climate change trends, threat to food and nutrition security, collection, conservation and management, and characterization and evaluation of crop germplasms especially traditional landraces are gaining momentum more than ever before. Aromatic rice is an elite category of cultivated rice having huge sociocultural heritage value, fetching premium prices globally. Hence, its identification, in situ conservation, and appropriate characterization are likely to augment reliability of this distinctive category of rice, and rice commodity chain actors. *badh2.1* is recognised as the major allele responsible for rice 2-acetyl-1-pyrroline aroma production in a vast number of aromatic rice globally. However, most of the previous works on the genetics and biochemical pathways of aroma expression in rice have encompassed mainly Basmati, Sadri, Della, Jasmine, and a few modern hybrids. But apart from these spotlighted varieties, a myriad of indigenous, aromatic rice germplasms exists. Allele-specific amplification, a low-cost, accurate method invented by Bradbury et al. 2005, can be utilized successfully for discriminating the rarely explored aromatic and nonaromatic rice as described.

Key words Aromatic rice landraces, 2-acetyl-1-pyrroline, Betaine aldehyde dehydrogenase 2, Crop germplasm, Allele specific amplification, Bradbury markers

1 Introduction

Crop germplasm or crop genetic resources include any living genetic resources like seed, tissue, or planting stock, representing the entire genetic variability or diversity available in a crop species. These resources encompass the variability that is raised through millennia of natural and artificial selection. They play a pivotal role in the maintenance of genetic diversity, crop breeding, conservation efforts, food and nutrition security research.

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_16,

1.1 Types of Crop Germplasm

1. *Advanced (or elite) germplasm*: Crop plant varieties have desirable traits, for example, high yield, disease resistance for a particular target environment. But the elite germplasms are often muddled with problems because of their shallow genetic base due to vigorously shared ancestry and restricted usage of exotic crop germplasms as parents.
   (a) *Cultivars/cultivated varieties* are crop plant varieties generated through selective breeding by farmers, breeders harnessing high yield potential, genetic uniformity but narrow genetic base, and low adaptability in comparison to landraces.
   (b) *Advanced breeding material* are new, prereleased better performing crop varieties cross-bred by crop breeders. These plants are not yet ready to be released but harbors valuable allelic combinations. Examples include nearly homozygous lines, transgenics, and mutants.
2. *Landraces* are dynamic population(s) of crop varieties with a historical origin, conspicuous identity, and bereft of formal crop improvement. They have evolved worldwide through centuries in traditional agroecoystems and are often part of traditional cuisines (e.g., use of Gobindobhog rice during Laxmi Puja in West Bengal, India). These crop varieties are locally adapted, have high genetic diversity, and are associated with traditional farming systems [1].
3. *Improved germplasms* are any plant material having single or multiple traits of interest incorporated through artificial selection or planned crossing.
4. *Wild or weedy* relatives are crop plants which share a common ancestry with a domesticated crop species. These genetically diverse plants often act as a potential reserve of various useful traits and are of critical importance for plant breeders and to meet the challenge of global food security through increased productivity.
5. *Genetic stocks* are germplasms focused on one specific or a limited number of defined variations useful to plant breeders for particular needs. They can be of three different types, namely, cytological stocks (e.g., amphiploids, chromosome addition/substitution, aneuploids), mutants (e.g., tilling populations, induced/insertion mutants), and germplasm sets (e.g., parental lines, mapping populations, reference germplasm).

The importance of crop germplasm diversity was truly realized through the Vavilovian concept (1924) of crop centers of origin (or centers of diversity). These areas act as hubs of potential genetically diverse crop resources. Vavilov's forecast on the presence of

earlier unknown crop plant forms and species from different centers of origin has been the guiding beacon for biogeographers for ages. His idea on coevolution of crop diseases and pests at specific centers of origin for specific crops has laid the foundation for crop hunters in search of a treasure trove of crop genetic diversity. One of the most remarkable influences of Vavilovian concept on human civilization is the initiative to establish international gene banks like International Maize and Wheat Improvement Center (CIMMYT) in Mexico (1943), International Rice Research Institute (IRRI) in Philippines (1960), for conservation of worldwide diversity of crop germplasm resources.

Of the different germplasm resources, landraces have got the spotlight for their existence since the origin of farming traditions [2]. They still are cultivated extensively in marginal environments and for niche markets. Landraces are well adapted to the regional farming systems, often are an integral part of ethnic cultures and agronomic practices, and occupy a crucial part of the local diet [3]. These crop plants are crucial in crop breeding programs by virtue of their important traits, for example, biotic and abiotic stress tolerance, better palatability, stable yield, and nutritional quality [4, 5]. Yet the current scenario of industrial agricultural since the advent green revolution is posing the most significant threat to agrobiodiversity. The trend of monocropping with genetically uniform hybrids and improved germplasms is causing unceasing replacement of locally adapted landraces, subsequently leading to severe genetic erosion [6, 7]. Thus, our food resources are gradually becoming more and more vulnerable. The situation is even more perilous as the need to adapt crops to changed climate conditions is soaring [8]. Hence, collection, conservation and management, and characterisation and evaluation of crop germplasms especially traditional landraces are gaining momentum more than ever before.

Rice (*Oryza sativa*) is one of the most ancient crops which have nurtured several civilizations. Asian cultivated rice is subdivided into five subpopulations, for example, *indica*, *aus*, *tropical japonica*, *temperate japonica*, and *Group V* [9, 10]. Aromatic rice (also called fragrant, perfumed or scented rice) is an elite category of cultivated rice. It owns about 15–18% share of the global rice business market fetching the premium prices globally [11]. Apart from economic value, aromatic rice also beholds a huge sociocultural heritage value [12, 13]. Hence, its identification, in situ conservation, and appropriate characterization are likely to augment reliability of this distinctive category of rice, and rice commodity chain actors also [14]. Furthermore, aromatic rice is not confined but allotted to different subpopulations of rice, as depicted in the Table 1**,** thus making the efforts to untangle the history of origin and evolution of aroma in rice a difficult task.

Table 1
Some examples of aromatic rice belonging to different subpopulations [5, 15–18]

Subpopulation	Name of aromatic rice
Indica	Gobindobhog, Badshabhog, Khao Dawk Mali (KDML) 105
Aus	Balam joha, Joha bora, Pokikoli joha, Solong joha, Baghe tulashi
Tropical japonica	Pandan Wangi 7, Jakou Mochi, Kaori Wase, Kamari, Nioi Mochi, Jakou-1, Jakou- 2, Jakou-3
Temperate japonica	Kyeema, Wase Hieri, Shiro Wase, Kabashiko-2, Nioi Kichi, Kaisen, Mangoku
Group V	Punjab basmati (Bauni basmati), Chenab basmati, Taraori, Sadri

Probing into the genetic basis of rice 2-acetyl-1-pyrroline (2-AP) aroma revealed a single recessive allele (*fgr*) associated with it [19, 20]. An 8-bp deletion in seventh exon of rice *badh2* gene and three single-nucleotide polymorphisms (SNPs) denoted as *badh2.1* allele is recognized as the cause of rice aroma production [21]. This recessive allele is reported in several aromatic rice varieties globally [21–23]. But apart from *badh2.1* several other mutated alleles of *badh2* responsible for aroma expression are pinpointed [22–29]. Most of the previous works on the genetics and biochemical pathways of aroma expression in rice have encompassed mainly Basmati, Sadri, Della, Jasmine, and a few modern hybrids [30–32]. But apart from these spotlighted varieties, a myriad of indigenous, aromatic rice germplasms live and breathe to be unveiled for solving the queries about enormous change which happened in morphology, physiology, genetics, and adaptivity traits of aromatic rice, necessary for proper conservation of these germplasms via successful breeding efforts [5, 22, 33]. Hence, a part of my doctoral work explored 84 *indica* rice landraces collected from indigenous farmers of different states of India and conserved on-farm at Vrihi Seed Bank of Basudha (www.cintdis.org/basudha, www.cintdis-.org/vrihi), located in the Rayagada district of Odisha, India [15, 33, 34]. The study consists of 55 aromatic and 27 nonaromatic rice landraces. The study aimed at finding the causative mutation underlying aroma in these landraces and its implication in broader context of evolution of aroma in *indica* rice subpopulation. Majority of the aromatic rice samples (44) have the 8-bp deletion. The rest of the samples, 11 out of 55 aromatic landraces, did not have *badh2.1* allele. The finding indicates the presence of already known (*badh2.2–2.10*) or unknown mutation(s) in a different part of the *badh2* gene or in the promoter region [22, 24] in these 11 samples.

In this chapter, I have focused particularly on the description of the protocol for identifying aromatic and nonaromatic rice landraces using allele-specific amplification [21, 25, 34]. Allele-specific amplification (ASA) is a cheap, accurate, time-saving method developed by [21] to distinguish alleles which differ by insertions or deletions and/or SNPs in a single PCR tube.

2 Materials

1. *Labware*: Mortar and pestle, gloves, tissue, spatula, scissor, beaker, magnetic stirrer, volumetric flasks, magnets, measuring cylinder, glass bottles, amber bottle, microfuge tubes, micropipettes, micropipette holder, PCR tubes, microtips, float racks, quartz cuvettes, PCR tube racks, agarose gel casting tray, gel casting combs.
2. *Chemicals*: Liquid nitrogen, absolute ethanol, chloroform, isoamyl alcohol, sterile water, boric acid, polyvinyl pyrrolidone (PVP), β-mercaptoethanol, disodium salt of ethylenediaminetetraacetic acid (Na_2EDTA), EDTA, Tris base, boric acid, Tris–HCl, sodium hydroxide, hydrochloric acid, sodium chloride, ribonuclease (RNAse), agarose, ethidium bromide (ETBr), DNA Ladders.
3. *Buffers*: Cetyl trimethyl ammonium bromide (CTAB), Tris–EDTA (TE), Tris–borate–EDTA (TBE) electrophoresis buffer, DNA sample loading buffer—glycerol, and tracking dyes (bromophenol blue and xylene cyanol FF).
4. *Instruments*: Fine balance, pH meter, vortex mixer, cold centrifuge, autoclave, water bath, refrigerator, −20 °C freezer, minicooler, UV–visible spectrophotometer, PCR thermocycler, microwave, gel electrophoresis system, transilluminator.
5. *Reagent Preparation*.
   (a) *Chloroform–isoamyl alcohol (24:1)*: Mix 96 ml of chloroform in 4 ml of isoamyl alcohol and store in amber bottle.
   (b) *Sodium chloride (5 Molar)*: Mix 292.2 g of sodium chloride in sterile water. Volume made up to 1000 ml in a volumetric flask with sterile water.
   (c) *Tris–HCl (1 Molar)*: Dissolve 121.8 g Tris–HCl in 700 ml sterile water. Adjust pH to 8.0 with hydrochloric acid or sodium hydroxide as required. Volume made up to 1000 ml in a volumetric flask with sterile water.
   (d) *Tris base (1 Molar)*: Dissolve 12.84 g of Tris base in 80 ml sterile water. Adjust pH to 8 using hydrochloric acid or sodium hydroxide as required. Volume made up to 100 ml in a volumetric flask with sterile water.

(e) *70% ice-cold ethanol*: Dissolve 70 ml absolute ethanol in 30 ml sterile water and store at 4 °C.

(f) *EDTA (0.5 Molar)*: Mix 18.6 g of disodium salt of EDTA in 80 ml sterile water. First adjust the pH of water to 7.5 and then add EDTA for ease. Only after complete dissolution of EDTA pH to 8 using sodium hydroxide. Volume made up to 100 ml in a volumetric flask with sterile water.

(g) *Ethidium Bromide (10 mg/ml)*: Add 1 g of ethidium bromide to 100 ml of H_2O. Stir on a magnetic stirrer for several hours for completely dissolving the dye. Store in an amber bottle at room temperature.

6. *Buffer preparation*.

(a) *CTAB*: Mix 100 ml of 1 M Tris–HCl (pH 8.0) is mixed with 280 ml of 5 M NaCl, 40 ml of 0.5 M Na2EDTA, 20 g of CTAB in 400 ml of distilled water. Ensure complete dissolution by placing the beaker on magnetic stirrer. Volume made up to 1 l in a volumetric flask with sterile water.

(b) *TE*: Mix 10 ml of 1 M Tris base (pH 8.0) and 2 ml of 0.5 M EDTA pH 8.0 in 900 ml water. Adjust pH to 8 using hydrochloric acid or sodium hydroxide. Volume made up to 1 l in a volumetric flask with sterile water. Autoclave and filter-sterilize before use.

(c) *Tris–borate–EDTA (TBE) 10×*: Dissolve 108 g of Tris base, 55 g of boric acid, and 40 ml 0.5 M Na2EDTA (pH 8.0) in 600 ml of sterile water. Stir on a magnetic stirrer. Volume made up to 1 l in a volumetric flask with sterile water. Store at room temperature. Prepare a 1× working solution from 10× stock by diluting with sterile water (*see* **Note 1**).

3 Methods

3.1 Genomic DNA Isolation Procedure [35, 36]

1. Weigh about 100 mg of leaf tissues using fine balance.
2. Take the samples in mortar and cut with scissor to small fragments.
3. Ground to leaves to a fine powder using liquid nitrogen and pestle.
4. Add 20 milligram of polyvinyl pyrrolidone (PVP), 2 ml of 2% CTAB buffer and make a paste. It helps in removal of phenolics by forming hydrogen bonds with these secondary metabolites.
5. Add 5 microliter (μl) of β-mercaptoethanol. Being a strong reducing agent, it helps to get rid of tannins and other

polyphenols occurring in crude plant extracts. Take precautions not to smell or touch this chemical as it is highly toxic (*see* **Note 2**).

6. Equally divide the sample into two 2 ml microcentrifuge tubes. Mix it on a vortex mixer.
7. Incubate the tubes at 65 °C for 2 h in a water bath or alternatively on a heating block.
8. Mix the samples intermittently at regular intervals by inverting (once in 30 mins) while incubation is ongoing.
9. Remove the tubes from the water bath and soak excess water with tissue.
10. Centrifuge at 16,600 × *g* for 10 min at room temperature and transfer the supernatant was to another 2 ml micro centrifuge tube.
11. Add 1000 µl of chloroform–isoamyl alcohol and mix gently for 5 min. It will remove the organic compounds.
12. Centrifuge again at 16,600 × *g* for 10 min at room temperature.
13. Collect the separated aqueous layer in another 2 ml tube.
14. Add 400 µl of 5 M sodium chloride in the aqueous solution slowly for removal of polysaccharides and follow by adding 800 µl of 70% ice-cold ethanol.
15. Mix gently by inverting several times to precipitate the DNA. Incubate tubes on a float rack at −20 °C for at least 2 h or keep overnight and continue the protocol next day.
16. Centrifuge the incubated tubes at 16,600 × *g* for 10 min. Carefully discard supernatant with the micropipette leaving the pellets only.
17. Wash the pellet with 70% cold ethanol (500 µl/tube).
18. Centrifuge again at 16,600 × *g* for 10 min and similarly discard the supernatant as before. Repeat the step another time for better precipitation of DNA.
19. Dry the pellet under laminar airflow chamber.
20. Dissolve the DNA pellet in 1X TE buffer (100 µl).
21. Add 4 µl RNase (10 mg/ml) and incubate at 37 °C for 2 h in an incubator.
22. Store the purified genomic DNA at 4 °C for temporary use and at −20 °C for long term storage.

3.2 Measurement of DNA Concentration in Spectrophotometer

To ensure the extracted genomic DNA has necessary concentration for downstream processing, quantify the extracted sample using UV–visible spectrophotometer. Pure DNA shows maximum absorption at 260 nm due to absorption of UV rays by the

heterocyclic nitrogenous bases of DNA. Genomic DNA samples are also measured at 280 nm which is the absorption peak for protein. Pure preparations of DNA show ratio of readings at 260–280 nm above 1.8. Presence of protein or phenol contamination will significantly lower the ratio and affect the accurate quantification of extracted genomic DNA.

3.2.1 Procedure

1. Make blank using 1X TE buffer.
2. Dilute 1 μl of the extracted genomic DNA sample to 100 μl. The dilution factor is 100.
3. Take readings take 260 nm and 280 nm.

$$\text{Absorbance}_{260\ \text{nm}} = 1 \text{ quantifies to } 50\ \text{ng}/\mu\text{l of dsDNA.}$$

$$\text{DNA concentration} = \text{Absorbance}_{260\ \text{nm}} \times 50\ \text{ng}/\mu\text{l} \times \text{dilution factor.}$$

3.3 Allele Specific Amplification in PCR

3.3.1 Procedure [21, 25]

Two different sets of primers described by [21, 25] are used in the study described in Table 2 (*see* **Note 3**). PCR reaction mixture volume is made to 23.5 μl and comprised of 14.5 μl of nuclease free water, 3 μl of 10× PCR buffer, 10 mM dNTP 2.5 μl, 0.5 U of Taq DNA polymerase, 1 μl of bovine serum albumin, 1 μl of sample DNA, and 1 μl of Bradbury markers or Shi markers. The PCR reactions conditions are as follows: preliminary denaturation at 95 °C for 3 min, continued for 35 cycles of 1 min at 95 °C, 1 min at 58 °C and 2 min at 72 °C, with 10 min of extension at 72 °C [34].

Table 2
Primer details used for detection of badh2.1 allele [21, 25, 34]

Marker name	Primer sequences (5′–3′)
A. Bradbury markers	
1. External sense primer (ESP)	TTGTTTGGAGCTTGCTGATG
2. Internal fragrant antisense primer (IFAP)	CATAGGAGCAGCTGAAATATATACC
3. Internal nonfragrant sense primer (INSP)	CTGGTAAAAAGATTATGGCTTCA
4. External antisense primer (EAP)	AGTGCTTTACAAAGTCCCGC
B. Shi markers	
1. FmBADH2E7 forward primer	GGTTGCATTTACTGGGAGTT
2. FmBADH2E7 reverse primer	CAGTGAAACAGGCTGTCAAG

3.4 Visualization of Amplified Fragments in Agarose Gel Electrophoretic System

3.4.1 Procedure

1. Gel casting tray preparation.
   (a) Seal the ends of the casting tray with tape to ensure non-leakage of gel on pouring.
   (b) Place the combs in the tray.
2. Agarose gel preparation.
   (a) Add agarose to 100 ml of TBE buffer.
   (b) Heat the container in microwave for 4 min until the agarose is completely dissolved and looks transparent.
   (c) Allow the agarose to cool for a minute.
   (d) Add 3 μl of ETBr/100 ml of agarose gel.
   (e) Mix the gel solution thoroughly by gentle swirling. Skim off any polymerized "skin" before pouring.
   (f) Pour the melted agarose into the gel tray and ensure absence of air bubbles.
   (g) Let the gel solidify. The gel should be between 3 and 5 mm thick (*see* **Note 4**).
   (h) Gently pull out the combs and remove the tape.
   (i) Place the gel in the electrophoresis chamber.
   (j) Add enough electrophoresis buffer such that about 2–3 mm of buffer remains above the gel.
3. DNA sample loading, gel running, and visualization.
   (a) Add 3–5 μl of 6× sample Loading Buffer to each genomic DNA sample. Record the order in which samples are loaded in each well.
   (b) Load 10 μl of DNA ladder into at least one well of each row of the gel.
   (c) Run the gel at 90–105 V, if needed for clearer bands low voltage (60 V) can be used. DNA will migrate toward the positive anode. Check intermittently that the DNA samples are migrating in the correct direction by observing the direction of loading dye after a couple of minutes.
   (d) Let the gel run until the dye reaches the end of the gel.
   (e) Use gloves and gel scoop for removing the gel from the agarose gel casting tray.
   (f) Place directly on a transilluminator and switch on UV lamp. DNA bands will be visualized in UV light (*see* **Note 5**).

4 Notes

1. Even with quick stirring, 10× TBE will take time to dissolve. Dissolve clumps by placing the beaker in water bath (37 °C to 42 °C). Upon storage precipitation of TBE may occur, which can be dissolved by heating.
2. Do not inhale β-mercaptoethanol.
3. Always wear gloves to avoid contamination from skin.
4. Solidified agarose can be stored at room temperature and remelted in a microwave oven by heating it for 3–5 min. Remelting will cause an increment in agarose gel concentration which should be compensated by adding sterile water. Take precautions while transferring the gel to the transilluminator as it might break.
5. Presence and absence of *badh2.1* alleles in the samples are determined by the presence of bands and further sequence alignment. In homozygous nonaromatic samples, two bands, 580 bp (control) and 355 bp, are observed, whereas heterozygous nonaromatic samples with *badh2.1* allele showed three bands: 580, 355, and 257 bp. In aromatic samples two bands, 580 bp (control) and 257 bp, are obtained in accordance with [21]. For details refer to [34].

References

1. Villa TC, Maxted N, Scholten M, Ford-Lloyd B (2005) Defining and identifying crop landraces. Plant Genet Resour 3(3):373–384
2. Zeven AC (1998) Landraces: a review of definitions and classifications. Euphytica 104: 127–139
3. Zeven AC (2002) Traditional maintenance breeding of landraces: 2. Practical and theoretical considerations on maintenance of variation of landraces by farmers and gardeners. Euphytica 123(2):147–158
4. Frankel OH, Brown AHD, Burdon JJ (1995) The conservation of plant biodiversity. Cambridge University Press, Cambridge
5. Chakraborty Thesis (2021). https://www.researchgate.net/publication/354059054_An_Investigation_Of_Genetic_And_Biochemical_Characters_Of_Aroma_In_Traditional_Indian_Rice_Landraces_And_Its_Larger_Ecological_Role_In_Plants
6. Frison EA et al (2011) Agricultural biodiversity is essential for a sustainable improvement in food and nutrition security. Sustainability 3: 238–253
7. Dwivedi SL, Ceccarelli S, Blair MW, Upadhyaya HD, Are AK, Ortiz R (2016) Landrace germplasm for improving yield and abiotic stress adaptation. Trends Plant Sci 21(1): 31–42
8. Ceccarelli S (2012) Landraces: importance and use in breeding and environmentally friendly agronomic systems. In: Maxted N et al (eds) Agrobiodiversity conservation: securing the diversity of crop wild relatives and landraces. CAB International, Wallingford, pp 103–117
9. Garris AJ, Tai TH, Coburn J, Kresovich S, McCouch S (2005) Genetic structure and diversity in *Oryza sativa* L. Genetics 169(3): 1631–1638
10. Zhao K, Tung CW, Eizenga GC, Wright MH, Ali ML, Price AH, Norton GJ, Islam MR, Reynolds A, Mezey J, McClung AM, Bustamante CD, McCouch SR (2011) Genome-wide association mapping reveals a rich genetic architecture of complex traits in *Oryza sativa*. Nat Commun 2:467
11. Giraud G (2013) The world market of fragrant rice, main issues and perspectives. Int Food Agribusiness Manage 16(1030-2016-82817):1–20
12. Bhattacharjee P, Singhal RS, Kulkarni PR (2002) Basmati rice: a review. Int J Food Sci Technol 37(1):1–12

13. Ahuja SC, Ahuja U (2010) Rice in social and cultural life of people. In: Sharma SD (ed) Rice, origin, antiquity and history. CRC Publication, USA, pp 39–84
14. Marie-Vivien D (2008) From plant variety definition to geographical indication protection: a search for the link between basmati Rice and India/Pakistan. J World Intellect Property 11(4):312–344
15. Ray A, Deb D, Ray R, Chattopadhayay B (2013) Phenotypic characters of rice landraces reveal independent lineages of short-grain aromatic indica rice. AoB Plants 5:plt032
16. Roy S, Banerjee A, Mawkhlieng B, Misra AK, Pattanayak A, Harish GD, Singh SK, Ngachan SV, Bansal KC (2015) Genetic diversity and population structure in aromatic and quality rice (*Oryza sativa* L.) landraces from North-Eastern India. PLoS One 10(6):e0129607
17. Roy PS, Rao GJ, Jena S, Samal R, Patnaik A, Patnaik SS, Jambhulkar NN, Sharma S, Mohapatra T (2016) Nuclear and chloroplast DNA variation provides insights into population structure and multiple origin of native aromatic rices of Odisha, India. PLoS One 11(9): e0162268
18. Okoshi M, Matsuno K, Okuno K, Ogawa M, Itani T, Fujimura T (2016) Genetic diversity in Japanese aromatic rice (*Oryza sativa* L.) as revealed by nuclear and organelle DNA markers. Genet Resour Crop Evol 63(2):199–208
19. Lorieux M, Petrov M, Huang N, Guiderdoni E, Ghesquiere A (1996) Aroma in rice: genetic analysis of a quantitative trait. Theor Appl Genet 93(7):1145–1151
20. Garland S, Lewin L, Blakeney A, Reinke R, Henry R (2000) PCR-based molecular markers for the fragrance gene in rice (Oryza sativa. L.). Theor Appl Genet 101(3):364–371
21. Bradbury LMT, Henry RJ, Jin Q, Waters DL (2005) A perfect marker for fragrance genotyping in rice. Mol Breed 16:279–283
22. Kovach MJ, Calingacion MN, Fitzgerald MA, McCouch SR (2009) The origin and evolution of fragrance in rice (*Oryza sativa* L.). Proc Natl Acad Sci U S A 106(34):14444–14449
23. Myint KM, Arikit S, Wanchana S, Yoshihashi T, Choowongkomon K, Vanavichit A (2012) A PCR-based marker for a locus conferring the aroma in Myanmar rice (Oryza sativa L.). Theor Appl Genet 125(5):887–896
24. Bourgis F, Guyot R, Gherbi H, Tailliez E, Amabile I, Salse J, Lorieux M, Delseny M, Ghesquiere A (2008) Characterization of the major fragrance gene from an aromatic japonica rice and analysis of its diversity in Asian cultivated rice. Theor Appl Genet 117(3): 353–368
25. Shi W, Yang Y, Chen S, Xu M (2008) Discovery of a new fragrance allele and the development of functional markers for the breeding of fragrant rice varieties. Mol Breed 22:185–192
26. Shao G, Tang S, Chen M, Wei X, He J, Luo J, Jiao G, Hu Y, Xie L, Hu P (2013) Haplotype variation at Badh2, the gene determining fragrance in rice. Genomics 01(2):157–162
27. Shi Y, Zhao G, Xu X, Li J (2014) Discovery of a new fragrance allele and development of functional markers for identifying diverse fragrant genotypes in rice. Mol Breed 33(3):701–708
28. He Q, Park YJ (2015) Discovery of a novel fragrant allele and development of functional markers for fragrance in rice. Mol Breed 35(11):217–227
29. Withana W, Kularathna R, Kottearachchi N, Kekulandara D, Weerasena J, Steele K (2020) In silico analysis of the fragrance gene (badh2) in Asian rice (Oryza sativa L.) germplasm and validation of allele specific markers. Plant Genet Resour 18(2):71–80
30. Ahn SN, Bollich CN, Tanksley SD (1992) RFLP tagging of a gene for aroma in rice. Theor Appl Genet 84(7–8):825–828
31. Nagaraju J, Kathirvel M, Kumar RR, Siddiq EA, Hasnain SE (2002) Genetic analysis of traditional and evolved basmati and non-basmati rice varieties by using fluorescence-based ISSR PCR and SSR markers. Proc Natl Acad Sci U S A 99(9): 5836–5841
32. Sakthivel K, Rani NS, Pandey MK, Sivaranjani AKP, Neeraja CN, Balachandran SM, Madhav MS, Viraktamath BC, Prasad GSV, Sundaram RM (2009) Development of a simple functional marker for fragrance in rice and its validation in Indian basmati and non-basmati fragrant rice varieties. Mol Breed 24(2): 185–190
33. Deb D (2005) Seeds of tradition, seeds of future, folk rice varieties of Eastern India. Research Foundation for Science, Technology and Ecology, New Delhi
34. Chakraborty D, Deb D, Ray A (2016) An analysis of variation of the aroma gene in rice (*Oryza sativa* L. subsp. *indica Kato*) landraces. Genet Resour Crop Evol 63(6):953–959
35. Doyle JJ, Doyle JL (1987) A rapid DNA isolation procedure for small quantities of fresh leaf tissue. Phytochem Bull 19:11–15
36. Lodhi MA, Ye GN, Weeden NF, Reisch BI (1994) A simple and efficient method for DNA extraction from grapevine cultivars and Vitis species. Plant Mol Biol Report 12(1): 6–13

Chapter 17

Efficient Protein Extraction Protocols for NanoLC-MS/MS Proteomics Analysis of Plant Tissues with High Proteolytic Activity: A Case Study with Pineapple Pulp

Esaú Bojórquez-Velázquez, José M. Elizalde-Contreras, Jesús Alejandro Zamora-Briseño, and Eliel Ruiz-May

Abstract

Proteomics is an essential tool to uncover the regulatory processes of fruit biology. In fruits with high proteolytic activity, the inhibition of endogenous proteases is key for successful protein extraction. In this chapter, we describe an efficient protocol for total protein extraction to deal with this inconvenience using pineapple pulp as an example. We corroborated the efficacy of our protein extraction protocols by carrying out nano LC-MS/MS analyses using a highly sensitive hybrid mass spectrometer. In doing so, we were able to identify over 3000 proteins in pineapple pulp. Our contribution paves the way for massive comparative proteomics scrutiny in pineapple fruits, as well as others plant tissues with high protease activity such as papaya, fig, and kiwi fruits.

Key words Proteases, Phenol, Proteomics, Mass spectrometry

1 Introduction

Proteomic studies in fruits are paramount for understanding the molecular foundation of fruit ontogeny from zygotic embryogenesis to fruit development and ripening to fruit–pathogen interactions [1]. Fruit tissues are usually intricate matrices with high contents of sugars, lipids, metabolites, and complex polymers. In addition to this, certain tissues, like peels, or subcellular compartments, such as apoplasts or vacuoles, frequently have a high amount of proteolytic activity (*see* **Note 1**). Proteolysis plays a fundamental role in processes during plant development including growth, seed germination, development, senescence, immunity, programmed cell death, hypersensitive response, and stress responses [2–4]. Proteases are directly associated with the correct functionality and homeostasis of the proteome (e.g., protein halftime and turnover)

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6_17,

and are regulated by several mechanisms including the production of immature proteins, compartmentalization, or specific protease inhibitors [5]. However, this endogenous proteolytic activity can interfere with biochemical studies in fruits by limiting the isolation of high-quality, nonhydrolyzed proteins, which are essential for proteomics analysis [6–8].

Two representative examples of tissues with high intrinsic proteolytic activity are pineapple and papaya fruits. In these fruits, cysteine proteases of the papain family are present at high concentrations during development and ripening (*see* **Note 2**), making it very difficult to characterize the proteomes of these tissues [9]. Currently, cocktails of protease inhibitors have been the primary approach to overcoming proteolytic activities and denature protein extraction protocols for its further analysis by electrophoresis, chromatography, and mass spectrometry. However, using protease inhibitor cocktails in extraction buffers does not guarantee the complete inhibition of proteases. In fact, we could expect partial proteolysis of plant enzymes by endogenous proteases. This condition can provide erroneous data during biochemical and proteomics studies [6, 7]. Therefore, protocols that provide high-efficiency inhibition of proteolytic activity and allow the extraction of high-quality proteins for proteomic studies are desirable.

This chapter compares five extraction protocols with different variants, including commercial protease cocktail, use of detergents, chaotropic compounds, and phenolic solution. We determined that using sodium dodecyl sulphate plus heating or the use of direct extraction with saturated phenol solution provide the most efficient inhibition of pineapple proteases. Based on our pipeline, we were able to identify over 3000 proteins in pineapple pulp independently with any of the three methods that were successful.

2 Materials

Prepare all solutions using ultrapure water (prepared by purifying deionized water) and analytical grade reagents.

2.1 Total Protein Extraction

1. Mortar and pestle.
2. Liquid nitrogen.
3. 2.0 mL microtubes
4. 15 mL conical tubes
5. Extraction reagents and buffers.
   - 7 M urea
   - 2 M thiourea
   - 1% (w/v) SDS

- 10% (w/v) SDS
- 0.1 M Trizma base, pH 8.5
- 30% (w/v) sucrose
- 0.15 M NaCl
- 1% (w/v) SDS
- Phenol solution.
- 10 mM Tris–HCl, pH 8.0
- 1 mM EDTA (Sigma-P4557)
- 2-mercaptoethanol
- Protease inhibitor cocktail (Sigma-P9599).

6. Protein quantification reagents (e.g., Thermo Scientific Pierce BCA Protein Assay kit- 23225).

2.2 SDS-Polyacrylamide Gel Electrophoresis

1. Resolving gel buffer: 1.5 M Tris–HCl, pH 8.8. Add about 100 mL water to a 1 L glass beaker. Weigh 181.7 g Tris–HCl and transfer to the beaker. Add water to a volume of 900 mL. Mix and adjust pH with HCl. Fill to 1 L with water. Store at 4 °C.
2. Stacking gel buffer: 0.5 M Tris–HCl, pH 6.8. Weigh 60.6 g Tris–HCl and prepare a 1 L solution as in previous step. Store at 4 °C.
3. 30% acrylamide–bisacrylamide solution (29.2:0.8 acrylamide–bisacrylamide): Weigh 29.2 g of acrylamide monomer and 0.8 g bisacrylamide (cross-linker) and transfer to a 100 mL graduated cylinder containing 40 mL of water. Store at 4 °C in a bottle wrapped with aluminium foil.
4. Fresh 10% ammonium persulfate (APS) aqueous solution.
5. TEMED (*N, N, N, N*′-tetramethyl-ethylenediamine): store at 4 °C.
6. Electrophoresis running buffer (10×): 30.0 g Tris, 144.0 g glycine, and 10.0 g SDS, dissolve in ultrapure water to a final volume of 1 L.
7. SDS lysis buffer (5×): 0.3 M Tris–HCl (pH 6.8), 10% SDS, 25% β-mercaptoethanol, 0.1% bromophenol blue, 45% glycerol. Leave a 1 mL aliquot at 4 °C for immediate use and store the remaining aliquots at −20 °C.
8. SDS protein sample buffer [10% glycerol (v/v), 1% SDS (w/v), 0.05 M Trizma base, pH 6.8, 5% (v/v) 2-mercaptoethanol plus a trace of bromophenol blue.
9. Casting for two 1.5 mm minigels (Resolving 12% acrylamide-gels: 5.26 mL water, 4.0 mL resolving buffer, 6.4 mL acrylamide–bis-acrylamide, 0.16 mL 10% SDS, 0.16 mL 10% APS, 0.016 mL TEMED. Stacking 4% acrylamide gels: 3.02 mL

water, 1.25 mL stacking buffer, 0.65 mL acrylamide–bisacrylamide, 0.05 mL 10% SDS, 0.05 mL 10% APS, 0.005 mL TEMED).

10. Prestained protein molecular weight markers (e.g., Thermo Scientific).
11. Gel casting and electrophoresis system (e.g., Mini-PROTEAN® Tetra Handcast System with Mini-PROTEAN® Tetra Cell, Bio-Rad).

2.3 Reduction, Alkylation, and Digestion

- 100 mM tris(2-carboxyethyl)phosphine hydrochloride (TCEP)
- 300 mM iodoacetamide (IAM)
- 300 mM dithiothreitol (DTT)
- Acetone, HPLC grade.

2.4 High Reverse-Phase (RP) Fractionation

- Reverse phase fractionation resin, triethylamine (0.1%) (e.g., Pierce™ High pH Reversed-phase peptide Fractionation Kit, Thermo Scientific-84868).

2.5 Other Materials and Equipment

- Gel documentation system (e.g., Gel Doc™ XR System, Bio-Rad®, and software Image Lab™).
- Centrifugal vacuum concentrator (e.g., CentriVap, LABCONCO®).
- Liquid chromatography–mass spectrometry (LC-MS) system.

3 Methods

A general flow chart with the methods used in this protocol is shown in Fig. 1

3.1 Protein Extraction from Tissue

1. Grind pineapple fruit pulp tissue to a fine powder in liquid nitrogen with a mortar and pestle. Keep in a 50 mL tube in liquid nitrogen while processing or store at −80 °C until analysis.
2. Mix the sample powder in a 1:10 (w/v) ratio in five different 2 mL microtubes with the following solutions, corresponding to each tested method.

   Tube A: 0.15 g + 1.5 mL of a solution of 7 M urea, 2 M thiourea, 1% (w/v) SDS.

   Tube B: 0.15 g + 1.5 mL of a solution of 10% (w/v) SDS.

   Tube C: 0.15 g + 1.5 mL of Standard extraction solution [0.1 M Trizma base, pH 8.5, 30% (w/v) sucrose, 0.15 M NaCl, 1% (w/v) SDS].

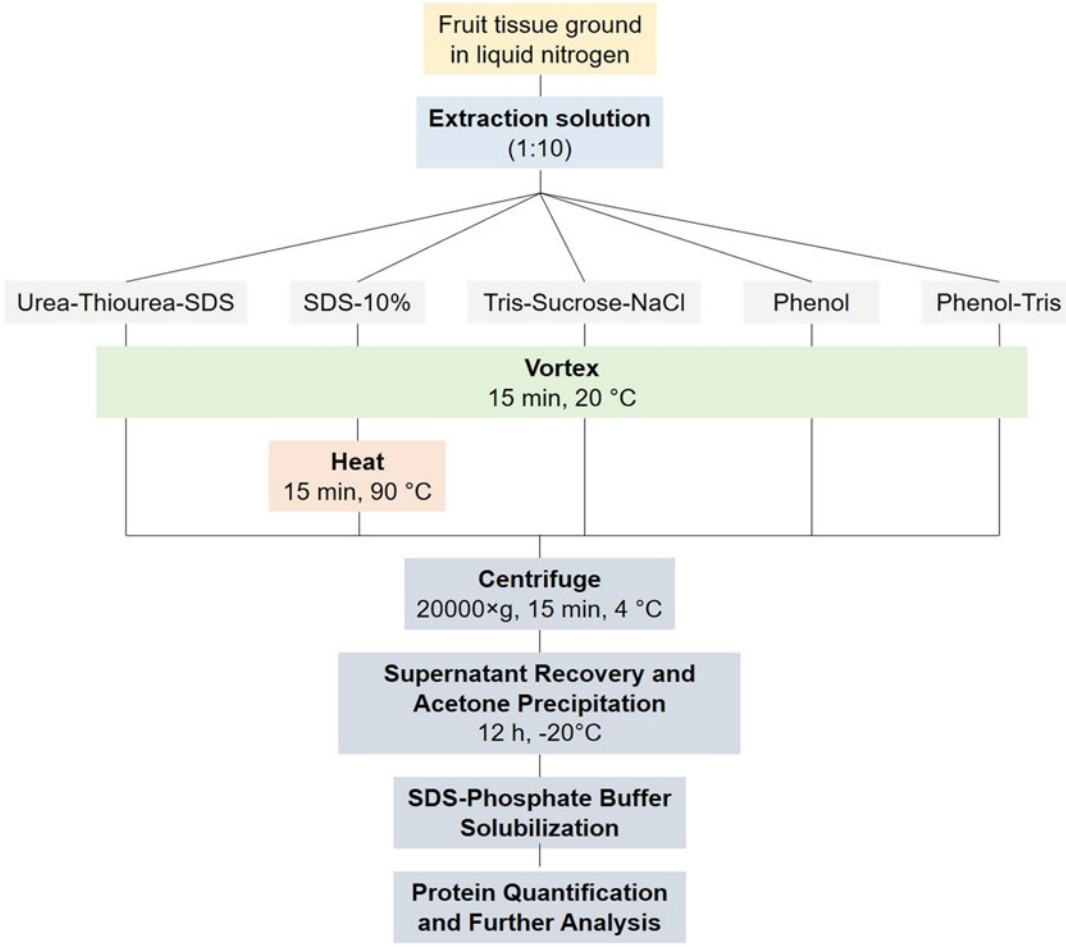


Fig. 1 Flow diagram of workflow for testing protease-knockdown extraction methods in pineapple fruit pulp tissue

Tube D: 0.15 g + 1.5 mL of phenol solution, 10 mM Tris–HCl, pH 8.0, 1 mM EDTA (Sigma-P4557).

Tube E: 0.15 g + 1.5 mL of a 1:1 mixture of Solutions C and D.

2-mercaptoethanol is added to each tube to reach a final concentration of 0.1 M. Solutions A, B, and C are supplemented with 1% of protease inhibitor cocktail (Sigma-P9599). Sample B is incubated for 10 min at 90 °C after mixing. Each tube is homogenized by vortexing for 15 min at 20 °C (*see* **Note 3**).

3. Centrifuge at 10,000 × *g* for 30 min at 4 °C and transfer the supernatant to a 15 mL conical tube.
4. Add ten volumes of ice-cold absolute acetone and incubate for 2–12 h at −20 °C.
5. Centrifuge at 3000 × *g* for 30 min at 4 °C.

6. Remove the supernatant after centrifugation and wash the pellet by vortex with five volumes of 80% ice-cold acetone–20% water solution.
7. Centrifuge at 3000 × *g* for 30 min at 4 °C.
8. Discard the supernatant and let the pellet dry under a laboratory fume hood.
9. Resuspend the dried pellet with 200–500 μL of phosphate-buffered saline (PBS) 1× (Sigma) plus SDS (1%). Vortex and sonicate for 20 min (*see* **Note 4**).
10. Centrifuge at 15,000 × *g* for 10 min at 20 °C. Transfer the supernatant to a new tube and measure protein concentration.
11. Store at −80 °C until use.

3.2 Protein Extract Quality Visualization by SDS-PAGE

1. Prepare a discontinuous SDS-PAGE system of 4% acrylamide for the stacking gel and 12% acrylamide for the resolving gel. Load 10–20 μg of each extract and run at 10 mA/gel through the stacking gel and increase to 25 mA/gel once the samples have entered the separation gel. Stop the run when bromophenol blue reaches the bottom of the gel, and stain with Coomassie Brilliant Blue or SYPRO Ruby Protein Gel Stain® (Thermo-Fisher Scientific). An example gel obtained with this method is shown in Fig. 2.

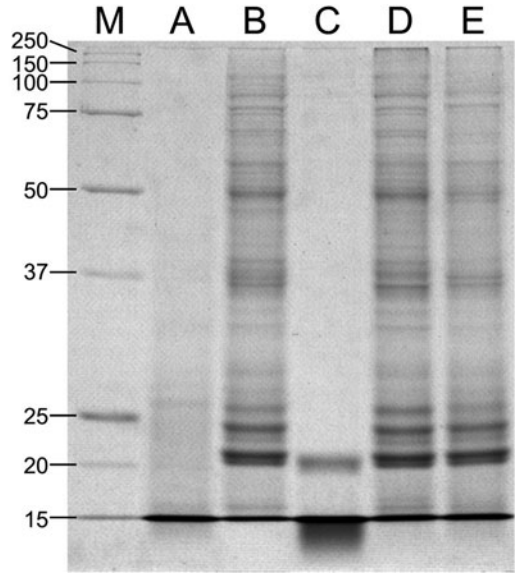


Fig. 2 SDS-PAGE profile of the tested methods for pineapple fruit pulp protein extraction. Despite the content of chaotropic agents and the presence of protease inhibitors, the sample was degraded (A and C). The use of high concentrations of SDS (10%) combined with heating (B), as well as the application of saturated phenol solution (D and E), eliminates the proteolytic activity of the sample, which allows a successful identification of proteins by mass spectrometry. M, Molecular mass marker (kDa)

3.3 Reduction, Alkylation, and Digestion

1. To 100 μg of protein extract, add water to a final volume of 100 μL.
2. Reduce proteins by adding TCEP to a final concentration of 10 mM and incubate for 45 min at 60 °C.
3. To alkylate the proteins, add iodoacetamide (IAM) to a final concentration of 30 mM and incubate in the dark for 60 min at room temperature (20–25 °C). Add dithiothreitol (DTT) to a final concentration of 30 mM and incubate for 10 min at room temperature.
4. Add 1 mL ice-cold absolute acetone and incubate for 2–12 h at −20 °C.
5. Centrifuge at 15,000 × *g* for 15 min at 4 °C. Discard the supernatant and dry the pellet in a vacuum concentrator.
6. Resuspend the pellet with 150 μL 50 mM tetraethylammonium bicarbonate (TEAB) + 0.1% SDS and sonicate for 15 min.
7. Add mass spectrometry grade trypsin in 1:30 (w/w, trypsin–protein) ratio and incubate for 3 h at 37 °C.
8. Centrifuge at 10,000 × *g* for 5 min at 20 °C, add trypsin at 1:60 (w/w, trypsin–protein) ratio, and incubate for 4 h at 37 °C.

3.4 High pH Reversed-Phase Peptide Fractionation and Enrichment

1. Dry the digested sample in a centrifugal vacuum concentrator.
2. Follow the manufacturer's instructions for unlabeled samples using a high-pH reversed-phase fractionation kit (Thermo Scientific-84868). Ideally, collect eight fractions corresponding to 5–50% acetonitrile concentration in the eluent.
3. Dry the fractions in a centrifugal vacuum concentrator and store at −80 °C until analysis by nanoLC-MS/MS.

3.5 LC/MS-MS Analysis

1. Inject suspended samples (20 μL of 0.1% formic acid) into a nanoviper C18 trapcolumn (3 μm, 75 μm × 2 cm, Dionex) at 3 $\mu L \cdot min^{-1}$ flow rate and separate it on an EASY spray C18 RSLC column (2 μm, 75 μm × 25 cm) with a flow rate of 300 $nL \cdot min^{-1}$ connected to an UltiMate 3000 RSLC system (Dionex, Sunnyvale, CA) and interfaced with an Orbitrap Fusion™ Tribrid™ (Thermo-Fisher Scientific, San Jose, CA) mass spectrometer equipped with an "EASY Spray" nano ion source (Thermo-Fisher Scientific, San Jose, CA).
2. For peptide separation, use a chromatographic gradient of MS grade water (solvent A) and 0.1% formic acid in 90% acetonitrile (solvent B) for 30 min, set as follows: 10 min solvent A, 7–20% solvent B within 25 min, 20% solvent B for 15 min, 20–25% solvent B for 15 min, 25–95% solvent B for 20 min, and 8 min solvent A.

3. Operate the mass spectrometer in positive ion mode with nano spray at 3.5 kV and source temperature set at 280 °C. For external calibrants, use caffeine, Met-Arg-Phe-Ala (MRFA), and Ultramark1621. Operate the mass spectrometer in a data-dependent mode to automatically switch between MS and MS/MS.
4. Acquire the full-scan MS spectra on an Orbitrap analyser, set scanning mass range to 350–1500 *m/z* at resolution of 120,000 (FWHM) using an automatic gain control (AGC) setting of 4.0e5 ions, maximum injection time of 50 ms, dynamic exclusion 1 at 90S and 10 ppm mass tolerance. Select a top speed survey scan of 3 s for subsequent decision tree-based Orbitrap collision-induced dissociation (CID) or higher-energy collisional dissociation (HCD) fragmentation. Set the signal threshold for triggering an MS/MS event to 1.0e4 and the normalized collision energy to 35 and 30% for CID and HCD, respectively. Set an AGC of 3.0e4 and isolation window of 1.6 *m/z* for both fragmentations. Additional parameters for CID include activation Q set to 0.25 ms and injection time set to 50 ms. For HCD, first set the mass to 120 *m/z* and injection time to 100 ms. Use the following settings for the decision tree: for HCD fragmentation charge states 2 or 3, scan in a range of 650–1200 *m/z*, charge states 4, scan in a range of 900–1200 *m/z*, and charge states 5, scan in a range of 950–1200 *m/z*; for CID fragmentation charge states 3 scan in a range of 650–1200 *m/z*, charge state 4 scan in a range of 300–900 *m/z*, and charge state 5 in scan range of 300–950 *m/z*. Data were acquired with Xcalibur 4.0.27.10 software (Thermo-Fisher Scientific). An example result of representative chromatograms when applied to nanoLC-MS/MS is shown in Fig. 3.

3.6 Database Search and Protein/Peptide Identification

1. Raw data can be analysed with Proteome Discoverer 2.4 (PD, Thermo Fisher Scientific Inc.) and subsequent searches using Mascot, SQUEST, and MS Amanda engine algorithms, using the *Ananas comosus* proteome database downloaded from UniProt (http://www.uniprot.org/). For searches, full-tryptic protease can be specified with two missed cleavages allowed, while carbamidomethylation of cysteine (+57.021 Da), methionine oxidation (+15.995 Da), and deamidation in asparagine/glutamine (+0.984 Da) are set as dynamic modifications. For the MS^2 method, in which identification can be performed at high resolution in the Orbitrap, precursor and fragment ion tolerances of ±10 ppm and ± 0.2 Da are recommended (*see* **Note 5**).

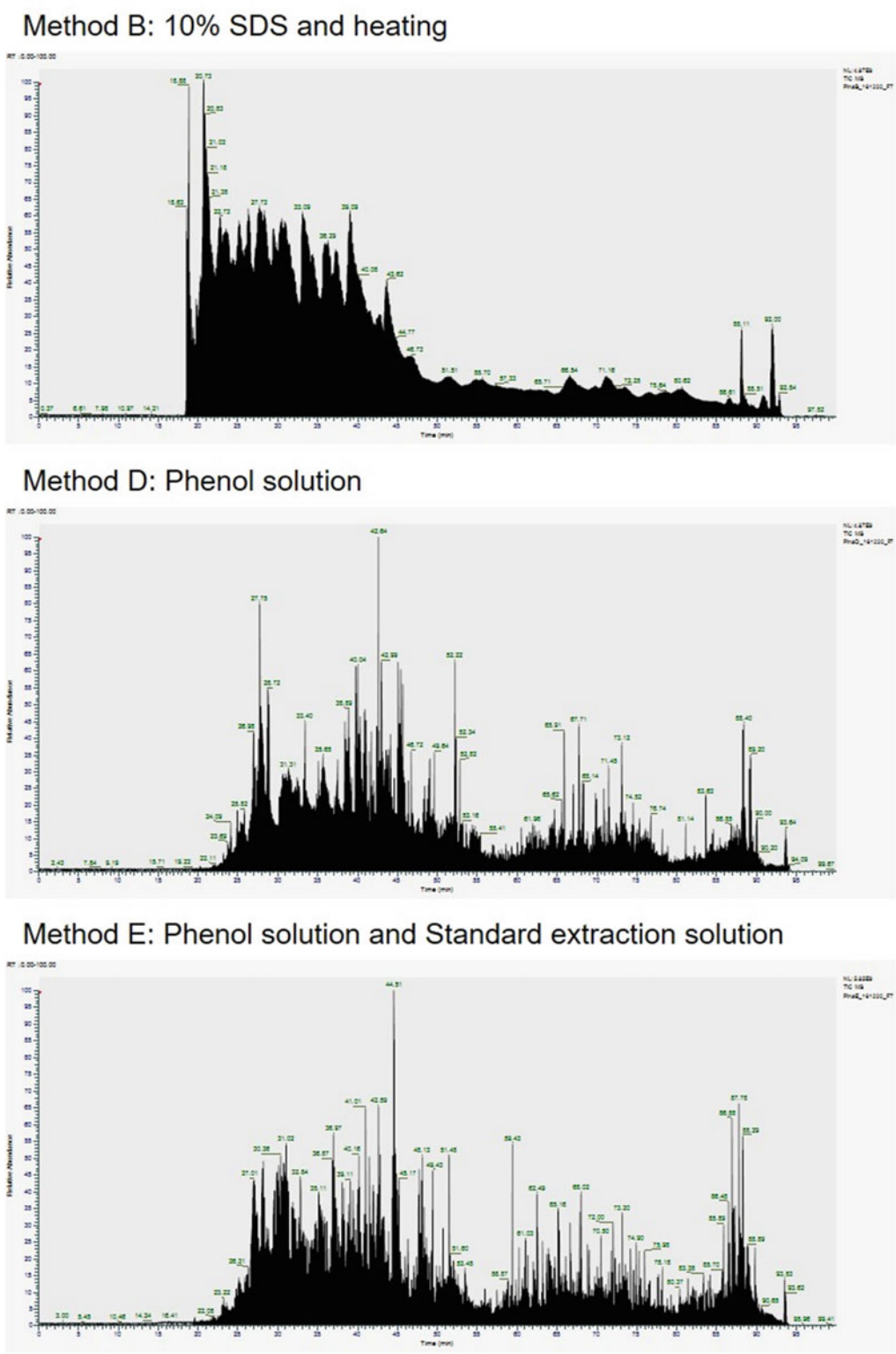


Fig. 3 Representative chromatograms of high-pH reversed-phase fraction 7 when applied to nanoLC-MS/MS of the three protein extraction methods that eliminate tissue protease activity

4 Notes

1. Although there is a wide range of protease inhibitors on the market, it is sometimes difficult to acquire specific cocktails to irreversibly inactivate the different types of enzymes that may be present in a tissue. In addition, the mean lifetime of some of these inhibitors is very short compared with the duration of sample handling processes in proteomics studies.
2. These types of enzymes can remain active even under harsh denaturation conditions or when partially hydrolysed and renaturalize after the solution is changed.
3. Homogenize the sample with extraction solution by vortex immediately after weighing. When using method D, the sample can be maintained overnight at 4 °C or −20 °C if it is homogenized in phenol solution. In this case, the extraction process can be continued the next day.
4. We observed that protease activity was not restored after resolubilizing the extracted protein in PBS with 1% SDS.
5. Following this protocol we identified a total of 3339 proteins with method B, 3050 with method D, and 3111 with method E. 3000 proteins. The easier and rapid method is option D since it does not require the preparation of other solutions or incubation steps for protein extraction.

Acknowledgments

Esaú Bojórquez-Velázquez thanks CONACYT for the postdoctoral fellowship CVU 559120.

References

1. Li T, Wang YH, Liu JX et al (2019) Advances in genomic, transcriptomic, proteomic, and metabolomic approaches to study biotic stress in fruit crops. Crit Rev Biotechnol 39(5):680–692
2. Liu H, Hu M, Wang Q et al (2018) Role of papain-like cysteine proteases in plant development. Front Plant Sci 9:1717
3. Buono RA, Hudecek R, Nowack MK (2019) Plant proteases during developmental programmed cell death. J Exp Bot 70(7): 2097–2112
4. Salguero-Linares J, Coll NS (2019) Plant proteases in the control of the hypersensitive response. J Exp Bot 70(7):2087–2095
5. Schuhmann H, Adamska I (2012) Deg proteases and their role in protein quality control and processing in different subcellular compartments of the plant cell. Physiol Plant 145:224–234
6. Plaxton WC (2019) Avoiding proteolysis during the extraction and purification of active plant enzymes. Plant Cell Physiol 60(4):715–724
7. Hellinger R, Gruber CW (2019) Peptide-based protease inhibitors from plants. Drug Discov Today 24(9):1877–1889
8. Niu L, Yuan H, Gong F et al (2018) Protein extraction methods shape much of the extracted proteomes. Front Plant Sci 9:802
9. Matagne A, Bolle L, El Mahyaoui R et al (2017) The proteolytic system of pineapple stems revisited: purification and characterization of multiple catalytically active forms. Phytochemistry 138:29–51

INDEX

Alejandro Pereira-Santana, Samuel David Gamboa-Tuz and Luis Carlos Rodríguez-Zapata (eds.), *Plant Comparative Genomics*, Methods in Molecular Biology, vol. 2512, https://doi.org/10.1007/978-1-0716-2429-6,

N

O

P

Q

R

S

T

U

V

W

Y

Z