kakapo: easy extraction and annotation of genes from raw RNA-seq reads

Karolis Ramanauskas and Boris Igić

Department of Biological Sciences, University of Illinois at Chicago, Chicago, IL, United States of America

ABSTRACT

kakapo (kākāpō) is a Python-based pipeline that allows users to extract and assemble one or more specified genes or gene families. It flexibly uses original RNA-seq read or GenBank SRA accession inputs without performing global assembly of entire transcriptomes or metatranscriptomes. The pipeline identifies open reading frames in the assembled gene transcripts and annotates them. It optionally filters raw reads for ribosomal, plastid, and mitochondrial reads, or reads belonging to non-target organisms (*e.g.*, viral, bacterial, human). kakapo can be employed for targeted assembly, to extract arbitrary loci, such as those commonly used for phylogenetic inference in systematics or candidate genes and gene families in phylogenomic and metagenomic studies. We provide example applications and discuss how its use can offset the declining value of GenBank's single-gene databases and help assemble datasets for a variety of phylogenetic analyses.

Subjects Bioinformatics, Ecology, Evolutionary Studies, Genomics
Keywords RNA-seq, Python, Phylogenetics, Transcriptome

INTRODUCTION

A variety of biological problems necessitate analyses of genetic sequence data in both basic and applied research. Until recently, there were two primary sources of low-throughput sequence data: newly generated sequences in a laboratory and sequences deposited in public databases, such as GenBank. In the past 20 years, a qualitative transition has taken place, driven by a technological shift in the kind and amount of data generated. The Sequence Read Archive (SRA) for high-throughput, massively parallel, sequencing data at the National Center for Biotechnology Information (NCBI) was established in 2009 (*Kodama, Shumway & Leinonen, 2012*). The size of the SRA, administered by the National Center for Biotechnology Information (NCBI), has since snowballed. It has approximately doubled (to 20 petabytes) just in the last three years, between 2019 and 2022 (*Katz et al., 2022*). In the meantime, the relative value of single-gene submissions to GenBank's non-redundant (nr) database has declined, despite its continued growth, as reliance on the inexpensive and large datasets produced by the new sequencing tools takes hold in many areas of biology.

Currently, about one-third of the SRA database is comprised of RNA-seq data, representing a wide range of taxa and genes appropriate for a variety of analyses at the intersection of genomics and evolution, such as those broadly referred to as "phylogenomics" and "metagenomics" (*Eisen, Kaiser & Myers, 1997*; *Handelsman et al., 1998*). While such data is free and publicly available, it is not necessarily easily accessible.

Submitted 20 January 2023
Accepted 23 October 2023
Published 27 November 2023

Corresponding author
Karolis Ramanauskas,
kraman2@uic.edu

Academic editor
Joseph Gillespie

Additional Information and Declarations can be found on page 10

DOI **10.7717/peerj.16456**

For example, on NCBI's website, only a subset of all SRA datasets can be queried with a sequence-search-tool BLAST, and assemblies are not required to be uploaded along with raw reads. Because of these and other hurdles, the weight of rapidly changing and tedious tasks required to process raw read data often falls on individual researchers. In our experience, many groups are not equipped to take on the significant required diversion to the research program to both enable the use of new tools for sequence analysis and their constant updating. Therefore, the complexity of analyses and lack of integrated tools to conduct them comprise key impediments in the implementation of phylogenomic methods (*Lozano-Fernandez, 2022*; *Wafula et al., 2023*)

kakapo is a modular pipeline intended to remove the barriers to entry for many kinds of phylogenomic analyses (*Ramanauskas & Igić, 2023*). It relies on established tools and stitches them together in a reproducible, self-documenting fashion. The flow of the pipeline is fully defined using one or more configuration files, which are provided as templates with reasonable default values. Without additional intervention, it stores intermediate results and does not repeat previously completed analyses. Prior to the present article, we have previously employed KAKAPO elsewhere in published work (*Sánchez-Cabrera et al., 2021*; *Ramanauskas & Igić, 2021*; *Ruiz-Vargas et al., 2023*). Here, we describe the implementation of key steps in the pipeline, and provide reproducible examples, which showcase its modular features. The key feature of KAKAPO is that it enables a flexibly narrowed focus of computational resources and researcher attention on local (targeted) assembly, with a broad variety of pre- and post-processing options, instead of requiring a global whole-transcriptome assembly.

IMPLEMENTATION

KAKAPO is open source software, released under the GNU General Public License v3.0 and available at https://github.com/karolisr/kakapo. The GitHub repository contains a Wiki with detailed explanations, including step-by-step, multi-platform installation instructions. Below, we give brief instructions and overview of KAKAPO use, as well as example data and analyses.

The pipeline requires Python version 3.8 or higher. It can be installed or upgraded on machines running GNU/Linux or macOS operating systems, using the Python package installer `pip`) by running the following command in a terminal window:

```
pip install --user --upgrade git+https://github.com/karolisr/kakapo
```

Once installed, the program is executed by typing KAKAPO, which prints a brief usage reference. KAKAPO checks if the user's system contains the required dependencies. If any of the dependencies are not found on the system, it attempts to install them automatically to the standard application data directory `$HOME/.local/share/kakapo`. This is done by running:

```
kakapo --install-deps
```

To increase repeatability and reproducibility, a run can be performed with `-force-deps` option, which ignores any dependencies found on the user's system. This ensures that the

same software versions are used for analyses performed at different times, without regard to the presence of the different versions that may be installed system-wide.

Process RNA-seq reads

The pipeline minimally requires the project configuration file.

```
kakapo --cfg project_configuration_file
```

The provided template contains broadly appropriate default values and arguments, so that the user should only need to provide any combination of three types of "targets": (1) SRA accessions, (2) paths to FASTQ files (plain or compressed), (3) a list of FASTA files containing previously assembled transcriptomes (see "Targeted transcript assembly" below). This step then downloads the FASTQ files and the associated metadata for the SRA accessions. Next, both types of FASTQ files—the ones provided by the user and the ones downloaded from NCBI—are processed identically as follows. First, the reads are optionally processed using Rcorrector (*Song & Florea, 2015*) to correct for random sequencing errors. The reads are then trimmed using Trimmomatic (*Bolger, Lohse & Usadel, 2014*) to remove bases with low Phred scores, low quality reads, and any adapter or other Illumina-specific sequences.

Filter RNA-seq reads

Often, RNA-seq reads contain RNAs from unintended or unwanted subjects, such as non-target organelles, microbial infections, and human contaminants. Therefore, KAKAPO provides optional filtering of reads in two different ways.

Filter with Bowtie 2

In the first step KAKAPO uses Bowtie 2 (*Langmead & Salzberg, 2012*) to optionally map the reads to reference genomes. Based on the taxonomic classification, KAKAPO automatically determines which DNA-containing organelles may be found in the sample and constructs a database from the appropriate RefSeq genomes by gradually broadening the taxonomic scope until at least one genome is found. If more than ten appropriate RefSeq genomes exist, a subset of ten of them is selected randomly. Alternatively, if the default behavior is undesirable, the user can override it by providing FASTA files of plastid and/or mitochondrial genomes. Any number of additional reference sequences can be provided as well.

In addition to the Bowtie 2 output in SAM format, KAKAPO produces a separate set of FASTQ files for each given reference. This is useful if, for example, the sample is suspected to be infected with—and contains reads from—a known pathogen.

Filter with Kraken 2

The second filtering step uses Kraken 2 (*Wood, Lu & Langmead, 2019*), and performs rapid taxonomic classification of short reads using prebuilt reference databases. By default, KAKAPO downloads six prebuilt Kraken 2 databases. For small and large subunit ribosomal RNA filtering, KAKAPO uses `16S_Silva138` database made available by the Kraken 2 developers at https://ccb.jhu.edu/software/kraken2. This database is constructed from a diverse set of rRNA sequences provided by SILVA (*Quast et al., 2013*). For bacterial,

archaeal, viral, and human read filtering, Kraken 2 developers provide `minikraken_8GB` database. Additionally, four custom-built databases, derived from RefSeq libraries, are provided: `mitochondrion_and_plastid`, `mitochondrion`, `plastid`, and `viral`. The user can optionally create custom Kraken 2 databases, to filter plasmid, protozoan, fungal, and other reads.

Filtered read set: alternative stopping point

Using only the steps outlined above, the user can use the generated output (a filtered set of FASTQ files) as a largely contaminant-free input dataset for downstream transcriptome assembly. If so, at this point in the workflow, the output can be found in a directory named using the SRA accession or a FASTQ file name provided in the configuration file:

```
[OUTPUT_DIRECTORY]/01-global/05-kraken2-filtered-fq-data/[SAMPLE_NAME]
```

For convenience of those users who simply want to use KAKAPO to process raw RNA-seq reads before using the assembler of their choice to perform full transcriptome assemblies, a `--stop-after-filter` option can be provided, which terminates the pipeline at this point.

Produce BLAST databases for filtered RNA-seq reads

KAKAPO performs two additional steps on the filtered RNA-seq reads. First, seqtk (*Li, 2012*) is used to convert the FASTQ files to FASTA files. Second, BLAST (*Altschul et al., 1990*) databases are generated using `makeblastdb` (included with BLAST+ set of executables) from the FASTA files generated above.

Process query sequences

KAKAPO uses a second type of configuration file in which the user can define one or more "search strategy" entries for the genes they wish to search for in the RNA-Seq reads. Each search strategy entry is intended to encapsulate the information about a gene or a gene family, which can then be used by KAKAPO to find matching RNA-seq reads and assemble transcripts of interest in a targeted manner. The user can tune the parameters of a search strategy to make the search as narrow or as wide as needed. Each search strategy entry can contain a combination of any number of Pfam (*Bateman et al., 2002*) family and NCBI protein accessions, NCBI Entrez query, and a set of user provided FASTA files. KAKAPO then combines the amino acid sequences obtained from different sources, and prunes the set based on the sequence identity and length range limits set by the user (also defined within a search strategy entry).

In order for KAKAPO to process query sequences and to perform the subsequent search, the user needs to provide the path of the search strategies file in addition to the project configuration file described previously:

```
kakapo --cfg project_configuration_file --ss search_strategies_file
```

Search for RNA-seq reads

To find the candidate RNA-seq reads matching the user provided query, KAKAPO first runs `tblastn` (*Altschul et al., 1990*) using the BLAST databases that were built earlier in the pipeline. The parameters used by `tblastn` can be changed in the project configuration

file. However, the settings provided in the template file have been observed to work well in all the cases we tested. When tuning BLAST parameters for this step, the user should err on more permissive values; the parameter values that produce more hits ought to generally be considered superior. This includes larger thresholds for `evalue`, `max_hsps`, and `max_target_seqs` parameters, and lower `qcov_hsp_perc` threshold.

To enrich the set of potentially matching reads, the set of hits from the previous step is passed to VSEARCH (*Rognes et al., 2016*) and used to search the original FASTQ files. Because separate queries may match the same read, the resulting set of reads is checked for duplicate ids and only the reads with unique ids are retained.

The resulting set of reads is a targeted subset of the original sample, biased for reads with higher than average sequence similarity to the query sequence.

Targeted transcript assembly

To assemble the transcripts, KAKAPO runs rnaSPAdes (*Rognes et al., 2016*) using the reads from the previous step as input. The resulting set of transcripts (if any) may or may not be a good match for the gene of interest, to find only those transcripts that match the provided query sequences, `tblastn` is used with the parameter values specified in the search strategies file. If a list of FASTA files containing previously assembled transcriptomes was provided by the user, they are also searched at this step.

Transcript annotation

Next, KAKAPO determines the genetic code for each sample, based on the sample origin (NCBI TaxID) and the genomic source (nucleus, chloroplast (if applicable), mitochondrion). Specifically, the transcripts originating from the reads that mapped to plastid or mitochondrial genomes are assigned the correct genetic code. This is followed by a search for open reading frames (ORFs) in the assembled transcripts, regardless of whether transcriptome assembly was performed using KAKAPO or provided by the user. Alternative ORFs are classified based on parameter settings in the relevant search strategy entry. The highest scoring ORF is labeled as a coding sequence (CDS). Coding sequences are translated and can be optionally piped through InterProScan 5 (*Jones et al., 2014*) for functional annotation.

At the end, KAKAPO produces a series of FASTA files (with associated annotation files in GFF format, where appropriate), containing:

- complete transcript sequences (nucleotide FASTA)
- annotations for the transcripts (GFF)
- extracted ORFs (nucleotide FASTA)
- extracted ORFs (amino acid FASTA)

A graphical overview of the KAKAPO workflow described above is presented in Fig. 1.

EXAMPLE DATA AND ANALYSES

Data

The dataset consists of previously generated reads from a single individual of common house cactus, *Schlumbergera truncata* (“false Christmas cactus” or “crab cactus”; *Ramanauskas*

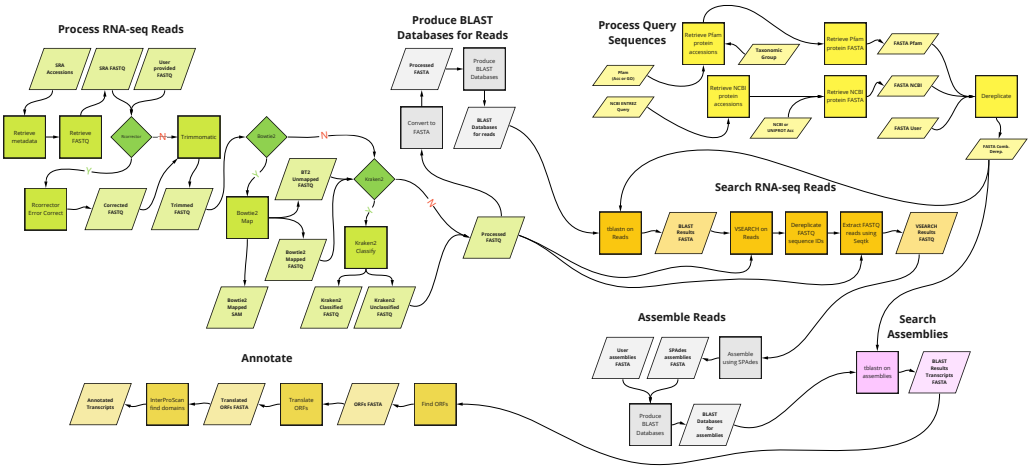

Figure 1 **Overview of the main components of KAKAPO workflow.**
Full-size DOI: 10.7717/peerj.16456/fig-1

& Igić 2021). It is a clonally reproduced epiphyte native to Brazil. A winter-flowering species, it is commonly sold as a houseplant, especially before the short-day peak on either hemisphere. But it is also critically endangered in its native habitat (*Taylor & Zappi, 2017*).

We extracted total RNA from a single style—approximately the size of 1/3 of a thin toothpick—sampled from a *S. truncata* individual accession 19SF1, obtained from the Chicago botanist Joey Santore of *Crime Pays but Botany Doesn't.* Sequencing library was prepared using the KAPA Stranded mRNA-Seq (Roche). The sample was one of eighty-seven total samples sequenced on a single lane of Illumina NovaSeq 6000 platform (paired-end 150 bp reads) at the Duke University Center for Genomic and Computational Biology.

Both example uses employ the same reads, generated from this sample. First, we demonstrate a phylotranscriptomic use, in which we employ a candidate-based approach to sequence and characterize genes responsible for self-incompatibility. Second, we demonstrate metatranscriptomic use, in which we assemble a genome of a common plant pathogen that afflicts the same individual. A working example of KAKAPO with configuration and input files can be found at a dedicated GitHub repository https://github.com/karolisr/kakapo-example. Sequence data is available on GenBank Sequence Read Archive, accession number SRR23214014.

Finding candidate genes in the RNA-seq haystack: self-incompatibility RNases

Most flowering plants are co-sexual—individuals produce both pollen and ovules (*Goldberg et al., 2017*). Such individuals can hypothetically self-fertilize, but most do not (*Raduski, Haney & Igić, 2012*). They instead employ a genetic self-incompatibility mechanism, which sorts incoming pollen, rejecting all pollen that matches a specific genomic region expressed in own their pistils. A widespread self-incompatibility mechanism partly relies on ribonucleases (RNases), specifically those from the T2/S-RNase protein family, called

S-RNases (*Igić, Lande & Kohn, 2008*). In many conservation projects, it is helpful to know the genotype at this locus of individuals that are about to be transplanted to prevent a situation in which reduction of diversity among S-RNases causes cross-incompatibility and reproductive failure (*Wagenius, Lonsdorf & Neuhauser, 2007*). Although it is common houseplant around the world, *Schlumbergera truncata* is endangered in part due to over-collection and habitat degradation (*Taylor & Zappi, 2017*).

Determining the identity of a plant pathogen: cactus virus X

The stems of *Schlumbergera truncata* individual, subject of our study, were unusually purple and experienced occasional unexplained cladode (stem segment) loss. As is the case with other commercially grown and clonally reproduced plants, individuals may acquire a variety of non-lethal chronic infections. Many pathogens may cause similar symptoms, or the infections remain asymptomatic. We use KAKAPO to interrogate the possible causes of a viral infection in Joey Santore's crab cactus. One possible culprit is a well-known ssRNA virus group, broadly termed "Cactus Virus X" (Alphaflexiviridae). Alphaflexiviruses have a single-stranded, positive-sense RNA genome, variable in length but generally approximately 5–9 kb (*Kreuze et al., 2020*).

KAKAPO configuration

KAKAPO was configured to run with all the optional read processing steps turned on: rcorrector, Bowtie 2, and Kraken 2. Bowtie 2 was set to filter mitochondrial and chloroplast reads. To check for the presence of Cactus virus X reads, the complete genome of Cactus virus X (GenBank accession NC_002815.2) was used as an additional reference for Bowtie 2 filtering step. To search for S-RNases, we prepared search strategies for T2/S-RNase gene family members. Additional search strategy for a control gene Elongation factor 1-α 1 or eEF1a1 was prepared. eEF1a1 was highly expressed in all tissues tested in *Ramanauskas & Igić (2021)*. To determine and compare the expression levels of KAKAPO-recovered genes, RNA-seq reads and assembly from *Ramanauskas & Igić (2021)* was used, which included nine *S. truncata* individuals (15H01–15H09).

RESULTS

Genotyping *Schlumbergera truncata* 19SF1 at the self-incompatibility locus

Using the T2/S-RNase search strategy KAKAPO produced eight distinct transcripts. Of these, four contained coding sequences near expected length. Each of the four sequences were identical across assembled lengths to *Schlumbergera truncata* RNases in other individuals, previously described in *Ramanauskas & Igić (2021)*. Two of these matched S-RNase alleles S1 and S2, which were previously sequenced. The remaining two matched non-S-locus (Class I and Class II) RNases, members of this large protein family (Fig. 2A). In Fig. 2B, we show that the expression of these two transcripts is high, we fail to detect other alleles, and implication is that the plant ought not be crossable with 15H01, 15H02, or 15H04, but ought to cross readily with other genotypes.

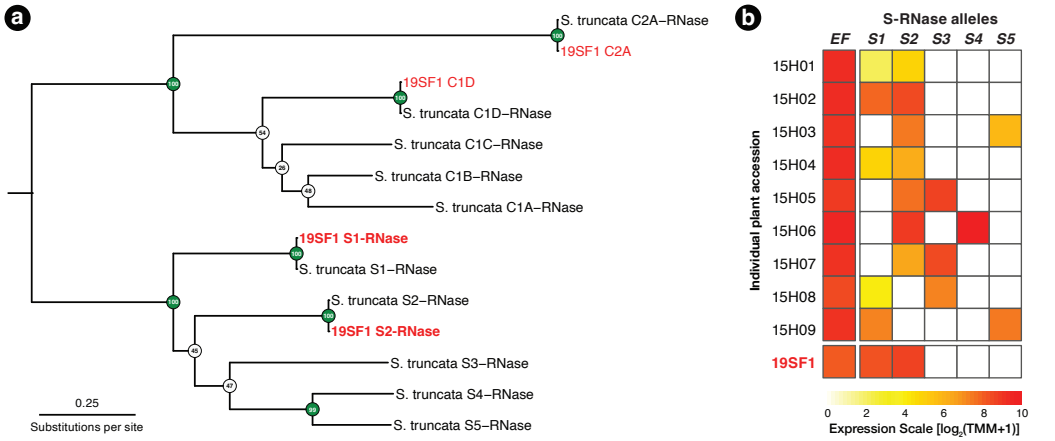

Figure 2 **(A) A RAxML gene tree of *Schlumbergera truncata* T2/S-RNases from *Ramanauskas & Igić (2021)* and KAKAPO-recovered sequences from reads from style tissue RNA-seq of sample 19SF1. Two new S-RNases are recovered, labeled S_1 and S_2. Other genes in the gene family are also found, labeled C1D and C2A, but these belong members of this large gene family with functions unrelated to self-incompatibility(non-S). (B) Summary of S-RNase expression data for nine *S. truncata* individuals discussed in *Ramanauskas & Igić (2021)*, as well as 19SF1, from which two S-allele sequences are extracted by KAKAPO.** Individual plant ID accessions are shown in rows on the left. Control gene Elongation factor 1-α 1 (EF) expression is presented in the first column. Pistil-expressed S-RNase (S_1–S_5) follows the expected pattern. Each genotype is heterozygous (expresses two S-RNase alleles).

Full-size DOI: 10.7717/peerj.16456/fig-2

Cactus virus X in 19SF1

A total of 54,282 RNA-seq reads from individual 19SF1 mapped to the 99.5% of the 6,614 bp Cactus Virus X reference sequence NC_002815.2. One hundred or more reads mapped to 96.13% and fifty or more reads mapped to 98.53% of the reference sequence (Fig. 3A). The genome we recovered is closely related to three CVX sequences (LC128411, KM365479, and AF308158) and not Zygocactus virus X, Schlumbergera virus X, or Pitaya virus X, three of several strains or species in this group (Fig. 3B).

DISCUSSION

KAKAPO allows users to extract and assemble a specified gene or gene family from any number of their own RNA-seq data or deposited sequence accessions. Its utility is considerable, because studies across life sciences increasingly rely on RNA-seq data, and datasets deposited to the NCBI Sequence Read Archive (SRA) are proliferating. In addition to serving as an archive for the original studies, these datasets present an opportunity for novel work, particularly in a variety of research programs concerned with the evolution of gene families. KAKAPO can be deployed in a variety of fields for extraction of particular genes in studies of gene function and evolution, or (for instance) in phylogenetics and systematics to extend coverage for studies that rely on hyb-seq (*Johnson et al., 2018*). Below, we describe similar pipelines, document key differences, and propose major uses of KAKAPO for biologists.

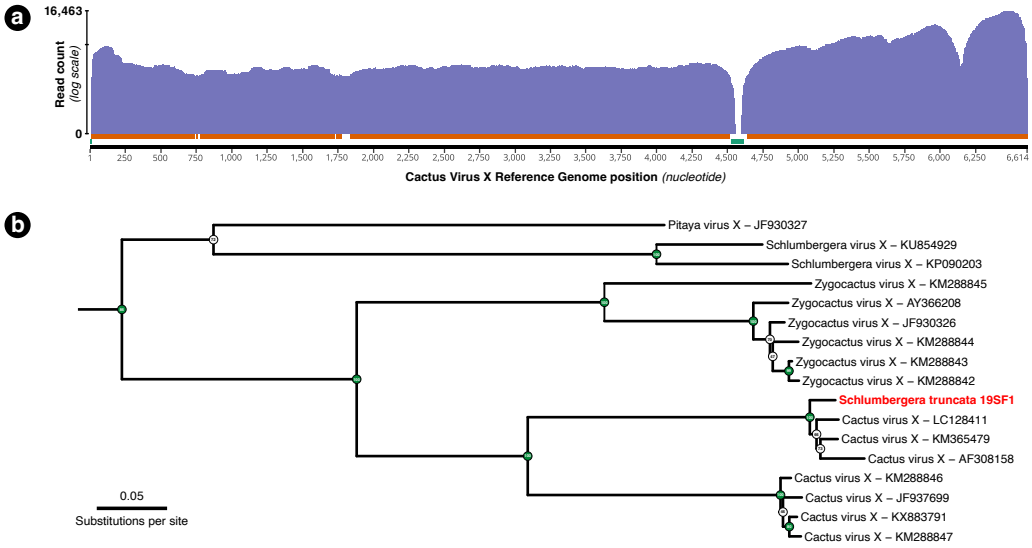

Figure 3 **(A) Coverage plot of the 54,282 *Schlumbergera truncata* RNA-seq reads from sample 19SF1 mapped to Cactus Virus X (CVX) genome. Regions indicated below with the orange bar have coverage of 100 or more reads (96.13%) and those with the green bar have fewer than 50 reads (1.47%). The apparent gap just after nucleotide 4,500 likely reflect a deletion in 19SF1. (B) A RAxML tree of a (previously) unknown pathogen and closely related genome sequences. KAKAPO-recovered sequences (reads from style tissue RNA-seq of sample 19SF1) and GenBank reveal that our sample contains a strain of CVX.**

Full-size DOI: 10.7717/peerj.16456/fig-3

Advantages of targeted assembly

Global and targeted assembly are two common strategies for assembling shotgun reads. Many presently available pipelines accomplish similar targeted (local) assembly tasks as KAKAPO, but it is virtually impossible to catalog and generalize each of their unique implementation steps. Instead of global assembly—often utilized in analyses of differential gene expression or gene content—most comparative phylogenetic and gene family analyses focus only on a subset of genes, perhaps those with a certain level of conservation, previous attention, or involved in particular pathways, responsible for key functions (tools reviewed in *Guo et al. 2019*). The focus on gene families, along with a persistent lack of trustworthy chromosome-level genome assemblies across the tree of life, in many cases obviates the need for attempts at *de novo* transcriptome or genome assemblies. Instead, it may be highly advantageous to target gene families, because they allow both a more focused analysis and efficient computational effort on the processes or genes of interest (*Guo et al., 2019*).

KAKAPO *vs.* related tools

We are not aware of any current tools that flexibly allow gene family detection from a variety of input sources, as implemented in KAKAPO. Many broadly related pipelines are available. For example, an early tool, `SAT-Assembler` focuses on local assembly, but does not perform a variety of pre- or post-processing steps (*Zhang, Sun & Cole, 2014*). Highly focused tools, such as those recently published by *Pucker (2022)* and *Buitkamp (2023)*, provide gene family-specific pipelines. *Pucker (2022)* focuses on identification and annotation of a

single plant gene family (the plant MYBs). It uses "bait" sequences distributed with the tool to target the gene family of interest, which suggests a potential for flexible extension (*Pucker, 2022*). *Buitkamp (2023)* instead takes a mapping (BWA-MEM; *Li 2013*) approach, to specifically recover MHC class I *alleles*. *Wafula et al. (2023)* provide a more general tool (`PlantTribes2` that also performs downstream analyses and customizable visualizations for arbitrary plant gene families from genome and transcriptome assemblies.

KAKAPO is different from most existing comparative gene family tools in some aspects. First, unlike most current tools, it is not organism- or gene family-specific. It is deliberately designed to target arbitrary taxa and genes. Second, it does not require pre-assembled input. Instead, it starts with FASTQ files or SRA accessions (which it corrects, trims, and filters; see Fig. 1, Process RNA-seq Reads). Other similar tools contain a subset of its functionality. KAKAPO workflow steps Process Query Sequences, Search RNA-seq Reads, and Assemble Reads (Fig. 1) may instead be, for example, substituted with tools that employ HMMs built from reference sequences (*Zhang, Sun & Cole, 2014*; *Li et al., 2017*).

The key feature of KAKAPO is the ease with which analyses of additional samples can be performed. The example datasets described above are small, they include only one sample, facilitating the ease of illustrative purposes with KAKAPO distribution. A user may simply add more FASTQ file paths or SRA accessions to the project configuration file, and rerun the analysis. Once the configuration files are created and tested on one sample, KAKAPO is intended to be used in a "set it and forget it" manner. The search strategy files are completely reusable. They are written once by the user, parameters tuned, if needed, and simply reused later. In most cases, only the project name, output directory, SRA accessions, and FASTQ file list will differ between projects. This should be particularly convenient for repeated inquiry in a specific gene family or families (a common theme of research), and provide a straightforward way to ensure easily reproducible research. Finally, KAKAPO design allows easy expansion. In future work, we intend to implement a graphical user interface (GUI), support for alternative software (mappers, assemblers, *etc.*), to allow users to choose a tool that may be better suited for them.

ACKNOWLEDGEMENTS

We ask users of the pipeline to contribute to the Kākāpō Recovery program, aimed at helping ease the persistent threats to the eponymous critically endangered species: https://www.doc.govt.nz/our-work/kakapo-recovery/get-involved/donate.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

This work was supported by the National Science Foundation grant NSF-DEB-1655692 to Boris Igić and Award for Graduate Research, Graduate College, University of Illinois at Chicago to Karolis Ramanauskas. There was no additional external funding received for this study. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
National Science Foundation: NSF-DEB-1655692.

Competing Interests

The authors declare there are no competing interests.

Author Contributions

- Karolis Ramanauskas conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the article, software implementation, and approved the final draft.
- Boris Igić conceived and designed the experiments, performed the experiments, analyzed the data, authored or reviewed drafts of the article, and approved the final draft.

DNA Deposition

The following information was supplied regarding the deposition of DNA sequences:
The sequences are available at GenBank: SRR23214014.

Data Availability

The following information was supplied regarding data availability:
kakapo is available on GitHub and Zenodo: https://github.com/karolisr/kakapo.
Karolis Ramanauskas. (2023). kakapo: Easy extraction and annotation of genes from raw RNA-seq reads. Zenodo. https://doi.org/10.5281/zenodo.8406922.

REFERENCES

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic local alignment search tool. *Journal of Molecular Biology* **215(3)**:403–410 DOI 10.1016/s0022-2836(05)80360-2.

Bateman A, Birney E, Cerruti L, Durbin R, Etwiller L, Eddy SR, Griffiths-Jones S, Howe KL, Marshall M, Sonnhammer ELL. 2002. The Pfam protein families database. *Nucleic Acids Research* **30(1)**:276–280 DOI 10.1093/nar/30.1.276.

Bolger AM, Lohse M, Usadel B. 2014. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* **30(15)**:2114–2120 DOI 10.1093/bioinformatics/btu170.

Buitkamp J. 2023. Uncovering novel MHC alleles from RNA-Seq data: expanding the spectrum of MHC class I alleles in sheep. *BMC Genomic Data* **24**:1 DOI 10.1186/s12863-022-01102-5.

Eisen JA, Kaiser D, Myers RM. 1997. Gastrogenomic delights: a movable feast. *Nature Medicine* **3(10)**:1076–1078 DOI 10.1038/nm1097-1076.

Goldberg EE, Otto SP, Vamosi JC, Mayrose I, Sabath N, Ming R, Ashman T-L. 2017. Macroevolutionary synthesis of flowering plant sexual systems. *Evolution* **71(4)**:898–912 DOI 10.1111/evo.13181.

Guo J, Quensen JF, Sun Y, Wang Q, Brown CT, Cole JR, Tiedje JM. 2019. Review, evaluation, and directions for gene-targeted assembly for ecological analyses of metagenomes. *Frontiers in Genetics* **10**:957 DOI 10.3389/fgene.2019.00957.

Handelsman J, Rondon MR, Brady SF, Clardy J, Goodman RM. 1998. Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. *Chemistry & Biology* **5(10)**:R245–R249 DOI 10.1016/s1074-5521(98)90108-9.

Igić B, Lande R, Kohn JR. 2008. Loss of self-incompatibility and its evolutionary consequences. *International Journal of Plant Sciences* **169**:93–104 DOI 10.1086/523362.

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-S, Baker WJ, Wickett NJ. 2018. A universal probe set for targeted sequencing of 353 nuclear genes from any flowering plant designed using k-medoids clustering. *Systematic Biology* **68(4)**:594–606 DOI 10.1093/sysbio/syy086.

Jones P, Binns D, Chang H-Y, Fraser M, Li W, McAnulla C, McWilliam H, Maslen J, Mitchell A, Nuka G, Pesseat S, Quinn AF, Sangrador-Vegas A, Scheremetjew M, Yong S-Y, Lopez R, Hunter S. 2014. InterProScan 5: genome-scale protein function classification. *Bioinformatics* **30(9)**:1236–1240 DOI 10.1093/bioinformatics/btu031.

Katz K, Shutov O, Lapoint R, Kimelman M, Brister JR, O'Sullivan C. 2022. The sequence read archive: a decade more of explosive growth. *Nucleic Acids Research* **50(D1)**:D387–D390 DOI 10.1093/nar/gkab1053.

Kodama Y, Shumway M, Leinonen R. 2012. The sequence read archive: explosive growth of sequencing data. *Nucleic Acids Research* **40(D1)**:D54–D56 DOI 10.1093/nar/gkr854.

Kreuze JF, Vaira AM, Menzel W, Candresse T, Zavriev SK, Hammond J, Ryu KH, Consortium IR. 2020. ICTV virus taxonomy profile: Alphaflexiviridae. *The Journal of General Virology* **101(7)**:699–700 DOI 10.1099/jgv.0.001436.

Langmead B, Salzberg SL. 2012. Fast gapped-read alignment with Bowtie 2. *Nature Methods* **9(4)**:357–359 DOI 10.1038/nmeth.1923.

Li H. 2012. seqtk, Toolkit for processing sequences in FASTA/Q formats. *Available at https://github.com/lh3/seqtk*.

Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. ArXiv arXiv:1303.3997 DOI 10.48550/arXiv.1303.3997.

Li D, Huang Y, Leung C-M, Luo R, Ting H-F, Lam T-W. 2017. MegaGTA: a sensitive and accurate metagenomic gene-targeted assembler using iterative de Bruijn graphs. *BMC Bioinformatics* **18(12)**:67–75.

Lozano-Fernandez J. 2022. A practical guide to design and assess a phylogenomic study. *Genome Biology and Evolution* **14(9)**:evac129 DOI 10.1093/gbe/evac129.

Pucker B. 2022. Automatic identification and annotation of MYB gene family members in plants. *BMC Genomics* **23(1)**:220 DOI 10.1186/s12864-022-08452-5.

Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, Peplies J, Glöckner FO. 2013. The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Research* **41(D1)**:D590–D596 DOI 10.1093/nar/gks1219.

Raduski AR, Haney EB, Igić B. 2012. The expression of self-incompatibility in angiosperms is bimodal. *Evolution* **66**:1275–1283 DOI 10.1111/j.1558-5646.2011.01505.x.

Ramanauskas K, Igić B. 2021. RNase-based self-incompatibility in cacti. *New Phytologist* **231(5)**:2039–2049 DOI 10.1111/nph.17541.

Ramanauskas K, Igić B. 2023. KAKAPO: easy extraction and annotation of genes from raw RNA-seq reads. *BioRxiv* 2023–2002 DOI 10.1101/2023.02.13.528395.

Rognes T, Flouri T, Nichols B, Quince C, Mahé F. 2016. VSEARCH: a versatile open source tool for metagenomics. *PeerJ* **4**:e2584 DOI 10.7717/peerj.2584.

Ruiz-Vargas N, Ramanauskas K, Tyszka AS, Mason-Gamer RJ, Walker JF. 2023. Transcriptome data from silica-preserved leaf tissue reveals gene flow patterns in a Caribbean bromeliad. *BioRxiv* 2023–2006 DOI 10.1101/2023.06.16.545126.

Sánchez-Cabrera M, Jiménez-López FJ, Narbona E, Arista M, Ortiz PL, Romero-Campero FJ, Ramanauskas K, Igić B, Fuller AA, Whittall JB. 2021. Changes at a critical branchpoint in the anthocyanin biosynthetic pathway underlie the blue to orange flower color transition in *Lysimachia arvensis*. *Frontiers in Plant Science* **12**:633979 DOI 10.3389/fpls.2021.633979.

Song L, Florea L. 2015. Rcorrector: efficient and accurate error correction for Illumina RNA-seq reads. *GigaScience* **4(1)**:48 DOI 10.1186/s13742-015-0089-y.

Taylor N, Zappi D. 2017. *Schlumbergera truncata*. *The IUCN Red List of Threatened Species* **2017**:e.T152554A121599528 DOI 10.2305/iucn.uk.2017-3.rlts.t152554a121599528.en.

Wafula EK, Zhang H, Von Kuster G, Leebens-Mack JH, Honaas LA, dePamphilis CW. 2023. PlantTribes2: tools for comparative gene family analysis in plant genomics. *Frontiers in Plant Science* **13**:1011199 DOI 10.3389/fpls.2022.1011199.

Wagenius S, Lonsdorf E, Neuhauser C. 2007. Patch aging and the S-Allee effect: breeding system effects on the demographic response of plants to habitat fragmentation. *The American Naturalist* **169(3)**:383–397 DOI 10.1086/511313.

Wood DE, Lu J, Langmead B. 2019. Improved metagenomic analysis with Kraken 2. *Genome Biology* **20(1)**:257 DOI 10.1186/s13059-019-1891-0.

Zhang Y, Sun Y, Cole JR. 2014. A scalable and accurate targeted gene assembly tool (SAT-Assembler) for next-generation sequencing data. *PLOS Computational Biology* **10(8)**:e1003737 DOI 10.1371/journal.pcbi.1003737.