HISS: Snakemake-based workflows for performing SMRT-RenSeq assembly, AgRenSeq and dRenSeq for the discovery of novel plant disease resistance genes.

Thomas M Adams[1], Moray Smith[1,2], Micha M Bayer[3], Ingo Hein[1,4]

[1]Department of Cell and Molecular Sciences, The James Hutton Institute, Invergowrie, DD2 5DA, United Kingdom

[2]School of Biology, University of St Andrews, St Andrews, KY16 9ST

[3]Department of Information and Computational Sciences, The James Hutton Institute, Invergowrie, DD2 5DA, United Kingdom

[4]School of Life Sciences, University of Dundee, Dundee, DD1 5EH

Abstract

Background

In the nine years since the initial publication of the RenSeq protocol, the method has proved to be a powerful tool for studying disease resistance in plants and providing target genes for breeding programmes. Since the initial publication of the methodology, it has continued to be developed as new technologies have become available and the increased availability of computing power has made new bioinformatic approaches possible. Most recently, this has included the development of a *k*-mer based association genetics approach, the use of PacBio HiFi data and graphical genotyping with diagnostic RenSeq. However, there is not yet a unified workflow available and researchers must instead configure approaches from various sources themselves. This makes reproducibility and version control a challenge and limits the ability to perform these analyses to those with bioinformatics expertise.

Results

Here we present HISS, consisting of three workflows which take a user from raw RenSeq reads to the identification of candidates for disease resistance genes. These workflows conduct the assembly of enriched HiFi reads from an accession with the resistance phenotype of interest. A panel of accessions both possessing and lacking the resistance are then used in an association genetics approach (AgRenSeq) to identify contigs positively associated with the resistance phenotype. Candidate genes are then identified on these contigs and assessed for their presence or absence in the panel with a graphical genotyping approach that used dRenSeq. These workflows are implemented via Snakemake, a python-based workflow manager. Software dependencies are either shipped with the release or handled with conda. All code is freely available and is distributed under the GNU GPL-3.0 license.

Conclusions

HISS provide a user-friendly, portable, and easily customised approach for identifying novel disease resistance genes in plants. It is easily installed with all dependencies handled internally or shipped with the release and represents a significant improvement in the ease of use of these bioinformatics analyses.

List of abbreviations:

SMRT-AgRenSeq: Single Molecule Real-Time – Association Genetics Resistance Gene Enrichment Sequencing

PacBio – Pacific Biosciences

RenSeq - Resistance gene enrichment Sequencing

dRenSeq – diagnostic Resistance gene Enrichment Sequencing

DAG – Directed Acyclic Graph

TSV – Tab-Separated Values

WGS – Whole Genome Sequencing

NLR – Nucleotide-binding, Leucine-rich Repeat

BED - Browser Extensible Data

SAM – Sequence Alignment Map

VCF – Variant Call Format

SNP – Single Nucleotide Polymorphism

Bp – Base pairs

BLAST – Basic Local Alignment Search Tool

NCBI – National Centre for Biotechnology Information

HISS – HIgh-throughput Smrt-agrenseq Snakemake

Background

Single Molecule Real-Time – Association genetics Resistance gene enrichment Sequencing (SMRT-AgRenSeq) is a recently developed approach for the identification of novel disease resistance genes in plants based on association genetics. Briefly, the approach utilises Pacific Biosciences (PacBio) HiFi Resistance gene enrichment Sequencing (RenSeq) reads to assemble reference contigs of a plant (cultivar or wild species) thought to contain the target resistance gene. Following this, a *k*-mer based association genetics approach [1] identifies contigs that are positively associated with the resistance phenotype. This approach was recently used to identify a candidate for the *Pr3* gene controlling resistance to *Puccinia recondita* f. sp. *secalis* in Rye [2]. These analyses can be further extended by adding a final step of graphical genotyping via diagnostic RenSeq (dRenSeq) [3, 4] to assess the presence and absence patterns of identified candidates to reduce the number of genes that require *in planta* testing.

To perform this analysis, a user must currently install numerous pieces of software such as Canu [5], Bowtie2 [6] and Samtools [7]. A user must further clone down software from several GitHub repositories for NLR-Annotator [8] and Association genetics Resistance gene enrichment Sequencing (AgRenSeq) [1]. Finally, users must be familiar with a scripting language such as bash to write scripts and perform the analyses. In this paper, we present HISS (HIgh-throughput Smrt-agrenseq Snakemake), consisting of three Snakemake workflows constituting: SMRT-RenSeq assembly, AgRenSeq for candidate identification and dRenSeq for

candidate validation. These reside in a single GitHub repository which contains files detailing the rules, yaml files for install of software via conda, and redistributes required software not available via conda. This represents a significant increase in ease of use, as a user now simply needs to: install anaconda or miniconda, install snakemake, optionally create a Snakemake run profile using cookiecutter, and download the latest release from our GitHub repository [9]. Snakemake then handles the execution of specific rules and reports to the user any issues with the configuration provided or errors occurring during the run. An example dataset is also provided to allow users to test the workflow on their systems.

Implementation

HISS contains three Snakemake [10] workflows to perform assembly of HiFi RenSeq reads, SMRT-AgRenSeq [2] to identify novel resistance gene candidates and dRenSeq [3] to validate candidate genes or to determine the presence of known genes (Fig. 1). HISS is distributed with yaml files specifying conda environments to ensure reproducibility and version control of software regardless of the host system. A user simply needs to install either anaconda or miniconda and then install Snakemake into their base environment. All other software dependencies will be installed by the workflows on first run. The workflows themselves consist of a series of rules, which Snakemake uses to produce a directed acyclic graph (DAG) and run the analyses. These workflows perform best on a distributed cluster-style system running a scheduler such as slurm, but they could also be run on a sufficiently well-resourced local machine running a Linux distribution. Users are supplied with template configuration files, which are modified by the user prior to performing the analyses. The separate workflows do not strictly depend on one another. However, we anticipate most users will start with SMRT-RenSeq assembly to produce resistance gene containing contigs for use in the SMRT-AgRenSeq workflow and identify candidate genes for their resistance phenotype of interest. These candidates can then be assessed for presence and absence in a panel of samples via dRenSeq. SMRT-RenSeq assembly requires HiFi reads for accurate reconstruction of nucleotide-binding leucine-rich repeat genes (NLRs). The additional samples used for AgRenSeq and dRenSeq only need short, paired-end Illumina reads, reducing the cost implications of these analyses.

Workflow summary

All the workflows in HISS begin by parsing a tab-separated values (TSV) file into a pandas dataframe to allow easier wildcarding in subsequent rules.

SMRT-RenSeq Assembly

<u>Read trimming</u>

The first step of this workflow is to perform minimal trimming of sequence from the provided HiFi reads. This step uses cutadapt [11] to remove a provided sequence pattern from the 3’ and 5’ ends of the reads. Users may need to conduct additional trimming depending on what pre-processing has been performed on the raw data prior to starting the workflow.

<u>Assembly of reads and basic statistics generation</u>

HiFi reads are assembled into contigs with HiCanu [12] by running Canu [5] with the -pacbio-hifi flag. In addition, useGrid is set to false. When running on a local system, Canu could not utilise grid functionality, whereas on a multi-node cluster system the workflow instead assembles

numerous samples in parallel rather than separating a single job across multiple nodes. As targeted enriched reads are used, assembly is also faster than a whole genome sequencing (WGS) assembly, again meaning the benefits of enabling useGrid would be minimal. The user specifies an expected genome size, which is passed to the genomeSize option and represents the expected size of the targeted enrichment space. The maxInputCoverage is also set to 10,000. An early step of Canu is the assessment of the number of bases in the input reads and comparing this to the provided expected assembly size to assess approximate coverage. If this exceeds the maxInputCoverage value, Canu will randomly subsample reads down to the parameters value. Following the completion of assembly, basic assembly statistics are collated by seqfu [13] and the number of reads and bases used for assembly are parsed out of the HiCanu assembly report.

Identification of putative resistance genes within contigs and assessment of statistics and coverage

NLR-Annotator [8] is used to identify regions in the assembled contigs that contain hallmarks of NLR-like disease resistance genes, based on previously defined MEME motifs [14]. NLR-Annotator version1 and NLR-Parser3 are used with default parameters and custom python code is used to create a summary file of the types of putative NLRs identified across the samples assembled. A custom python script is used to produce a Browser Extensible Data (BED) format file from the NLR-Annotator results. The trimmed HiFi reads are mapped back to the assembled contigs using minimap2 [15] with the map-hifi option. Samtools [7, 16] is used to produce a file detailing the coverage of each putative NLR using the bedcov command. Finally, a custom python script is used to calculate average coverage across the gene and written to a new file. The python scripts are all present in the GitHub repository for the project.

AgRenSeq

Read trimming

Illumina RenSeq reads of the diversity panel provided by the user are pre-processed with fastp using default options [17]. Reads that have already been trimmed and quality filtered experience minimal data loss.

k-mer counting and aggregation

For each set of RenSeq reads, *k*-mers are counted via jellyfish count [18] with options -C -m 51 -s 1G -t 4. A tab-delimited dump file is created using jellyfish dump with options -L 10 -ct. Dump file paths are aggregated into a single file as a prerequisite for the AgRenSeq *k*-mer presence/absence matrix creation. The created matrix contains presence and absence scores for the identified *k*-mers in each sample.

AgRenSeq association scoring

The AgRenSeq matrix is combined with the phenotypic scores specified by the user and the association mapping is performed on each of the RenSeq reference assemblies provided [1]. Each RenSeq assembly is also filtered with NLR-Annotator [8] as a prerequisite for association mapping, to retain only contigs with NLR genes for further analysis.

Result processing and reporting

The result of each AgRenSeq run is provided as a plot of the association scores of each *k*-mer to their respective contig, with the *k*-mers that exceed the predefined association threshold highlighted. To predict the positions within a genome of the contigs that exceed the threshold, RenSeq reference assemblies are mapped to a user-provided genome via BLASTN [19] using default options. Highly associated assemblies are highlighted to indicate physical linkage. Plots are created in R [20] with the libraries dplyr [21] and ggplot2 [22].

dRenSeq

Read trimming

This workflow starts by removing sequencing adaptors provided in user specified fasta files with cutadapt [11]. At this step the reads are also quality-trimmed, retaining a minimum length of 50 bp and using a quality score threshold of 20 at both the 3' and 5' ends of reads in both the R1 and R2 file.

Align reads to reference FASTA file

The user provided reference FASTA file (e.g., contains NLRs derived from the assembled contigs and that display positive association with the traits) is first indexed with bowtie2 [6]. Bowtie2 is again used to align the trimmed reads to the indexed FASTA file, using the sample name as the read group identifier, a user specified minimum mapping score to allow mapping at different mismatch rates, a maximum insert size of 1,000, the --very-sensitive flag, the --no-unal flag to suppress unaligned reads in the output file, the --no-discordant option to keep only concordant alignments and a user-specified maximum number of alignments for each read. The resulting sequence alignment map (SAM) file is then sorted and indexed with samtools [7, 16] to produce a sorted and indexed BAM file.

Assess coverage of reference genes with no mismatches

The BAM file produced above is filtered with sambamba [23] to only keep mappings with the [NM] tag at 0, meaning zero mismatches. Bedtools [24] is used with the coverageBed command to calculate coverage across all sites in the reference FASTA file. Coverage is then calculated across the gene length and written to a single file containing coverage information for all genes in all samples. The file is transposed with pandas to produce a more human-readable coverage file.

Discussion

HISS is shipped with an example dataset, focusing on the potato *Rx* resistance gene which provides resistance against potato virus X in potato [25] with Gemson as a reference resistant cultivar.

The example workflow began with assembling the HiFi reads of Gemson. This produced an assembly of 3,531 contigs representing 42,004,772 base pairs (bp) with an N50 of 12,421 and an auN of 15,062 [26]. NLR-Annotator [8] predicted 2,527 putative NLR genes spread across 1,799 contigs. This consisted of 1,431 NLRs marked as complete and 578 marked as complete pseudogenes. Including partial NLR predictions brought the total NLR genes lacking the pseudogene tag to 1,724 and an additional 803 with the pseudogenes tag. This assembly was used as a reference for the association genetics workflow in which we assessed 117 potato cultivars: 84 negative for *Rx* resistance and 33 containing the *Rx* resistance gene. With an

association threshold of 26, this provided four strong candidate contigs for *Rx*. When analysed by the basic local alignment search tool (BLAST) against version 4.03 of the DM genome [27], the contigs all lay within the same 1Mbp bin on chromosome 12 as expected [25] (Fig. 2). Each of the four contigs had only a single predicted NLR from NLR-Annotator, which were subjected to dRenSeq analysis with the final workflow. Additionally, a panel of previously identified resistance genes from potato, namely: *R1*, *R2-like*, *R3a*, *R3b*, *R8* and *R9a* [28–33], were added to prevent any samples having zero reads mapping to the reference sequences, as this is flagged as an error by the workflow since it can indicate poor sequencing quality or a lack of mappings. Inspecting the coverage values from the dRenSeq analysis (Supplementary Table S1) showed that except for one sample, all four candidates were 100% represented. Only the sample 15_JHL_140_A1 showed 100% coverage of only one candidate. This candidate resides on tig00001343 and following a BLAST against the national centre for biotechnology information (NCBI) nr/nt database [19, 34] showed a hit with the reference *Rx* sequence at 100% identity and 100% query coverage. The remaining three candidates on tig00001423, tig00002055 and tig00002590 yielded top hits for *Rx*/*Gpa2* which are highly similar in sequence. *Gpa2* encodes an NLR that provides resistance against nematodes, which is physically linked to *Rx* [35]. A BLASTN analysis performed using the alternate sequence of *Gpa2* termed *Nem-Gpa2*$^{\Delta C292}$ [4] as a query against the three candidates above showed that tig00002055 has a match with 100% identity and 100% query coverage. Thus, we successfully identified *Rx* and the closely related and physically linked gene *Gpa2* in our SMRT-AgRenSeq analysis. Implementation of dRenSeq identified a rare recombination event in our panel that effectively removed three of the four candidates for identification of *Rx*.

Whilst the above example has focused on a known resistance gene, the strength of this combined approach equally applies to identifying elusive genes. Indeed, SMRT-AgRenSeq has recently been used to identify candidate genes for recalcitrant resistance in Rye [2]. With the addition of our dRenSeq workflow, the number of candidates identified by SMRT-AgRenSeq can further be refined, reducing both the cost and time required to screen candidate genes *in planta*.

Availability:

Project Name: HISS

Project homepage: https://github.com/SwiftSeal/HISS

Operating System(s): Linux

Programming languages: Python, Java, Bash and R

Other requirements: Python 3.10.1, Snakemake 7.12.1, conda 4.13.0, pandas 1.4.3, cookiecutter 2.1.1 (for cluster run profiles only)

License: GNU GPLv3.0

Restrictions to use by non-academics: N/A

Read data for example available at ENA bioprojects: ERP141787 and ERP141790

Declarations:

Competing interests

The authors declare that they have no competing interests.

Funding

This work was supported by the Rural & Environment Science & Analytical Services (RESAS) Division of the Scottish Government through project JHI-B1-1, the Biotechnology and Biological Sciences Research Council (BBSRC) through award BB/S015663/1 and the Royal Society through award NAF\R1\201061. MS was supported through the East of Scotland Bioscience Doctoral Training Partnership (EASTBIO DTP), funded by the BBSRC award BB/T00875X/1.

Authors' contributions

TMA developed the workflows for dRenSeq and SMRT-RenSeq Assembly; MS developed the workflow for AgRenSeq; TMA and MS reviewed the code and contents of the GitHub repository; MB initially automated the dRenSeq approach; TMA and MS drafted the manuscript; MB and IH conceived the integration of dRenSeq in the workflow and substantively revised the manuscript; IH provided the biological material for the study; all authors reviewed and approved the final manuscript.

Acknowledgements

We thank Yuhan Wang, Lynn H Brown and Yuk Woon Cheung for testing the workflows on different datasets. We also thank Miles R. Armstrong for the initial development of the dRenSeq protocol. The authors acknowledge the Research/Scientific Computing teams at The James Hutton Institute and NIAB for providing computational resources and technical support for the "UK's Crop Diversity Bioinformatics HPC" (BBSRC grant BB/S019669/1), use of which has contributed to the results reported within this paper. We also thank John T Jones for his supervision of Moray Smith's PhD and for providing comments on the draft.

Author's information

TMA and MS contributed equally to this work.

References

1. Arora S, Steuernagel B, Gaurav K, Chandramohan S, Long Y, Matny O, et al. Resistance gene cloning from a wild crop relative by sequence capture and association genetics. Nat Biotechnol 2019 372. 2019;37:139–43; doi:10.1038/s41587-018-0007-9.

2. Vendelbo NM, Mahmood K, Steuernagel B, Wulff BBH, Sarup P, Hovmøller MS, et al. Discovery of Resistance Genes in Rye by Targeted Long-Read Sequencing and Association Genetics. Cells 2022, Vol 11, Page 1273. 2022;11:1273; doi:10.3390/CELLS11081273.

3. Van Weymers PSM, Baker K, Chen X, Harrower B, Cooke DEL, Gilroy EM, et al. Utilizing "Omic" technologies to identify and prioritize novel sources of resistance to the oomycete pathogen phytophthora infestans in potato germplasm collections. Front Plant Sci. 2016;7:672; doi:10.3389/FPLS.2016.00672/BIBTEX.

4. Armstrong MR, Vossen J, Lim TY, Hutten RCB, Xu J, Strachan SM, et al. Tracking disease resistance deployment in potato breeding by enrichment sequencing. Plant Biotechnol J. 2019;17:540–9; doi:10.1111/PBI.12997.

5. Koren S, Walenz BP, Berlin K, Miller JR, Bergman NH, Phillippy AM. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. Genome Res.

2017;27:722–36; doi:10.1101/GR.215087.116.

6. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods 2012 94. 2012;9:357–9; doi:10.1038/nmeth.1923.

7. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25:2078–9; doi:10.1093/bioinformatics/btp352.

8. Steuernagel B, Witek K, Krattinger SG, Ramirez-Gonzalez RH, Schoonbeek HJ, Yu G, et al. The NLR-annotator tool enables annotation of the intracellular immune receptor repertoire. Plant Physiol. 2020;183:468–82; doi:10.1104/pp.19.01273.

9. Adams T, Smith M. HISS. 2022. https://doi.org/10.5281/ZENODO.7271099

10. Köster J, Mölder F, Jablonski KP, Letcher B, Hall MB, Tomkins-Tinch CH, et al. Sustainable data analysis with Snakemake. F1000Research. 2021;10:33; doi:10.12688/f1000research.29032.2.

11. Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.journal. 2011;17:10; doi:10.14806/ej.17.1.200.

12. Nurk S, Walenz BP, Rhie A, Vollger MR, Logsdon GA, Grothe R, et al. HiCanu: Accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. Genome Res. 2020;30:1291–305; doi:10.1101/GR.263566.120.

13. Telatin A, Fariselli P, Birolo G. Seqfu: A suite of utilities for the robust and reproducible manipulation of sequence files. Bioengineering. 2021;8:59; doi:10.3390/bioengineering8050059.

14. Jupe F, Pritchard L, Etherington GJ, MacKenzie K, Cock PJA, Wright F, et al. Identification and localisation of the NB-LRR gene family within the potato genome. BMC Genomics. 2012;13:1–14; doi:10.1186/1471-2164-13-75/FIGURES/5.

15. Li H. Minimap2: Pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34:3094–100; doi:10.1093/bioinformatics/bty191.

16. Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, et al. Twelve years of SAMtools and BCFtools. Gigascience. 2021;10:1–4; doi:10.1093/GIGASCIENCE/GIAB008.

17. Chen S, Zhou Y, Chen Y, Gu J. fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics. 2018;34:i884–90; doi:10.1093/BIOINFORMATICS/BTY560.

18. Marçais G, Kingsford C. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics. 2011;27:764–70; doi:10.1093/BIOINFORMATICS/BTR011.

19. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215:403–10; doi:10.1016/S0022-2836(05)80360-2.

20. R Core Team (R Foundation for Statistical Computing). R: A language and environment for statistical computing. 2022; https://www.r-project.org. Accessed 28 Oct 22

21. Wickham H, François R, Henry L, Müller K. dplyr: A Grammar of Data Manipulation. 2022; https://dplyr.tidyverse.org. Accessed 28 Oct 22

22. Wickham H. ggplot2: Elegant Graphics for Data Analysis. 2016; https://ggplot2.tidyverse.org. Accessed 28 Oct 22

23. Tarasov A, Vilella AJ, Cuppen E, Nijman IJ, Prins P. Sambamba: fast processing of NGS alignment formats. Bioinformatics. 2015;31:2032–4; doi:10.1093/BIOINFORMATICS/BTV098.

24. Quinlan AR, Hall IM. BEDTools: A flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26:841–2; doi:10.1093/bioinformatics/btq033.

25. Bendahmane A, Kanyuka K, Baulcombe DC. High-resolution genetical and physical mapping of the Rx gene for extreme resistance to potato virus X in tetraploid potato. Theor Appl Genet 1997 951. 1997;95:153–62; doi:10.1007/S001220050543.

26. Li H. auN: a new metric to measure assembly contiguity. 2020. https://lh3.github.io/2020/04/08/a-new-metric-on-assembly-contiguity. Accessed 28 Oct 2022.

27. Pham GM, Hamilton JP, Wood JC, Burke JT, Zhao H, Vaillancourt B, et al. Construction of a chromosome-scale long-read reference genome assembly for potato. Gigascience. 2020;9:1–11; doi:10.1093/GIGASCIENCE/GIAA100.

28. Ballvora A, Ercolano MR, Weiß J, Meksem K, Bormann CA, Oberhagemann P, et al. The R1 gene for potato resistance to late blight (Phytophthora infestans) belongs to the leucine zipper/NBS/LRR class of plant resistance genes. Plant J. 2002;30:361–71; doi:10.1046/J.1365-313X.2001.01292.X.

29. Huang S, Van Der Vossen EAG, Kuang H, Vleeshouwers VGAA, Zhang N, Borm TJA, et al. Comparative genomics enabled the isolation of the R3a late blight resistance gene in potato. Plant J. 2005;42:251–61; doi:10.1111/J.1365-313X.2005.02365.X.

30. Lokossou AA, Park T ho, van Arkel G, Arens M, Ruyter-Spira C, Morales J, et al. Exploiting Knowledge of R/Avr Genes to Rapidly Clone a New LZ-NBS-LRR Family of Late Blight Resistance Genes from Potato Linkage Group IV. Mol Plant-Microbe Interact. 2009;22:630–41; doi:10.1094/MPMI-22-6-0630.

31. Li G, Huang S, Guo X, Li Y, Yang Y, Guo Z, et al. Cloning and Characterization of R3b; Members of the R3 Superfamily of Late Blight Resistance Genes Show Sequence and Functional Divergence. Mol Plant-Microbe Interact. 2011;24:1132–42; doi:10.1094/MPMI-11-10-0276.

32. KwangRyong J. Unveiling and deploying durability of late blight resistance in potato: from natural stacking to cisgenic stacking [PhD thesis]. 2013. [Wageningen (NL)]: University of Wageningen

33. Vossen JH, van Arkel G, Bergervoet M, Jo KR, Jacobsen E, Visser RGF. The Solanum demissum R8 late blight resistance gene is an Sw-5 homologue that has been deployed worldwide in late blight resistant varieties. Theor Appl Genet. 2016;129:1785–96; doi:10.1007/S00122-016-2740-0/FIGURES/5.

34. Sayers EW, Bolton EE, Brister JR, Canese K, Chan J, Comeau DC, et al. Database resources of the national center for biotechnology information. Nucleic Acids Res. 2022;50:D20–6; doi:10.1093/NAR/GKAB1112.

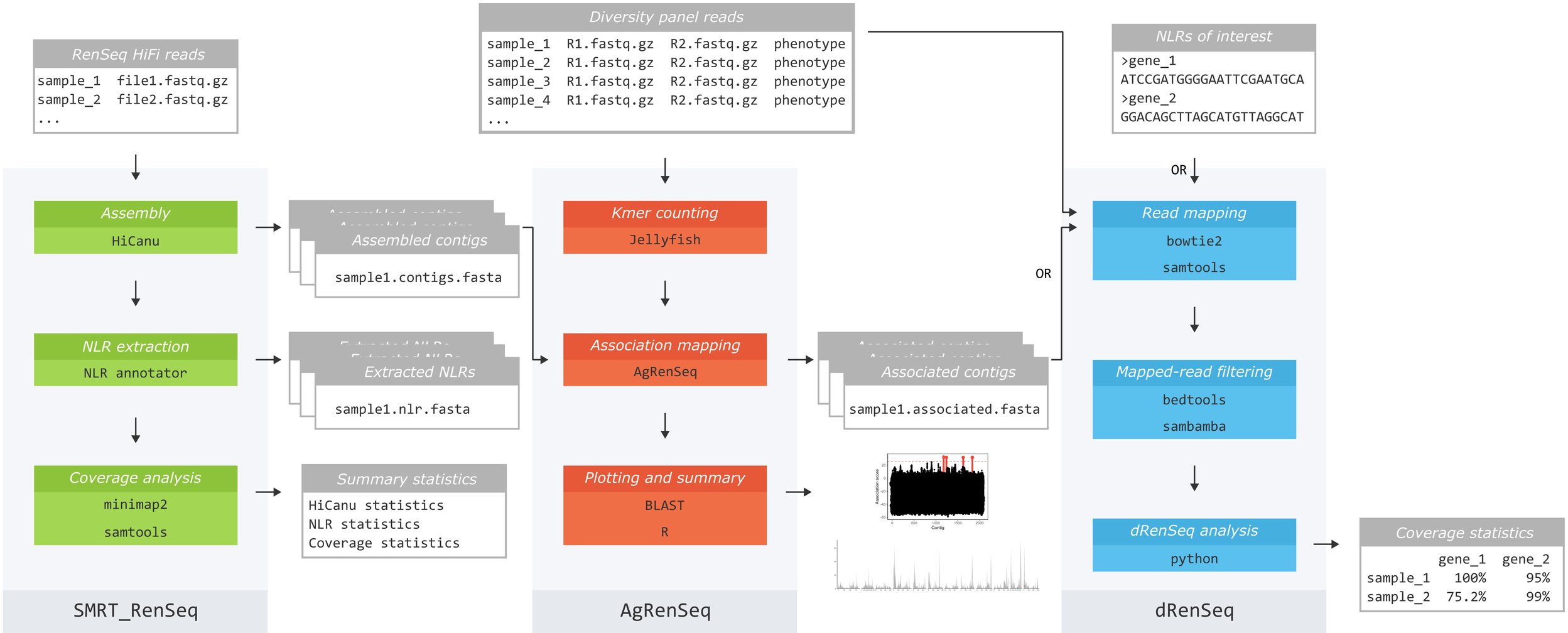
35. Rouppe Van Der Voort J, Wolters P, Folkertsma R, Hutten R, Van Zandvoort P, Vinke H, et al. Mapping of the cyst nematode resistance locus Gpa2 in potato using a strategy based on comigrating AFLP markers. Theor Appl Genet 1997 955. 1997;95:874–80; doi:10.1007/S001220050638.

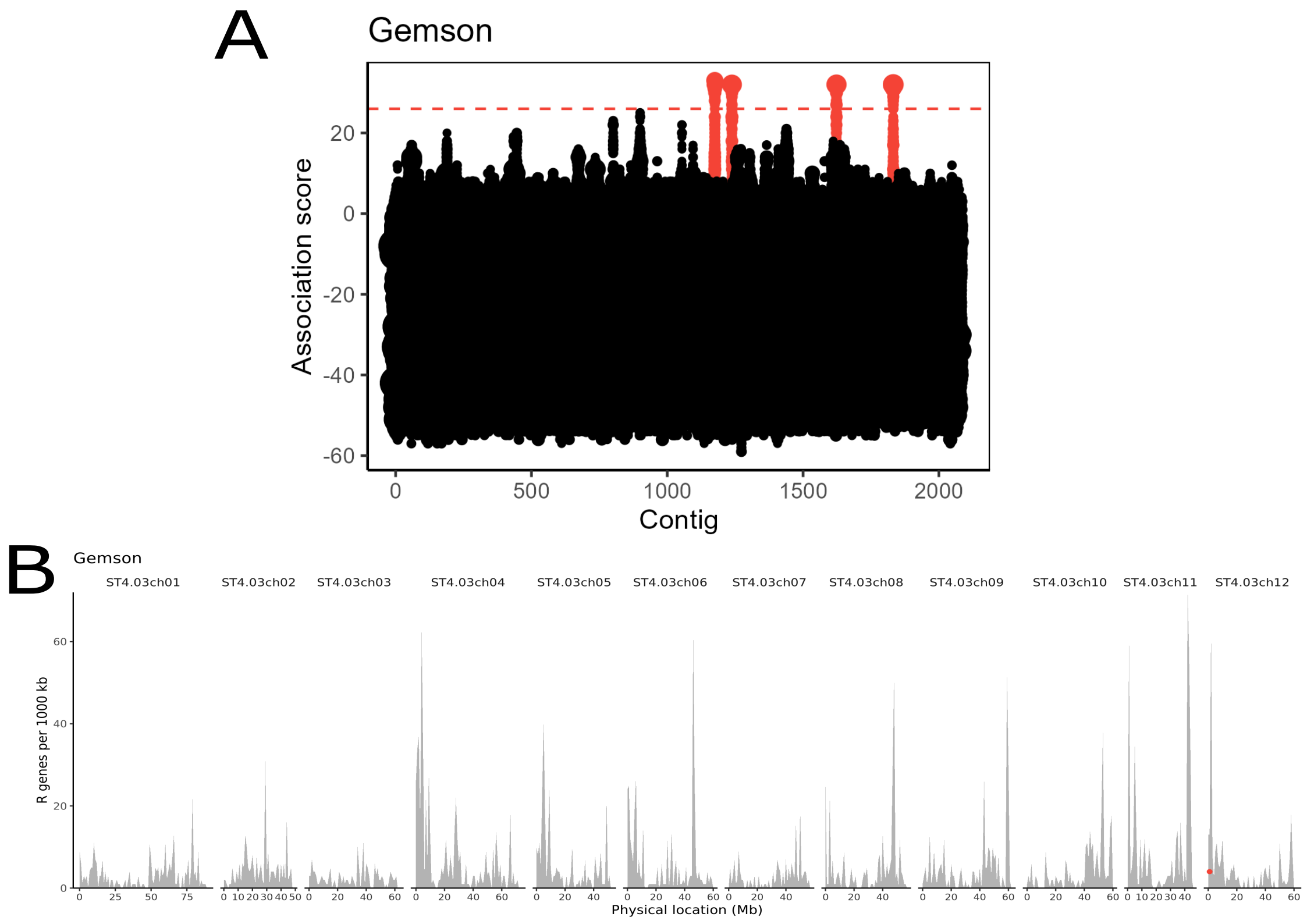
Figure Legends:

Fig. 1: Outline of the snakemake RenSeq workflows. SMRT-RenSeq (green) assembles RenSeq HiFi reads and produces assembly and NLR summary statistics. AgRenSeq (red) takes a metadata file of diversity panel reads and can use the output of SMRT-RenSeq as a reference for *k*-mer mapping. It outputs highly associated contigs and NLR loci as well as *k*-mer scoring plots and mapping of contigs to a reference genome. dRenSeq (blue) can use a list of NLRs of interest or the output from AgRenSeq to calculate read coverage.

Fig. 2: Output plots produced by the Ag-RenSeq snakemake workflow. **A** Dot plot of the analysed contigs in the Ag-RenSeq workflow. Each contig in the assembly is arranged on the x-axis, with *k*-mers plotted on the y-axis. The size of the dot represents the number of *k*-mers. Contigs with *k*-mers above a threshold association score of 26 are highlighted in red. **B** A plot of NLRs by chromosomal location based on the DM reference assembly. The grey bars represent the number of NLRs per 1,000 kb. The red dot represents the location of contigs scored as positive by the association genetics following a BLAST analysis.

Supplemental Table S1: Transposed coverage values table for the example workflow. The reference *R* genes have been removed for clarity. Sample coverage values are colour coded with green representing the highest value and orange representing the lowest value. All samples with the *Rx* gene, except one, show 100% coverage in all four candidates. The 15_JHL_140_A1 represents a rare recombination event in this resistance gene cluster, indicating tig00001343_nlr_1 as a strong candidate for *Rx*.


RenSeq HiFi reads
sample_1 file1.fastq.gz
sample_2 file2.fastq.gz
...
Assembly
HiCanu
NLR extraction
NLR annotator
Coverage analysis
minimap2
samtools
Assembled contigs
sample1.contigs.fasta
Extracted NLRs
sample1.nlr.fasta
Summary statistics
HiCanu statistics
NLR statistics
Coverage statistics
SMRT_RenSeq
Diversity panel reads
sample_1 R1.fastq.gz R2.fastq.gz phenotype
sample_2 R1.fastq.gz R2.fastq.gz phenotype
sample_3 R1.fastq.gz R2.fastq.gz phenotype
sample_4 R1.fastq.gz R2.fastq.gz phenotype
...
Kmer counting
Jellyfish
Association mapping
AgRenSeq
Plotting and summary
BLAST
R
AgRenSeq
Associated contigs
sample1.associated.fasta
OR
NLRs of interest
>gene_1
ATCCGATGGGGAATTCGAATGCA
>gene_2
GGACAGCTTAGCATGTTAGGCAT
OR
Read mapping
bowtie2
samtools
Mapped-read filtering
bedtools
sambamba
dRenSeq analysis
python
dRenSeq
Coverage statistics
gene_1 gene_2
sample_1 100% 95%
sample_2 75.2% 99%


A
Gemson
Association score
20
0
-20
-40
-60
0
500
1000
1500
2000
Contig
B
Gemson
ST4.03ch01
ST4.03ch02
ST4.03ch03
ST4.03ch04
ST4.03ch05
ST4.03ch06
ST4.03ch07
ST4.03ch08
ST4.03ch09
ST4.03ch10
ST4.03ch11
ST4.03ch12
R genes per 1000 kb
Physical location (Mb)