Organization And Introgression Mechanics Of Phaseolus Vulgaris (Common Bean)

Seth Munholland
University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation
https://scholar.uwindsor.ca/etd/8323

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.
ORGANIZATION AND INTROGRESSION MECHANICS OF *PHASEOLUS VULGARIS* (COMMON BEAN)

by

Seth Munholland

A Dissertation
Submitted to the Faculty of Graduate Studies
through Biological Sciences
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada

© 2020 Seth Munholland
ORGANIZATION AND INTROGRESSION MECHANICS OF PHASEOLUS VULGARIS (COMMON BEAN)

by

Seth Munholland

APPROVED BY:

________________________
M. Daley, External Examiner
University of Western Ontario

________________________
L. Rueda
School of Computer Science

________________________
M. J. Crawford
Department of Biomedical Sciences

________________________
T. Noel
Department of Integrative Biology

________________________
W. L. Crosby, Advisor
Department of Biomedical Sciences

November 27, 2019
DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.
ABSTRACT

*Phaseolus vulgaris* is a major food crop grown and consumed around the world. A new world vegetable, the common bean underwent two separate domestication events, both pre-Columbus. These events generated two different land races, the Mesoamerican and Andean, named for the area where the domestication took place. Since the initial domestications the land races have been generally evenly cultivated, but despite its popularity the common bean has only very recently been fully sequenced. One of the issues faced by bean growers worldwide is Common Bacterial Blight (CBB). A disease caused by *Xanthomonas axonopodis*, CBB causes crop loses ranging from 20–40% every year but does not affect all species within *Phaseolus* evenly; *P. acutifolius*, for example, shows an innate resistance to CBB. To leverage this advantage, researchers at the University of Guelph, in partnership with the Ontario Agricultural College, developed a cultivar of Mesoamerican *P. vulgaris* that was introgressed with PI440795, a *P. acutifolius* accession, and backcrossed repeatedly with several other Mesoamerican *P. vulgaris* accessions to generate ‘OAC-Rex’, a plant that displays the crop-desired traits of *P. vulgaris* and the disease resistance traits of *P. acutifolius*. Genetic introgression is the process of crossing distantly related organisms followed by repeated backcrossing, resulting in a viable offspring that displays characteristics of each parent. Though rarely occurring, it can be observed in both plants and animals and is often exploited in a crop development context to generate new cultivars. Unfortunately, though regularly observed, introgression has been followed on a predominantly phenotypic level, usually many generations after the event, and as such molecular aspects of this phenomenon are largely unknown.

By studying OAC-Rex, PI440795, and G-19833 (an Andean cultivar whose whole-genome has been published) introgression was examined directly and a method for the detection of regions within the introgressed genome uniquely donated from either parent was developed. A computational examination of these regions revealed an apparent lack of patterning associated to known recombination fingerprints. That said, the scale of the introgression led to the proposed hypothesis that introgression follows a high number of short gene conversion like events. Given the genomic resources already available, *Arabidopsis thaliana* or *Saccharomyces* spp. were suggested as viable
alternatives to refine the proposed method and/or confirm/refute the gene conversion hypothesis.
To Mom and Dad

– Or –

To me, in all my glory; I have done the impossible and that makes me mighty
ACKNOWLEDGEMENTS

It would be wrong of me to start this addressing anybody but Dr. Bill Crosby. Years ago, you gave me the opportunity to join a lab after the briefest (and perhaps weirdest) of interviews I think I’ve ever had. Since that day you’ve been a reliable, kind, and generous mentor. The breadth and depth of your knowledge is staggering, as is your utter mastery of the English language. You’ve been open to all my inane ramblings, trusted me far more than I probably should have been, and always sought to help me grow as a researcher and a human being. The words do not exist to properly express my gratitude for everything you’ve done for me.

To Dr. Don Karl Roberto, who was stuck dealing with me more than anyone, a more patient soul I shall never meet. Thank you for our science chats, our philosophical, grammatical, pedantic, and bureaucratic debates, a shared love of everything a long time ago in a galaxy far, far away, and thank you for helping me sounds as smart as I think I am.

Thank you to Claudia DiNatale, who got me started on the bioinformatics path when I first joined the lab with the impetus from Dr. Hugh Fackrell. I still have no idea what Perl script you were debugging, but I’m glad I recognized the code and asked. This whole work would not have happened had you not been there and been so willing to indulge this curious computer geek.

Dr. Mohammad Haj Dezfulian, possibly the smartest man I’ve ever known, thank you for teaching me the limits of my own intelligence and always pushing me to further them.

Next, I want to extend a heartfelt thank you to my M.Sc. And Ph.D. committee members; Dr. Michael Crawford, Dr. Luis Rueda, and Dr. Tanya Noel. You’ve stuck with me as I bounced between ideas before finally (maybe) figuring out what was going on. I’d like to thank Dr. Mark Daley for graciously agreeing to be my external examiner, and <chair> for taking the time to chair my defence. A special commendation goes to Dr. Oliver Love, for going above and beyond in the role of chair during my transfer exam and being a proxy interrogator.

I would like to thank my everyone else who’s been in Crosby Lab with me over the years; Dr. Espanta Jalili, Jacob Bender, Curtis Foreman, Florida Doci, Bledi Elshani,
Evgeni Gentchev, and Kerry Khoo. Over the years and in your own ways you made this lab a second home to me.

On the technical side of things, I owe much of this project to Adrian Platts and Brad Jones. Adrian, you helped me get my feet under me when I first started and got dropped into the middle of an ongoing work that I was not prepared for. You helped me learn the foundations of assembly and spread from there to all things bioinformatics. It’s only in hindsight that I realize how asinine some of the questions I had were, thank you for walking me through it all. Brad, my coding was a joke when I started this, and while it’s still good for a laugh, you’ve made me so much better than I could have gotten on my own. Thank you for taking time out of your day every time I had a question and turning them into lessons that helped me develop an invaluable skillset.

To my parents, already dedicated, but not properly acknowledged; thank you for never making me feel like this was a poor decision, for bailing me out when I got in over my head, and for being good parents nagging a delinquent child into doing his homework. Mom, our nightly phone calls helped me more than I think I’ve acknowledged, especially once I was on my own. Dad, I honestly can’t remember an instance when you didn’t make time for me whenever I asked, and it’s helped me process a lot. I can’t forget my siblings; Jessi our chats when I first moved out here helped me settle down, and Ty our chats every time I came home helped me realize how much I was learning (and how much I still got wrong). Finally, aunt Iris, thank you for being you. It’s always a fun time when you’re around.

My many friends in Windsor through these years have helped me stay sane and while they may not have contributed to the science, they were invaluable. Joe Duke, my one-time hetero life mate, was exactly the friend I needed when I started grad school and was the best roommate I could ever ask for. I have been, and always shall be, your friend. Brayden Labute and Gianfranco Grande, you’ve been in the bio building as long as I have and this last year with Monday D&D has been something I didn’t realize I was missing until I got back into it. You adjusted your lives around it every week and it’s brought me so much happiness, +50XP. To the former D&D posse; Danielle Gunsch, Alyssa Frazao, and Kyle Stokes, you’ve been friends through my time here and joined me in various positions on the DBGSA. I have only fond memories of each of you. I’ve
spent many a BioSocial with Lincoln Savi sharing my love of board games and all things 3D printing. Those BioSocial board games would not be the same without Nick Tran, who always seemed to have a plan, and Natalie Gosselin, with whom I still must disagree on principal. Thanks to Mohammed Bourouh I will never look at mango juice the same ever again. To Ellen “Wednesday” Laurie and Tina Suntress I want to say thanks for being a regular in the early days of the BioSocial and helping me plan all the social events, even after you graduated. Patricia Okpara, thank you for letting me not be the only one who had their freak flag flying high; Carol-Anne Barlow, thank you for listening and being curious while you kept me looking like an academic; and, lastly, Robert Gombar, while you and I may have taken different paths in grad school, your friendship and appreciation of cognac before a final made them much less of an ordeal.

I owe a special thank you to Sonmy Busto. You’ve been there for me through so much that I don’t know where to start. You’ve believed in me and supported me when I was at my lowest, you encouraged me to keep at it and knew I would accomplish this from the get-go. You’re very dear to me and though our relationship has changed many times over the many years I will always be there for you.

Equally important in keeping me sane, though from half a country away, are my friends back home in Calgary. Gwynne Morgan takes the cake here. While Don was putting up with me in person, you were there electronically. Our daily chats gave me a solid anchor and lifeline to the life I, rather abruptly, put on hold to come back to school. You’ve helped me see a lot of flaws in who I was and have been extraordinarily patient in helping me sort through those short-comings and grow. Thank you from the bottom of my heart. Perhaps my oldest friend, Luke Pronyshyn, made me feel like I was gone for a weekend whenever I came back from a four-month semester. The first and last pint back home were something I looked forward to and cherished dearly. To Marissa Heatherington, though my scholastic journey had already begun when we met you’ve been an ever-reliable ray of cheery sunshine since. You’ve noticed when I’ve been off even before I have, and I’ve come to rely on your friendship and support. I look forward to many a late-night pie in the future. Jon Larson, troll of trolls, you’ve been a friend in frustration and a fellow nerd when I needed someone to grumble or geek out with. To Jordan Nichols and Carson Wood thank you both for making time whenever I’ve been
home to let me have an ordinary Saturday with friends. Finally, to Lee Deguire for taking care of me that one time and Andrew Holt for always being someone I can turn to when I have a question, regardless of the topic.

From the ISCB I want to thank Belinda Hanson and Diane Kovats. I have no idea how I impressed both of you to the extent that I have, but thank you for being some of my most ardent supporters. You two have never been the limiting factor and I know that we’ll figure out a way to bring our plans to fruition.

From the University of Windsor, I want to say thank you to Bob Hodge, on top of letting me get away with not strictly adhering to protocol you’ve been a fellow Star Wars enthusiast and tech geek. I want to thank Nancy Barkley, Martha Hiuser, and Chris Côté, for doing so much behind the scenes that I know nothing about, but I know I benefited from.

I want to give a blanket shout-out to the many online communities that I’ve turned to at various times, particularly when times got tough. To my guildmates in Ataraxia (which still needs to be renamed “Hostile Work Environment”), particularly Sneakos the Rogue, team PG, and Gorlox (who is a people), I say Lok’tar Ogar. Bidet to my fellow Critters, is it Thursday yet? I also couldn’t have done this without the random Perl monks and Stack Overflow members who helped me fix my code, so I didn’t have to always harass Brad.

While my emotional support animals, Lucy and Chompers (aka: Monster-paws, Miss, or just Cleo), are very recent additions to my life, it would be remiss of me not to give them credit for the much-needed stress relief during the writing of this tome.

Lastly, to Dr. Mallory Wiper, I hope you enjoyed the period double-spacing. Fight me.
# TABLE OF CONTENTS

DECLARATION OF ORIGINALITY ................................................................. iii
ABSTRACT ...................................................................................................... iv
DEDICATION ............................................................................................... vi
ACKNOWLEDGEMENTS ............................................................................ vii
LIST OF TABLES ......................................................................................... xiii
LIST OF FIGURES ....................................................................................... xiv
LIST OF APPENDICES .............................................................................. xvi
LIST OF ABBREVIATIONS .......................................................................... xix

CHAPTER 1: INTRODUCTION ...................................................................... 1
  Genus Phaseolus .................................................................................. 2
  Field Genetics and Contemporary Breeding ..................................... 7
  Introgression as a Potential Tool for Crop Development ............... 13
  Objectives of this Work ................................................................. 15

CHAPTER 2: METHODOLOGY .................................................................. 16
  Biological Sample Selection & Preparation ................................. 17
  Computational Resources .......................................................... 18
  Genome Sequencing and Assembly ........................................... 19
  Genome Feature Annotation .................................................... 31
  Genome Visualization ............................................................... 35
  Data Refinement and Management .......................................... 36
  Assembly Strategy ........................................................................... 41
  Annotation ......................................................................................... 48
  Identification of Points of Introgression (POIs) ......................... 49

CHAPTER 3: RESULTS ........................................................................... 54
  Sequencing ......................................................................................... 55
  Assembly ............................................................................................. 59
  Annotation ............................................................................................. 83
  POI Identification .............................................................................. 86
  Data Co-Visualization ................................................................. 99

CHAPTER 4: DISCUSSION ..................................................................... 106
  Phaseolus Genome Structure and Organization ...................... 107
  Sequencing & Assembly – Merits of the Approach .................. 112
POI Identification.................................................................................................................. 116
POI Analysis .......................................................................................................................... 122
Implications of this Work ..................................................................................................... 126
Future Directions ................................................................................................................ 131
LITERATURE CITED ............................................................................................................. 134
APPENDICES ........................................................................................................................ 151
Appendix A – Software Tools Utilized .................................................................................. 152
Appendix B – Custom Code Generated for this Work ......................................................... 164
Appendix C – Translation Tables .......................................................................................... 171
Appendix D – Syntenic Alignments of G-19833 v1 and OAC-Rex
Pseudochromosomes v2 via ‘Mauve’..................................................................................... 174
Appendix E – Individual Scaffold_600 ‘RDP4’ Generated Graphs....................................... 177
Appendix F – Individual Scaffold_570 ‘RDP4’ Generated Graphs ....................................... 183
VITA AUCTORIS .................................................................................................................... 188
LIST OF TABLES

CHAPTER 3: RESULTS

Table 3.1 – OAC-Rex Raw Short-read Statistics......................................................... 55
Table 3.2 – OAC-Rex Trimmed Short-read Statistics .................................................. 56
Table 3.3 – OAC-Rex Raw Long-read Statistics ........................................................ 56
Table 3.4 – PI440795 Raw Read Statistics ................................................................. 57
Table 3.5 – PI440795 Trimmed Short-read Statistics .................................................. 58
Table 3.6 – OAC-Rex ‘Ray’ Statistics ......................................................................... 59
Table 3.7 – OAC-Rex Scaffold Statistics ..................................................................... 61
Table 3.8 – OAC-Rex Pseudochromosome Statistics .................................................. 66
Table 3.9 – Statistics of PI440795 Genome Assemblies .............................................. 72
Table 3.10 – Annotation Statistics of OAC-Rex Assemblies and PI440795
Pseudochromosomes................................................................................................. 84
LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Figure 1.1 – Common Dry Bean Varieties .......................................................... 3
Figure 1.2 – P. vulgaris Leaf Infected with CBB .............................................. 8
Figure 1.3 – Genetic Recombination as Initiated by a Double-strand Break ...... 11

CHAPTER 2: METHODOLOGY

Figure 2.1 – Pedigree of OAC-Rex .................................................................. 17
Figure 2.2 – Contig Generation Process ............................................................... 21
Figure 2.3 – Scaffold Assembly Process ............................................................... 29
Figure 2.4 – Screenshot of the ‘GBrowse’ Gene Browser ................................. 45
Figure 2.5 – G-19833 v1 Aligned to OAC-Rex Pseudochromosomes v2 via ‘Mauve’ .................................................................................................. 46
Figure 2.6 – ‘GBrowse_syn’ Screenshot ............................................................... 47
Figure 2.7 – Side by Side Comparison of ‘GBrowse’ and ‘JBrowse’ ................. 49
Figure 2.8 – Chromosome 8 Synteny Between G-19833 v1 and OAC-Rex Pseudochromosomes v2 via ‘Mauve’ ................................................................. 50
Figure 2.9 – Scaffold_600 Parentage, Filtered for 1 stdev .............................. 53

CHAPTER 3: RESULTS

Figure 3.1 – Syntenic Alignments of G-19833 v1 and OAC-Rex Pseudochromosomes v2 of Chr04 via ‘Mauve’ ................................................................. 77
Figure 3.2 – Syntenic Alignments involving PI440795 ‘Ray’ Generated Contigs via ‘Mauve’ .......................................................................................... 78
Figure 3.3 – Syntenic Alignments of OAC-Rex Pseudochromosomes v2 and OAC-Rex Pseudochromosomes v3.3 via ‘Mauve’ ................................. 79
Figure 3.4 – Syntenic Alignments of OAC-Rex Pseudochromosomes v3.3 and OAC-Rex Pseudochromosomes v4 via ‘Mauve’ ................................. 80
Figure 3.5 – Genome-wide Syntenic Alignments of Single-entry Compressed OAC-Rex Pseudochromosomes and G-19833 v2 via ‘SyMAP’ .............. 82
Figure 3.6 – ‘Mauve’ Alignment of Initial Points of Introgression in Chromosome 8 from OAC-Rex Pseudochromosomes v2 ........................................ 87
Figure 3.7 – Alignment of the Terminal 2 Mbp Sequence Ranges of Chromosome 8 in OAC-Rex Pseudochromosomes v4 via ‘RDP4’ .......................................................... 88
Figure 3.8 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Parentage ...................... 90
Figure 3.9 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Parentage ...................... 91
Figure 3.10 – OAC-Rex ‘AllPaths-LG’ Scaffold_1971 Parentage ..................... 92
Figure 3.11 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signals Generated via ‘RDP4’ ........................................................................................................... 93
Figure 3.12 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signals Generated via ‘RDP4’ ........................................................................................................... 96
Figure 3.13 – JBrowse Screenshots of Chr04:383593–1396784 from OAC-Rex Pseudochromosomes v4 ................................................................. 100
Figure 3.14 – JBrowse Screenshots of Chr08:62819995-63166454 from OAC-Rex Pseudochromosomes v4 ................................................................. 103
Figure 3.15 – JBrowse Screenshots of Chr08:61349612-63304788 from OAC-Rex Pseudochromosomes v4 ................................................................. 104
LIST OF APPENDICES

Appendix A – Software Tools Utilized
   Table A.1 – Software Tools Used ................................................................. 152

Appendix B – Custom Code Generated for this Work
   Table B.1 – Custom Code Generated for this Work .................................... 164

Appendix C – Translation Tables
   Table C.1 – IUPAC Nucleotide Code and Meaning ..................................... 171
   Figure C.1 – ASCII Table ............................................................................. 172
   Figure C.2 – Canonical Codon Translation Table ......................................... 173

Appendix D – Syntenic Alignments of G-19833 v1 and OAC-Rex
   Pseudochromosomes v2 via 'Mauve'
      Figure D.1 – Chromosome 01 ..................................................................... 174
      Figure D.2 – Chromosome 02 ..................................................................... 174
      Figure D.3 – Chromosome 03 ..................................................................... 174
      Figure D.4 – Chromosome 05 ..................................................................... 175
      Figure D.5 – Chromosome 06 ..................................................................... 175
      Figure D.6 – Chromosome 07 ..................................................................... 175
      Figure D.7 – Chromosome 09 ..................................................................... 175
      Figure D.8 – Chromosome 10 ..................................................................... 176
      Figure D.9 – Chromosome 11 ..................................................................... 176

Appendix E – Individual Scaffold_600 ‘RDP4’ Generated Graphs
   Figure E.1 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from
      ‘3Seq’ via ‘RDP4’ ......................................................................................... 177
   Figure E.2 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from
      ‘Bootscan’ via ‘RDP4’ .................................................................................. 177
   Figure E.3 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from
      ‘BURT’ via ‘RDP4’ ....................................................................................... 178
   Figure E.4 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from
      ‘Chimaera’ via ‘RDP4’ .................................................................................. 178
Figure E.5 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Distance Plot via ‘RDP4’ ................................................................. 178
Figure E.6 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘GENECONV’ via ‘RDP4’ ................................................................. 179
Figure E.7 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘MAXChi’ via ‘RDP4’ ................................................................. 179
Figure E.8 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘PhylPro’ via ‘RDP4’ ................................................................. 180
Figure E.9 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from the RDP method via ‘RDP4’ ................................................................. 180
Figure E.10 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘SiScan’ via ‘RDP4’ ................................................................. 181
Figure E.11 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘TOPAL’ via ‘RDP4’ ................................................................. 181
Figure E.12 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘VisRD’ via ‘RDP4’ ................................................................. 182
Appendix F – Individual Scaffold_570 'RDP4' Generated Graphs

Figure F.1 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘3Seq’ via ‘RDP4’ ................................................................. 183
Figure F.2 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘Bootscan’ via ‘RDP4’ ................................................................. 183
Figure F.3 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘BURT’ via ‘RDP4’ ................................................................. 184
Figure F.4 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘Chimaera’ via ‘RDP4’ ................................................................. 184
Figure F.5 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Distance Plot via ‘RDP4’ ................................................................. 184
Figure F.6 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘GENECONV’ via ‘RDP4’ ................................................................. 185
Figure F.7 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘MAXChi’ via ‘RDP4’ ................................................................. 185
Figure F.8 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘PhylPro’ via ‘RDP4’ ................................................................................................................. 186

Figure F.9 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from the RDP method via ‘RDP4’ ................................................................................................................. 186

Figure F.10 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘SiScan’ via ‘RDP4’ ................................................................................................................. 187

Figure F.11 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘TOPAL’ via ‘RDP4’ ................................................................................................................. 187

Figure F.12 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘VisRD’ via ‘RDP4’ ................................................................................................................. 187
LIST OF ABBREVIATIONS

%N Percent indeterminate nucleotide
A Adenine
AA Amino acid
APE Artificial paired-end short-read
ASCII American standard code for information interchange
BLAST Basic local alignment search tool
bp Base pair
C Cytosine
CBB Common bacterial blight
cDNA Complementary deoxyribonucleic acid
CEG Common eukaryotic gene
CGI Common gateway interface
CPU Central processing unit
DDR3 Double data rate type 3
DNA Deoxyribonucleic acid
DSB Double-strand break
DSBR Double-strand break repair
G Guanine
GB Gigabyte
Gbp Billion base pairs
gDNA Genomic deoxyribonucleic acid
GHz Gigahertz
GMOD Generic model organism database
GPU Graphical processing unit
HDD Hard disc drive
HRR Homologous recombinational repair
in/del Insertion or deletion
iSCSI Internet small computer systems interface
IUPAC International Union of Pure and Applied Chemistry
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>kbp</td>
<td>Thousand base pairs</td>
</tr>
<tr>
<td>LTR</td>
<td>Long terminal repeat</td>
</tr>
<tr>
<td>MHz</td>
<td>Megahertz</td>
</tr>
<tr>
<td>Mbp</td>
<td>Million base pairs</td>
</tr>
<tr>
<td>mRNA</td>
<td>Messenger ribonucleic acid</td>
</tr>
<tr>
<td>MSA</td>
<td>Multi-sequence alignment</td>
</tr>
<tr>
<td>N</td>
<td>Indeterminate nucleotide</td>
</tr>
<tr>
<td>NCBI</td>
<td>National Center for Biotechnology Information</td>
</tr>
<tr>
<td>ncRNA</td>
<td>Non-coding ribonucleic acid</td>
</tr>
<tr>
<td>NFS</td>
<td>Network file system</td>
</tr>
<tr>
<td>NHEJ</td>
<td>Non-homologous end joining</td>
</tr>
<tr>
<td>OS</td>
<td>Operating system</td>
</tr>
<tr>
<td>PB</td>
<td>Petabyte</td>
</tr>
<tr>
<td>piRNA</td>
<td>Piwi ribonucleic acid</td>
</tr>
<tr>
<td>PE</td>
<td>Paired-end</td>
</tr>
<tr>
<td>POI</td>
<td>Point of Introgression</td>
</tr>
<tr>
<td>POSIX</td>
<td>Portable operating system interface</td>
</tr>
<tr>
<td>QTL</td>
<td>Quantitative trait locus</td>
</tr>
<tr>
<td>RAID</td>
<td>Redundant Array of Independent Disks</td>
</tr>
<tr>
<td>RAM</td>
<td>Random-access memory</td>
</tr>
<tr>
<td>RNA</td>
<td>Ribonucleic acid</td>
</tr>
<tr>
<td>RNAseq</td>
<td>Ribonucleic acid sequencing</td>
</tr>
<tr>
<td>rRNA</td>
<td>Ribosomal ribonucleic acid</td>
</tr>
<tr>
<td>SDSA</td>
<td>Synthesis-dependent strand annealing</td>
</tr>
<tr>
<td>SMRT</td>
<td>Single molecule, real-time</td>
</tr>
<tr>
<td>SNP</td>
<td>Single nucleotide polymorphism</td>
</tr>
<tr>
<td>T</td>
<td>Thymine</td>
</tr>
<tr>
<td>TB</td>
<td>Terabyte</td>
</tr>
<tr>
<td>U</td>
<td>Uracil</td>
</tr>
<tr>
<td>TE</td>
<td>Transposable element</td>
</tr>
<tr>
<td>tRNA</td>
<td>Transfer ribonucleic acid</td>
</tr>
</tbody>
</table>
CHAPTER 1: INTRODUCTION
Genus Phaseolus

A collection of approximately 70 angiosperms, the genus *Phaseolus* is native to the Americas and part of the legume family Fabaceae (Delgado-Salinas, Turley, Richman, & Lavin, 1999). The family is well known for its symbiotic relationship with Rhizobia, a collection of bacteria that affix atmospheric nitrogen (known as “diazotrophs”) after infecting the roots to create nodules. This fixated nitrogen is used by the plant for healthy growth and to increase yield, while the bacteria primarily benefit from malate or succinate as a carbon and energy source provided by the plant (M. Andrews & Andrews, 2017). Much of the fixed nitrogen is often kept in the soil as nitrates since many members of the *Phaseolus* genus are annuals and the root nodules can be tilled into the soil before the next round of planting. This added nitrogen can be used by the next rotation of crops which reduces, and potentially eliminates the need for nitrogen fertilizer (Herridge, 2013).

There are five species within the genus that have been domesticated; *lunatus* (lima bean), *coccineus* (runner bean), *dumosus* (year bean), *acutifolius* (tepary bean) and *vulgaris* (common bean). Of these five, the most prevalent is unquestioningly *Phaseolus vulgaris* (Bitocchi et al., 2017).

*P. vulgaris*

The common bean (*P. vulgaris*) is an herbaceous annual grown around the world. There are over 130 cultivars, artificially generated lineages from selective breeding (essentially the plant equivalent of breed), many of which are readily recognizable, including navy bean, kidney bean, string bean, white bean, pinto bean, and green bean, to list a few (Bitocchi et al., 2017; Facciola, 1998). Along with squash and maize, it made up the ‘three sisters’ central to agriculture of many native American groups. The plant itself grows in two forms, the more common being bush (known as “dwarf”) beans which can grow without a support structure, as opposed to pole (known as “climbing” or “runner”) beans. This structure can be other plants, a synergy provided by maize in early agriculture, or an artificial support such as a trellis (du Plessis, Fourie, Liebenberg, Liebenberg, & van Zyl, 2009; Freytag & debouck, 2002).
In either case, the green or purple trifoliate leaves broaden sharply from the stalk, tapering quickly to a somewhat protracted point with smooth edges reminiscent of a spade from a deck of cards. The flowers of *P. vulgaris* can be purple, white, or pink and are irregular and papilionaceous, or butterfly-like, with a large petal called the standard, a wing petal on either side thereof, and a keel made of two partially or completely join petals terminating in a coil. Typically, the flower has ten stamens, nine of which form a tube around the ovary and the last of which is separate and sits above the ovary. The fruit of the plant grows in green, purple, yellow, or black pods containing up to 12 seeds, both seeds and pods are commonly referred to as beans. These seeds are smooth and kidney shaped with an array of colours, often mottled combinations as shown in Figure 1.1, below (OECD, 2016; Purseglove, 1977).

![Common Dry Bean Varieties](http://blog.generationcp.org/wp-content/uploads/2012/10/Beans4_web.jpg)

**Figure 1.1 – Common Dry Bean Varieties**

_Demonstrating the varied forms of mature *P. vulgaris* seeds, retrieved from http://blog.generationcp.org/wp-content/uploads/2012/10/Beans4_web.jpg_

**Geography**

Common bean originated in a sub-tropical to temperate climate of Mesoamerica (Gepts & Debouck, 1991). First identified in the wild in Guatemala in the mid 20th century, wild *P. vulgaris* plants have been found in locations ranging from northern Mexico to northern Argentina (Acosta-Gallegos, Kelly, & Gepts, 2007; McBryde, 1945). This distribution is interrupted, however, by pockets of unfavourable conditions. *P. vulgaris* is known to prefer temperate regions since it is not frost tolerant and pod production can be delayed by low temperatures. This poor performance in cold climates
limits the growing range to altitudes below 3 km. Extreme heat is also detrimental, causing flowers to abscise (Beebe, Rao, Devi, & Polania, 2014; du Plessis et al., 2009).

The wild variants of *P. vulgaris* are all climbing vines, a needed feature since the light reaching the forest floor is limited by taller vegetation in its wild habitat. It is also an opportunistic invader, taking advantage of disturbances in ecological niches to establish itself. The taller flora act as the support structure for wild common bean allowing it to compete for sunlight. Both excessive rainfall and drought are detrimental to *P. vulgaris* growth, as it prefers well-drained sandy loam or clay (du Plessis et al., 2009).

**Domestication**

Domestication of *P. vulgaris* involved a complicated series of events: at one point believed to have come from Asia, *P. vulgaris* has since been traced back to central America. Some records indicate that it was domesticated by 5000 B.C., though genetic studies have revealed that domestication may have begun as early as 10,000 years ago. Further complicating our understanding is the apparent dual domestications of *P. vulgaris*, one in Mesoamerica, and one amongst the Andes, for which the respective landraces have been named (Cichy et al., 2015; Gepts & Debouck, 1991).

These two gene pools have demonstrated a partial reproductive isolation from each other that is apparent in both wild and domesticated plants, which generates a unique situation for a crop species. This isolation has allowed the generation of lineages that have independent evolutionary histories. Of note are four genetic groupings within the Mesoamerican pool that confer a rather broad diversity (Bitocchi et al., 2017).

This greater diversity led to the proposal that the Mesoamerica pool was the origin of *P. vulgaris*, which then extended south. This hypothesis was supported by the greater relatedness of wild *P. vulgaris* to those domesticated in Mesoamerica (Delgado-Salinas, Bibler, & Lavin, 2006). However, a third pool of wild bean has also been identified in areas between Peru and Ecuador showing distinct genetic characteristics associated to the phaseolin type I protein. In fact, Colombian wild populations appear to be intermediates with the Mesoamerican landraces, leading to the hypothesis that this third pool is ancestral and that it then spread north and south to generate the other two pools (Bitocchi et al., 2012).
Lastly, there was a severe genetic bottleneck identified as having occurred in the Andean pool before domestication (Cichy et al., 2015). All these confounding factors were eventually clarified to determine that the origin of *P. vulgaris* is indeed Mesoamerica, the four subgroups spread south into the Andes where they split to form the Andean and the Peru-Ecuadorian pools before some selective force caused the bottlenecking in the Andean pool after which the separate domestications occurred on the Mesoamerican and Andean pools (Bitocchi et al., 2017).

The bottleneck in the Andean pool may have had a dissuading effect on the use of Andean land races in current breeding efforts. The broader genetic diversity available to Mesoamerican landraces has been posited as a potential reason for their recent breeding gains. Overall, domestication has generally made the leaf, pod, and seed size larger, as well as being responsible for the wide range of seed colouration. It has also reduced seed dormancy, pod dehiscence, and led to a reduced genetic diversity when compared to wild relatives (Cichy et al., 2015).

**G-19833**

An inbred line of an Andean landrace, G-19833 was the first completed genome assembled for *P. vulgaris*. Before its sequencing, G-19833 was characterized for the purposes of developing a dataset against which the Mesoamerican BAT93 genome could be compared, BAT93 having been used for several sequencing efforts. The assembled sequence and annotations for the G-19833 cultivar were made publicly available and its more mature state compared to that of BAT93 made it a prime candidate to represent *P. vulgaris* in this study (Altrock, Fonsêca, & Pedrosa-Harand, 2011; Fonseca et al., 2010; Schmutz et al., 2014).

**P. acutifolius**

Commonly called the tepary bean, *P. acutifolius* can be climbing, trailing, or dwarf. Able to grow up to four meters long, though averaging two, the plant is characterized by its narrow, pointed leaves. The flowers are white or light coloured and like *P. vulgaris* the seeds come in a wide array of colours, maturing 60–120 days after planting, depending on the cultivar. Tepary beans have also been grown for thousands of
years by native American groups in Mexico and the southern United States (Freytag & debouck, 2002; Wolf, 2018).

**Pathogen Resistance Traits**

A key trait of the tepary bean, particularly in a research and/or breeding context, is its tolerance to harsher conditions and its resistance to a wide array of pathogens. Able to withstand temperatures from 8–38°C, as well as a dry atmosphere, *P. acutifolius* has been recorded surviving excessive rainfall by sacrificing yield for vegetative growth. It is able to grow in shallow, poor quality soil and demonstrates a notable drought tolerance growing in regions with as little as 400 mm of annual rainfall (Wolf, 2018; Mukund Zambre, Montagu, Angenon, & Terryn, 2006). Beyond the broader climate range in which *P. acutifolius* can grow, it has also been shown to be resistant to many microbial pathogens, such as *Acanthoscelides obtectus, Xanthomonas axonopodis, Fusarium oxysporum*, or *Zabrotes subfasciatus* (Drijfhout & Blok, 1987; Jiménez, de la Fuente, Ordás, García Domínguez, & Malvar, 2017; Schinkel & Gepts, 1988; M. Zambre et al., 2005).

**PI440795**

The cultivar PI440795 of *P. acutifolius* arises in literature with some regularity due to its resistance to common bacterial blight (CBB) - a *X. axonopodis*-associated disease described later in the chapter (Smith, Michaels, Navabi, & Pauls, 2012). This cultivar played a key role in the development of OAC-Rex, acting as the source for its CBB resistance trait and, as such, was selected as the ideal candidate to act as a representative of *P. acutifolius* in this project.

**Cultivation**

Common bean is perhaps the most cultivated legume for direct human consumption. The entire plant is used to meet various agricultural needs. The dried seed and unripe fruit have been discussed above, but the straw is a source of fodder and the leaf is safe for human and livestock consumption as well. Grown around the world, it provides an excellent source for protein, vitamins, minerals and fiber and regular consumption has been linked to health benefits. Although the more commonly cultivated landraces tend to be of Mesoamerican origin, Andean landraces have seeds of 25–40 g
per 100 seeds (classified medium) or >40 g per 100 seeds (classified as large), while Mesoamerican cultivars are small (<25 g per 100 seeds, Bitocchi et al., 2012; Gepts & Debouck, 1991; Schmutz et al., 2014).

This variance in seed size may be the reason for the varied predominance in cultivation of one landrace over the other. In parts of Africa, Europe, and South America, the Andean landraces are more common, while Mesoamerican landraces are far more commonly grown in North and Central America. Developed nations such as Canada often grow large fields of bean, supplementing soil with Rhizobia inoculations, fertilizers, and adjusting water as needed for optimal productivity. In less developed nations Phaseolus cultivation generally occurs at a smaller scale in nitrogen- or phosphorous-poor soils (Cichy et al., 2015).

Bean production is grouped into three categories based on their end-use: dry beans, as the fully matured seeds; snap beans, as the pods harvested before seed development; and shell beans, which are harvested when physiologically matured (Freytag & debouck, 2002). In 2017, it was estimated that worldwide 36 million hectares were used to grow dry beans, yielding a total of 31 million tonnes, while 1.5 million hectares were used to grow 24 million tonnes of green beans. The largest producer of green beans by tonnes produced was China at 19 million tonnes (~80%) with Indonesia in second at just under 1 million tonnes (~3.8%). For dry beans India was the largest producer at 6 million tonnes (~20%) with Myanmar immediately behind them at 5 million tonnes (~17%). In the same year Canada produced 256,835 tonnes of dry bean (~0.82%) across 116,390 hectares to be 20th on the list and 53,950 tonnes of green bean (~0.22%) on 8,696 hectares making them 18th overall ("FAOSTAT," 2019).

**Field Genetics and Contemporary Breeding**

Human artificial selection has been an important factor in the evolution of crop species for approximately 10,000 years. The domestication and cultivation or organisms to make them better suited to human needs is probably no more readily apparent than it is in crops. A comparison of wild flora to their domesticated counterparts reveals a startling change in morphology, chemical composition, flavour, and nutritional value. This process
has largely been performed with no understanding of the underlying genetics being manipulated, instead focusing solely on phenotypic selection, due to a lack of knowledge and technological inability to properly investigate (Alberts et al., 2008; Kasha, 1999).

In the last 70 years molecular sciences have made major breakthroughs in understanding what genes are, how they function, and their interactions with each other and the environment, a field known as genomics. Genomic approaches have been used to refine the initial phenotype-based system converting it to a genotypic driven one. A prime example of this is marker assisted selection, the use of specific molecules or sequence fingerprints to identify a genetic element associated to a desired phenotype (Kasha, 1999).

**Common Bacterial Blight (CBB)**

Common bacterial blight is a disease caused by *Xanthomonas axonopodis* pv. *phaseoli* that affects several species of Phaseolus. Endemic to Southern Ontario, it can impair yield by 20–40%. Identifiable first as small water-soaked spots on the underside of leaflets which grow and combine, eventually becoming brown and dry. These brown spots are almost always edged by a bright yellow ring as shown in Figure 1.2. Infected pods develop the same spots and the bacteria may collect in the centre as a large yellow clump. These pod spots will eventually become a dark brown divot in the pod.

![Figure 1.2 – P. vulgaris Leaf Infected with CBB](http://www.omafra.gov.on.ca/english/crops/field/news/croptalk/2015/ct-0315a6.htm)

If the pod is infected before seed development the pods will form in a shriveled state, showing telltale yellowish spots under the seed coat. If these seeds are subsequently grown the spots will be manifest above the cotyledonary nodes of the progeny seedling (American Phytopathological Society, 2010; Boersma, Hou, Gillard, McRae, & Conner, 2015).

*Xanthomonas axonopodis pv. phaseoli*

The bacteria associated with CBB infect plants via natural openings such as stomata or wounds after being deposited on leaves via water, either through splashing or aerosolization. It can also be transmitted through the germination of infected seed in proximity of healthy plants. Infection is most likely during warm wet weather, particularly during storms with strong winds that facilitate the aerosolization of bacteria-containing droplets. Contaminated workers and equipment also serve as transmission vectors on larger farms. Between growing seasons, the bacteria survives using weeds as a host, on crop debris, and in contaminated seed (Boersma et al., 2015).

**Management**

Specific practices can be employed to limit the spread of *X. axonopodis*; these practices include avoiding overhead irrigation, preventing reuse of irrigation water, using resistant varieties, avoiding working in fields when plants are wet, or rotating crops to non-host plants such as grains (Akhavan et al., 2013). Chemical controls, for example the use of copper bactericides, are not generally effective though they may offer some assistance in preventing spread (Frate, Gepts, & Long, 2018; Hall, 2015).

**Introgression**

Sometimes called introgressive hybridization, introgression is defined as the repeated backcrossing of an interspecific hybrid resulting in a net gene flow from one species into the gene pool of another. This process differs from simple hybridization in one key point: hybrid offspring carry a uniform mixture of gene content from both parents whereas an introgression clearly favours one parent (Peter R. Grant, B. Rosemary Grant, & K. Petren, 2005). The backcrossing of the uniform offspring causes this preference and can be a source for inflow of genetic variation since,
interestingly, when an introgression occurs it is not a speciation event. Instead the introgressed offspring is definitively of the same species as the ‘dominant’ parent (Dowling & Secor, 1997; Harrison & Larson, 2014). As such, introgression has been suggested as a possible driver for the rapid diversification of organisms from an ancestral species known as adaptive radiation, and appears to be ubiquitous across eukaryota. It has been associated with hybrid zones - places where interbreeding species meet - and has been proposed as a mechanism exploited by invasive species (Buggs, 2007; Hata, Uemura, Ouchi, & Matsuba, 2019).

Introgression has long been exploited by crop breeders, often without an understanding of the biological events occurring in the crops and despite the potentially far reaching implications of the process, the underlying genetic events and the biological processes driving them are relatively poorly understood (Eshed & Zamir, 1995). Before the discovery of the lonicera fly, a cross between *Rhagoletis mendax* and *Rhagoletis zephyria*, it was believed to only occur naturally in plants and it is almost never identified until many generations after the initial hybrid backcrossing (Schwarz, Matta, Shakir-Botteri, & McPheron, 2005). Introgressions can be generated by forced crossings, a process far more common in plants since animal mating behaviours add greater complexity on top of the genetic barriers that may already exist. A likely reason for this lack of understanding is the commonplace occurrence of hybrid sterility or hybrid breakdown, two conditions that describe the general inability of interspecific hybrids to produce viable offspring (Ehrman, 1962; Z. Li, Pinson, Paterson, Park, & Stansel, 1997).

Inter-specific sterility is believed to be primarily caused by genomic incompatibility of the hybrid to either parent. Mules famously have 63 chromosomes, one more than a donkey and one fewer than a horse (Henry, Gastal, Pinheiro, & Guimarmes, 2018); however, the mechanism(s) behind coping with the mass influx of new genomic material when the hybrid is fertile remains unknown.

**Genomics**

*Recombination*

Since an introgressed organism is viable — albeit with a host of potentially novel genes — some form of inter-recombination must be occurring. Genetic recombination is
the rearrangement of genetic information between organisms leading to a shuffling of traits in offspring relative to its parents. Occurring during meiosis or mitosis, this can lead to gene conversion or chromosomal crossover. Recombination can also occur during DNA repair, for example non-homologous end joining (NHEJ) or homologous recombinational repair (HRR) (Alberts et al., 2008).

**Homologous Recombinational Repair**

![Diagram of genetic recombination initiated by a double-strand break](image)

*Figure 1.3 – Genetic Recombination as Initiated by a Double-strand Break*
The two models of homologous recombination leading to gene conversion, specifically in response to a double-strand break. The final result may end in crossover, a likely, though not guaranteed, outcome from double-strand break repair (DSBR) pathway (Sung & Klein, 2006).

The general process of homologous recombination repair is illustrated in Figure 1.3. In response to a double-strand break there will be a resection of 5’ ends of both strands to generate single-strand overhangs. One of the 3’ overhangs will take part in a strand invasion of a homologous stretch of the genome, whereby it will dissociate the homologous sequence to bind with the complimentary strand of the donor region generating a “D-loop”. DNA polymerase will then extend the invading 3’ strand at which point the possible outcome will follow one of two pathways; synthesis-dependent strand annealing (SDSA) or double-strand break repair (DSBR, sometimes called the double Holliday junction model). If the donor strand is of a different allele then the process will result in replacement of the allele in the original strand with that of the donor strand in a process known as gene conversion (Alberts et al., 2008; Bernstein, Bernstein, & Michod, 2011).

**SDSA**

SDSA is a relatively simple pathway and one that does not result in crossover i.e. where the strands have not undergone exchange between the donor strands and the initial, broken strands. Here the Halliday junction between the donor and invading strands moves towards the end of the now-extended invading strand, a process called branch migration, resulting in the release of the invading strand. The released strand is long enough to overlap with, and complimentary to, the remaining 3’ overhang from the initial resection. These overhangs can anneal, any over-extension can be removed, and DNA polymerase can synthesize any remaining single-strand regions resulting in a contiguous double stranded DNA molecule with a short segment from the homologous donor region (Alberts et al., 2008; Helleday, Lo, van Gent, & Engelward, 2007).

**DSBR**

In this model the 3’ overhang that had not taken part of the initial strand invasion binds to the displaced donor strand in the D-loop forming a second Halliday junction as it too is extended. Each of these Halliday junctions are then resolved by a nicking endonuclease cutting one strand involved in the junction with a crossover likely to occur
though it is not guaranteed and depends on how the junctions are resolved. If one junction cuts the crossing strand (the purple arrows in Figure 1.3) and the other cuts the non-crossing strand (the orange arrows in Figure 1.3), the result will be a crossover and conferring a non-parental di-type to the progeny (Alberts et al., 2008; Lehninger, 2005; Sung & Klein, 2006).

**NHEJ**

Another option to repair DSB is NHEJ wherein the repair pathway attempts to correct the DSB by directly ligating the two ends together. During a DSB there are sometimes single stranded overhangs; should these overhangs exhibit homology to each other it is likely that NHEJ will accurately repair the break. However, if the overhangs are missing or do not have homology to each other the repair process will excise sequences from one or both strands until a partial match is generated, at which point the repair process will proceed as above. NHEJ is best characterized for its involvement in the process of V(D)J recombination in vertebrate immune systems (Budman & Chu, 2005; Moore & Haber, 1996).

**Introgression as a Potential Tool for Crop Development**

A lack of understanding of genomics implies an incomplete understanding of species and speciation, as evidenced by the large number of taxonomic reclassifications that have occurred since the development of gene sequencing tools (Schwenter, Timms, & Richter, 2011). This is understandable since shared morphology has been a primary element in categorizing species that can sometimes be misleading, and is further complicated in crop breeding by the apparent ability of plants to cross species boundaries more easily than animals. These instances, where introgression mechanisms and requirements at a genomic level is forced by artificial human selection based solely on morphological characteristics, can generate crop lines of complex unknown genomic composition. A molecular understanding of introgression would highlight where and how these phenotypes are crossing species boundaries. This would serve, at the very least, to identify a marker set to better select subjects for introgression-based breeding, in turn
improving the success rate of cultivar generation with desired inter-specific traits and minimizing potential side-effects.

**Phaseolus Development in Canada**

There are several research groups investigating how to develop new bean varieties, along with other pulses, better suited to agricultural conditions across Canada such as the Ontario Bean Growers, Saskatchewan Pulse Growers, Agriculture and Agri-food Canada, or the Ontario Agricultural College at the University of Guelph. These organizations employ a range of experimental approaches to study *Phaseolus* with the aim to understand the impacts of dietary inclusion on human health or to improving the nutritional value, appeal to consumers, shelf life, and even flavour. They also look to increase yield by reducing loss to pathogens, weeds, or pests, to generate cultivars with improved tolerance to drought or salinity. This work is done as a close partnership between scientists, farmers, economists, and marketers at universities and research institutes across the country. In Ontario, a collaborative effort between the University of Windsor, the University of Guelph, Agriculture and Agri-Food Canada, Ontario Bean Growers, Hensall District Co-operative, Agriculture Adaptation Council, Coloured Bean Growers, and University of Ottawa has focussed on developing and studying OAC-Rex since 2006 as part of the Applied Bean Genomics and BioProducts project.

**OAC-Rex**

OAC-Rex is a *P. vulgaris* cultivar with a complex pedigree that begins with a known introgression with *P. acutifolius* and includes backcrossing with several *P. vulgaris* cultivars. Developed in the late 1980s, it was the first *P. vulgaris* variety with traits desirable for yield, body morphology, seed size, life history, and importantly, a resistance to CBB. This resistance trait has made it a prime cultivar for further breeding, which has led to the development of other CBB resistant lines. OAC-Rex has also been studied to identify pathways involved in seed darkening, the impact of a bean-based diet on gut microbiome, the genetic basis of CBB resistance, the impact of Anthracnose infection and how to treat it, as well as means to improve yield by molecular marker-
based selection (Freixas Coutin et al., 2017; Michaels, Smith, Larsen, Beattie, & Pauls, 2006; Perry et al., 2013).

Objectives of this Work

The work embodied in this thesis had four main objectives at its outset:

1. Assemble and annotate the ‘OAC-Rex’, *P. acutifolius* (PI440795), and *P. vulgaris* (G-19833) genomes;
2. Characterize points of introgression in ‘OAC-Rex’;
3. Verify annotations and characterizations via known introgression loci;
4. Identify traces of genomic rearrangement and/or expression regulation at/within points of introgression;

These objectives were modified during the course of the investigation to remove the assembly and annotation of G-19833 as this was completed and published to Phytozone (https://phytozome.jgi.doe.gov/pz/portal.html, Schmutz et al., 2014).
CHAPTER 2: METHODOLOGY
Biological Sample Selection & Preparation

OAC-Rex is the result between the crossing of two crossbred lines, HR20-728 and MBE 7, that in turn were derived from a total of four base lines: three *P. vulgaris* cultivars (ICA Pijao, Ex Rico 23, and Midnight), as well as the *P. acutifolius* cultivar PI440795 (Figure 2.1). HR20-728 is a cross of Midnight and Ex Rico 23, while MBE 7 is a cross between Ex Rico 23 and the offspring of PI440795 and ICA Pijao. PI440795 was specifically chosen for its resistance to CBB, while each of the *P. vulgaris* lines were selected for their agricultural traits. OAC-Rex was grown to an F10 generation between Southern Ontario and New Zealand from 1988–1994, before the plants were submitted to variety registration trials. F9 plants were subsequently used for breeder seed production in Twin Falls, Idaho. (Michaels et al., 2006).

![Figure 2.1 – Pedigree of OAC-Rex](image)

*Figure 2.1 – Pedigree of OAC-Rex*

*A summary of the repeated crossing and selection process that conferred CBB resistance to OAC-Rex (Pauls, K.P., Personal Communication).*

From breeder seed stock, 10 g of young leaves were harvested from three- to four-week-old plants and used to extract genomic DNA. Leaf tissue was frozen in liquid nitrogen, ground to a powder, and resuspended in Homogenization Buffer (Sigma, St.
Louis MO). The solution was filtered through porous cloth before several rounds of centrifugation and washing to derive a fraction-enriched in nuclei. DNA was extracted using a Qiagen® DNeasy Plant Kit and subsequently sequenced at the Center for Applied Genomics (Perry et al., 2013, TCAG; SickKids Hospital, Toronto, ON).

**Computational Resources**

Large scale computation analyses performed in support of this project were distributed across multiple resources based in Southern Ontario and Quebec in partnership with collaborators for this and other projects performed concurrently. Of note are 3 computation centers.

Firstly, Dr. Thomas Bureau at McGill University, Montreal generously provided access to his Mustang server, a Dell® PowerEdge® R910 with 1 x Intel® Xeon® E7-4850 @ 2.00 GHz, 32 x 8 GB 1066 MHz DDR3 RAM, and 1 x 16 TB RAID5 array, and running CentOS® 5.

In Windsor, 3 desktop workstations were used incorporating the following hardware and O/S resources. **Banting**: 1 x Intel® Core® i7-920K @ 3.20 GHz, 4 x 4 GB 1333 MHz DDR3 RAM, 4 x 8 GB 1333 MHz DDR3 RAM, 100 GB swap space, 1 x 1 TB 7200 rpm dedicated OS HDD, 4 x 3 TB 7200 rpm HDD in RAID5 array, and running Ubuntu® 12.04 until long term support was discontinued, at which point the system was upgraded to 14.04. **McCIntock**: 1 x Intel® Core® i7-950 @ 3.07 GHz, 6 x 4 GB 1066 MHz DDR3 RAM, 100 GB swap space, 1 x 1 TB 7200 rpm dedicated OS/swap HDD, 4 x 3 TB 7200 rpm HDD in a RAID5 array, and running Ubuntu® 14.04. **Einstein** (Synology® DS1812+ NAS): 1 x Intel® Atom® D2700 @ 2.13 GHz, 1 x 1 GB 1066 MHz DDR3 RAM, 8 x 3 TB 7200 rpm HDD in RAID5 array.

Finally, the ‘Beanblade’ 4-node cluster belonging to Dr. K. Peter Pauls at the University of Guelph is a Supermicro™ SuperServer 6028TP-HC1R – 2U TwinPro2 rack server with 8 x Intel® Xeon® E5-2650L v3 @ 1.8 GHz (2/node), 32 x 32 GB 2133 MHz DDR4 RAM (512 GB on node 1, 192 GB on node 2, and 160 GB on nodes 3 & 4), 100 GB swap space per node, 4 x 500 GB 7200 rpm OS dedicated HDD (1/node), 4 x 3 TB 7200 rpm HDD (1/node) in a RAID5 array, and running Ubuntu® 16.04. Of note on
Beanblade is the distributed nature of the RAID5 array: each HDD is on a separate node that is shared via iSCSI to the master node, where the RAID is assembled and then shared as an NFS back to each of the slave nodes.

Canadian computational researchers also regularly make use of the shared resource centers that are combined under the auspices of Compute Canada, notably manifest in Ontario as the Shared Hierarchical Academic Research Computing Network (SHARCNet®). Boasting 35000 CPUs, 320 GPUs, up to 3 TB of RAM on specific nodes and over 3 PB of storage space, it is a powerful resource available via membership for academic research. The resources administered by SHARCNet® were largely ones that were not used during this project. Upon acquisition of Beanblade, benchmarking tests of common assembly pipelines that were to be extensively used were simultaneously run on both platforms. The resources requested from SHARCNet® matched the full capabilities of the entire Beanblade cluster where identical data was loaded to SHARCNet® and Beanblade, with the identical command being provided to the same version of the ‘Ray’ contig assembly software. Beanblade completed its run, had the data downloaded, examined for structural statistics, the assembly command altered, rerun, the new run downloaded, examined, and a third run begun before SHARCNet® completed its initial run of the data. While requesting more resources from SHARCNet® was possible, estimations for wait times increase as more resources are requested and the net impact on processing efficiency is a negative one: while double the resources would ordinarily complete a given computational task in half the time, the wait time to procure access to the relevant resources were more than the time saved. Given the significant computational power of Beanblade, combined with the ability to rapidly administer any application-specific requirements, this platform constituted the principal resource in support of the project.

Genome Sequencing and Assembly

Investigating genetic introgression at the resolution necessary to draw conclusions regarding molecular mechanisms required mature genome sequence assemblies and annotations for the introgressed offspring and both input parents. OAC-Rex was an ideal
candidate with which to study the outcome of genetic outcrossing in this important agricultural species give its well-defined pedigree involving the introgression of *P. vulgaris* and *P. acutifolius* input lines. To that end, the genomes of OAC-Rex, PI440795, and a *P. vulgaris* reference were needed. The techniques and technologies available, and subsequently used varied between each assembly spanning the lifetime of the project. Early in our efforts to assemble a high-quality genome sequence for OAC-Rex, a genome for a *P. vulgaris* Andean reference line, G-19833, was released by others (Schmutz et al., 2014). Following successful assembly of OAC-Rex, the PI440795 genome was similarly assembled as part of the Canadian project.

**Comparative Merits of Sequencing Platforms**

Multiple DNA sequencing platforms and procedures exist, each with their own specific benefits, limitations, and costs. The primary distinction between these platforms is the length of the reads generated, generally classified into ‘short read’ and ‘long read’ technologies (Straiton, Free, Sawyer, & Martin, 2019).

Broadly speaking, short read sequence data are those generated with a read length below 1,000 bp. Short read sequencing predates long read approaches and were largely based on the Sanger dideoxy-termination method established in the late 1970s (Sanger, Nicklen, & Coulson, 1977). The basic approach uses uniquely fluorescently labelled nucleotides incorporated to DNA in a PCR-like amplification process. Following the incorporation of each new nucleotide the labels are read, usually via laser excitation emission detection, generating a series of nucleotide specific sequences. When combined with a randomized fractioning of the target DNA, ligation of an adapter, and massive parallelization the process becomes readily scalable providing the ability to quickly and efficiently generate enough short sequence reads to statistically represent an entire higher-metazoan genome (Schuster, 2007; Tucker, Marra, & Friedman, 2009).

Randomization of the fractioning, known as shotgun sequencing, is an important aspect for later assembly. Using known target sites — such as a restriction digest sequences — will, in theory, create identical reads. This can be useful to identify the rate of incorrect incorporation of a nucleotide into a DNA strand during sequencing. However, for larger sequences (i.e., greater than 1,000 bp) using an identical sequence to
fraction does not allow the reconstitution of the original sequence (Anderson, 1981). Random fractions can help mitigate this issue, however they are limited by repeat sequences spanning more than the read length, as well as the possibility that stochastic fractioning can occur at the same location in different copies of the source DNA as shown in Figure 2.2. The former is a limitation of short read assembly approaches, whereas the impact of the latter can be further mitigated by sequencing more than a single copy of the target DNA, collectively known as coverage depth (Meyerson, Gabriel, & Getz, 2010).

**Multiple Copies of a Genome**

---

**Reads**

---

**High Coverage**

---

**Low Coverage**

---

**Consensus Sequence**

![Figure 2.2 – Contig Generation Process](http://gcat.davidson.edu/phast/img/coverage.png)

Illustrating the random fractioning of a genome followed by sequencing to generate sequence reads. These reads will be stochastically distributed across the genome to generate regions of high and low coverage. Increased read depth can offset potential gaps in sequencing coverage when these reads are reconstituted into contigs. Retrieved from: http://gcat.davidson.edu/phast/img/coverage.png.

Single-strand termini are most likely to be incorrectly matched during elongation of the new daughter strand due to the nature of the DNA polymerase used. To address this constraint, many sequencing approaches ligate adapters of known unique sequences to the ends of the fractionated strands, which can then be parsed out of the data before assembly. These adapters can also be used as a unique, identifiable label, allowing for
‘multiplexing’: the simultaneous sequencing of DNA for multiple sources in a single sequencing run. Once the sequence data has been collected it can be sorted by the unique adapter, which can then be removed from the read data. These adapters also facilitate a control against contamination, since only the target DNA will have been ligated and, thus, any contaminant DNA is highly unlikely to share the sequence (Illumina, 2017).

The high demand for large-scale genome sequence data has spurred the development of cheap, fast technologies, usually achieved by mass parallelization of the base strand sequencing method. These technologies are collectively referred to as second generation, or next-generation, sequencing methods achieve high-throughput by having the ligated adapter bind to a unique complementary strand immobilized to a solid surface (Voelkerding, Dames, & Durtschi, 2009).

Manufacturers of different sequencing apparatus have established a number of protocols; the most common among them include ion semiconductor, pyrosequencing, sequencing by synthesis, sequencing by ligation, and chain termination (L. Liu et al., 2012; Quail et al., 2012). These approaches were most famously used to generate an initial draft of the Human genome sequence in 2001. Illumina® has further refined this approach, resulting in a significant reduction in the cost/base ratio and an increase in the read yield in newer systems such as the NovaSeq™ system, which aims to achieve whole genome sequencing for $100 or less (Philippidis, 2019).

More recently, new technologies have exploited a growing understanding of the chemical-physical characteristics of DNA to generate sequence reads much longer than ‘traditional’ sequencing methods. Often referred to as ‘third generation sequencing’, these methods involve two main approaches: zero-mode waveguides employed in single-molecule real-time (SMRT®) sequencing, and nanopore oriented approaches employed by Oxford Nanopore™, Quantapore, and Stratos Genomics (Eisenstein, 2012; Levene et al., 2003). These technologies share the benefit of much longer sequence reads averaging ~30 kbp and up to 2.2 Mbp, an inbuilt ability to potentially detect epigenetic signals and enhanced throughput (Payne, Holmes, Rakyan, & Loose, 2018).

Despite these advantages there exist common shortcomings to third-generation approaches, almost entirely associated to error rates (Gupta, 2008). The processes operate in real time and, as the strands are moved rapidly through the target apertures, the signal
from any single base can be blurred with those from flanking bases. The common accuracy of these methods is approximately 90%, which raises additional challenges when trying to identify individuals within a population of organisms of the same or similar species that share more than 99.9% primary sequence identity in their respective genomes (Gupta, 2008).

For the project described here, the Illumina® HiSeq® 2000 platform was primarily used. Alternative technology platforms included the Ion Torrent™ self-titled platform, Roche® 454® pyrosequencing, Applied Biosystems SOLiD™, and Illumina® MiSeq®. The HiSeq® platform was the most appropriate choice for my work, given the combination of improved cost per base, base call accuracy, platform availability, read yield, and time required.

The HiSeq® platform is capable of sequencing short or long-insert paired-end reads. Long-insert paired-end reads are known as mate-pairs (MPs), while short-insert are simply called paired-end (PE) reads. Generally, PEs have a single read length between each of the reads at either end of the fragment being sequenced, while MPs can vary from 2–20 kbp. MP generation involves additional DNA preparation steps: the ends of long fragments are biotinylated, circularized, fragmented, and subjected to biotin amplification before adapter ligation. The known distance between the sequenced ends in turn lends itself to assisting later analyses directed to the scaffolding of individual reads (Bioinformatics, 2017; Illumina, 2017).

As mentioned, repeat regions of a genome spanning a distance larger than about twice the read length (e.g., satellite DNA regions) present challenges with short read assemblies. Since these will generate entire reads consisting of a repeated element, combining the reads into a contig will generate multiple potential contigs from the same reads with equal likelihood of accuracy. MPs can address these challenges where the satellite regions match the MP distance; however, this approach lacks reliability (Treangen & Salzberg, 2011). A better approach to generating correct DNA sequence data spanning such repeat regions is to use long read data from third-generation sequencing platforms (Biosciences, 2015). This project relied on PacBio® Sequel® system technology to generate four OAC-Rex and five PI440795 long-read libraries.

These libraries used SMRT® sequencing, wherein a DNA polymerase is
immobilized to the bottom of a nanopore well (called a ‘zero-mode waveguide’ (ZMW) hole) and is bound to a single strand of target DNA. The properties of light in a small aperture are such that the optical field within the ZMW is limited to the area immediately around the polymerase molecule (Foquet et al., 2008). Nucleotides bound to 1 of 4 fluorescent tags (to identify each nucleotide) are added to the reaction well and, as a nucleotide is incorporated into the elongating strand, the formation of the phosphodiester bond releases the fluorescent molecule which can be detected by laser excitation. The movement of the DNA and nucleotides within the well will quickly displace the free dye molecule and eliminating the signal as each new labeled nucleotide is added to the elongating strand, thus generating a DNA sequence in real time (Levene et al., 2003).

**Data Specifications**

As sequence data is gathered it is converted into the nucleotide sequence of G, A, C, or T/U. Given potential ambiguity during sequencing a definitive base call is not always possible, where the probability of a given nucleotide read is expressed according to an IUPAC standardization established for the purpose as shown in Table C.1 (Appendix C).

Over the past few decades, large-scale sequence data has been gathered in many formats and involving varied platforms. More recently however, a de facto standard output from high-throughput sequencers utilizes the fastq format. This format combines sequence data with its associated quality data (probability of error) in a single file and is the first state of the data used in most assembly pipelines. Other formats include; EMBL, GCG(-RSF), GenBank, IG, or sequencer native formats (GmbH, 2019; Leinonen, Sugawara, Shumway, & International Nucleotide Sequence Database, 2011).

Quality data from sequencers is most often translated into a ‘PHRED’ score, which is an ASCII representation of a numerical value associated to a likelihood of the associated sequence call being accurate. Developed by Phil Green in the 1990s, the ‘PHRED’ scoring algorithm is an automated process used to identify reads from electrophoresis gels and alleviates the bottleneck of manual identification. This program calculates the predicted location of a signal peak in a gel and assigns a probability of the base call accuracy based on the actual location of a peak relative to the prediction.
equation of ‘PHRED’ scores is $Q = -10\log_{10}P$; a commonly accepted minimum score of ‘PHRED’ 20 translates to a 99% probability of an accurate base call, and a practical upper range of ~40, which correlates to 99.99% accuracy (Ewing & Green, 1998; Ewing, Hillier, Wendl, & Green, 1998). Since each nucleotide is represented by a single character, while the quality score is almost always a two-digit number, ‘PHRED’ scores are used as an American Standard Code for Information Interchange (ASCII) decimal value, converting them to a single character as described in Figure C.1 (Appendix C).

This conversion allows for rapid and unambiguous reading of billions of base calls produced from a single sequencing run, together with their associated quality scores by most sequence analysis applications. The first 31 characters of ASCII do not represent actual characters, but special conditions for the computer to recognize (Association, 1963; A. N. S. Institute, 1986; U. S. o. A. S. Institute, 1967). As such, ‘PHRED’ scores are calculated as $Q+33$, making a ‘PHRED’ score of 1 be represented by ‘!’, or for older sequencers ‘Q+64’, making a ‘PHRED’ score of 1 be represented by ‘@’ (Cock, Fields, Goto, Heuer, & Rice, 2009).

When assembly pipelines combine fastq reads to their various stages they perform an array of statistical analyses of their assembled outputs to determine the most likely accurate assembly. As such, the quality scores initially provided by the sequencers are no longer applicable and a per-base quality score cannot reliably be determined or represented within the range of the ASCII table. These output sequences are stored in many different formats, the most common and simple of which is fasta. Fasta files may be single-entry or multi-entry with each entry consisting of 2 parts; an identifier, denoted be a ‘>’ followed by a unique name along with optional additional data associated to the entry, followed by the sequence data on at least one new line (Lipman & Pearson, 1985; W. R. Pearson & Lipman, 1988).

Once constructed, the assemblies serve as the template for annotation, the process by which structural and genetic components of the associated genomes are identified and positioned. There are many different algorithmic approaches for identifying the array of different genetic and structural components that can be annotated. The basic approach is identifying nucleotide sequence characteristics/patterns that are associated with known genetics elements; for example, reading frames and codons to translate into proteins. To
orient all the annotated elements, the nucleotide positions within the assemblies are used as anchor points. These nucleotide positions can be the locus that encode a genetic element, a region targeted by an element, or are in some other way linked to an element (Pevsner, 2009; Stein, 2001). A standard format for this kind of information is the General Feature Format version 3 (gff3), a multi-entry file using nine tab-delimited fields on a single line per entry to describe the characteristics of the annotated element. This standardized format for representing annotations can be used in many different gene browsers (Generic Model Organism Database, 2007b).

**De novo versus Guided Assembly**

A current limitation in the sequencing of large eukaryotic genomes is that no sequencing technology can span the length of an entire chromosome in a single read. As such, reads must be assembled. Two main approaches for read assembly are widely used: *de novo* and guided assembly, each of which retain their individual merits and demerits depending on the objective of the assembly (L. Liu et al., 2012; Quail et al., 2012).

*De novo* (‘from the beginning’) assemblies are those derived entirely from the data contained within the sequencing reads. These are the first-generation genome assemblies made for a newly sequenced organism. Multiple algorithms exist to achieve this objective, again, each with their own associated benefits and drawbacks. Two approaches dominate the current methodological landscape: ‘greedy’ algorithm versus De Bruijn graphing. The principal strength of *de novo* assemblies is the resolution they can provide to an assembled genome. *De novo* assemblies can identify genetic elements down to the nucleotide, which can be used to identify point mutations, SNPs, or in/dels. Because these changes can involve as little as one nucleotide, in a chromosome of up to thousands of millions, accurately identifying them requires extensive read depth of high-quality reads to be able to reconstruct them. This increase in data complexity is accompanied by increased processing time and associated computational resources (Bang-Jensen, Gutin, & Yeo, 2004; Idury & Waterman, 1995; Eugene W. Myers, 1995).

Guided assemblies require the *a priori* existence of a substantially complete and related assembly, which is used as a base against which the reads are aligned. The major advantage in this approach is that it minimizes gaps, since less certainty in the raw reads
can be offset by the assumed quality of the reference. That assumption of reference quality can be a dangerous one since it may misguide the assembly; however, the assembly process does not entirely rely on the reference sequence, such that output may still overcome shortcomings in the reference should the reads be of sufficient quality. The immediate benefit of a guided assembly is a drastically reduced read depth needed for a high-quality assembly, which in turn reduces the computational load and processing time. The trade-off is that the assembled genome will be of lower resolution and is likely to have sequence variations that go undetected (Lischer & Shimizu, 2017).

Whether de novo or guided approaches are used, the general strategy is the same for a given assembly; filter out duplicate reads keeping the highest quality version, followed by generating gap-free contiguous sequences called contigs. If sequence data with known size is available these can be used to align contigs to each other, inserting gaps as necessary, to generate scaffolds. Scaffolds are subsequently anchored to a specific linkage group to identify the chromosomes to which they belong, followed by additional work to fill in any gaps in the assembled sequences (Batzoglou et al., 2002; Lander et al., 2001; E.W. Myers et al., 2000).

Given the unknown parental input and subsequent genome rearrangements that can be associated with introgression approaches, a de novo assembly is necessary in order to better elucidate what components and mechanisms may be involved. That said, the close evolutionary relationship between OAC-Rex, PI440795, and G-19833 means that a guided assembly approach is useful to anchor those scaffolds/contigs that could not be unambiguously mapped de novo. Indeed, it has been argued that a combination of the two approaches, though somewhat unorthodox, would likely generate the best available product for OAC-Rex (English et al., 2012).

**Contig Generation**

The first step in the assembly process is to generate contigs from the base DNA read data. The process is almost identical regardless of algorithmic approach: reads are aligned against each other to find overlapping regions between one read and all others in the set. From the potential alignments a statistical comparison is performed to identify the highest likelihood match; where two or more alignments are equally likely, the chosen
match is often randomly selected from those possible. Once the alignment has been determined, the reads are combined to generate a new contig, which is returned to the read pool while the component reads are removed. The next read in the pool is selected and the process is repeated until no further combining can be done with the reads left in the pool. The pool can be further filtered for specific characteristics, such as minimum length or number of reads incorporated (Fullwood, Wei, Liu, & Ruan, 2009; Staden, 1980).

The process of generating contigs is a vital first step in assembly and can be done with short or long reads to expand the contig set. Combining the foundational DNA reads into contigs makes them easier to manage as a smaller computational data set, and their (generally) greater length allows for superior annotation of the physical and genetic elements they contain. The process of generating contigs also allows the identification of closely-related sequence variants, referred to as single nucleotide polymorphisms (SNPs), since these will exhibit a high-quality mismatch between them and their wildtype counterparts (Boisvert, Laviolette, & Corbeil, 2010; E.W. Myers et al., 2000).

The output from contig generation can vary widely between algorithms, depending upon the cut-offs set at execution. Assembled contigs can range in size from one base pair larger than the smallest read up to millions of nucleotides long. The average length of a contig set is, understandably, impacted by the average read length, but high quality reads with good depth coverage can construct very large high-quality contigs for subsequent analysis (Boisvert et al., 2010; Butler et al., 2008; E.W. Myers et al., 2000).

**Scaffold Assembly**

Once contigs are assembled they can potentially be combined into larger sets known as scaffolds. The key distinction between contigs and scaffolds is the presence of gaps. Scaffolding requires a dataset with a known distance between reads, such as Illumina® mate-pair reads. By aligning these reads to the contig set, contigs that are not yet connected can be linked by the known distance between them, with any unknown sequence range between them filled with an ‘N’ placeholder (see Figure 2.3). Long reads can be used to scaffold as well, although this is distinct from the contig extension process since it requires the low-quality of the bases in the long read being below a threshold set.
for the scaffolding algorithm. These low-quality bases may also be replaced with the ‘N’ placeholder for the purposes of scaffold organization (Butler et al., 2008; Waterston, Lander, & Sulston, 2002).

Scaffolds can be further completed via gap closing, a process where the scaffolds/contigs/reads not incorporated in higher order assemblies are aligned to the scaffolds. A lower quality threshold is generally required to gap fill than to contig or scaffold. Optimally, the reads being aligned to the gapped scaffold will have an anchor on both sides of the gap to support the filling. This process can be done iteratively in an attempt to further combine the available reads (R. Li et al., 2010).

**Pseudochromosome Construction**

After assembling the reads into either contigs or scaffolds based on their alignments to each other, an important next step is the grouping of the assembled reads based on linkage groups or chromosomes. There is no single approach to accomplish this, but they all require some kind of associated pattern to be exploited within the assembled reads. Using a genomic element with a known position, such as a karyotype map or a SNP profile as an anchor, the assembled reads can be linked to a specific chromosome.
The exact location on the chromosome may not be identifiable depending on the element-anchor used, and after multiple assembled reads are anchored a gap-fill with either ‘N’, or ‘-' may be required if the gap is of an indeterminate length (Schmid & Deininger, 1975).

Once again, upon anchoring the scaffolds, contigs, and reads to a specific chromosome to generate a pseudochromosome, gap filling can be performed. This is particularly useful in situations where a region of interest has only low-resolution genetic information: for example, a known qualitative trait locus (QTL). Associating contigs/scaffolds to their anchors can bring the tails of the reads within sufficient proximity to generate detectable patterns. This approach can potentially identify genetic elements that span whatever gap that may remain between two anchored reads (R. Li et al., 2010).

Narrowing the focus of study to a particular region does not always require anchoring the reads. From a functional perspective, scaffolds and contigs can contain all the necessary information to be annotated accurately. Structurally speaking, the assembled genome will contain much of the same information whether anchored or not, with the obvious exception of identifying what chromosome a given read may belong to.

For the purposes of this project, it was desirable to associate the reads with specific chromosomal locations since the CBB resistance traits had only been associated to imprecise chromosomal locations in the form of QTLs associated with molecular markers: notably SU91, UBC420, and PvCTT001. Given that QTLs are, generally, low resolution qualifiers, the ability to draw any informative conclusions vis-à-vis introgression events and the genetic determinants involved requires having a finer scale assembly associated to those regions. Knowing the approximate location of these loci allows for a targeted investigation of the regions to identify the genetic element(s) likely responsible for the resistance trait (Perry et al., 2013).

Another useful approach for assembling pseudochromosomes involves the use of iterative assembly techniques, in which emerging pseudochromosomes can be used as a reference for another assembly of the same reads. This approach will often refine the resulting assembly, providing a better representation of the actual genome being assembled. A caveat is that this approach depends on the accuracy of the initial assembly: any mistakes that are not corrected before using a pseudochromosome set will only be
Genome Feature Annotation

Annotation is the process of identifying patterns within a sequence that encode specific genomic structural or genetic elements and is vital for the utilization of sequence information for inferring biological functions. Knowing which loci are responsible for a specific function allows for directed experimentation to alter that function to a specific end, such as improving crop quality, treating disease, or strategies directed to metabolic engineering. A fundamental rule to annotation is that the higher quality the read, the better the annotations; any errors or uncertainty within an assembly itself may mask a key element in the derived annotation (Ekblom & Wolf, 2014).

The process of annotating genetic elements varies widely depending on what elements are being annotated, although the basic concept is the same. Biological studies have long identified shared components of biological molecules/processes. Using these shared elements as a basis, one can find a pattern that is uniquely associated with a genetic element. Once these patterns have been identified in one genome, the same pattern can be sought out in other organisms known to share the element. The existence of this pattern across organisms suggests that the pattern is informative of genomic content. Where a pattern is not identified across organisms, this suggests that the pattern is novel to the initial organism in which it was identified. Most annotations are putative predictions based on these underlying patterns and, therefore, are strictly theoretical. Additional validating experiments are needed to generate supporting evidence of any functional prediction (Consortium, 2011; Stein, 2001).

Perhaps the most common genetic element annotated is that of a gene that expresses a transcript or protein as the functional product. Defined as a sequence of DNA that encodes a functional RNA, these are the basic instructions for biological machinery. The basic approach in identifying a gene involves the identification of an open reading
frame, a specific frame of reference by which to interpret DNA via codons (H. Pearson, 2006).

Since DNA exists on earth as deoxynucleotides, but code for 20 canonical amino acids (AA), the genetic code is based on at least 3 nucleotides being required to combinatorially encode for all 20 AA. This set of three nucleotides is referred to as a codon. Depending on the initiating nucleotide involved, all subsequent triplet codons will define a reading frame from which a protein product will be defined. For the most part, it is the AUG codon that initiates the start of coding in eukaryotes (Table C.1). Using this triplet to identify potential start locations, the codon sequence can then be read until any of 3 canonical “stop” codons. Once this nucleotide range has been identified, it can be examined for other patterns such as intron/exon boundaries or regulatory regions. After identifying the structural components of the gene, any resulting predicted polypeptide sequence can be examined for functional aspects, such as known protein domains like a zinc-finger, or common post-translational modification sites (Alberts et al., 2008).

There are many different algorithms that can be used to identify genomic features, each with their own requirements, strengths, and weaknesses. A useful approach is that of consensus annotation, wherein the outputs of different algorithms are combined to assess where they agree on a specific annotation feature. Given the strength of evidence from differing sources, the consensus annotation is very likely an accurate representation of reality, despite its purely predictive nature (Campbell, Holt, Moore, & Yandell, 2014; Cantarel et al., 2008). The result of an effective annotation is not only the location of potential genes, but an estimation of the functional roles those genes may play. This data is often stored as a gff entry to an associated genome that can be examined and shared by other researchers in support of a wide array of biology projects (Campbell et al., 2014; Cantarel et al., 2008).

Not all genetic elements are defined as genes. There are instances where an informative pattern is the fingerprint of some biological process that has occurred. A prime example of this is long terminal repeats (LTRs), a sequence of nucleotides that repeats up to thousands of times. LTRs are characteristic of retrotransposons as a class of transposable element (TE), or retroviruses that leave a distinct fingerprint on both ends of the inserted DNA. LTRs are identical in sequence on both 5’ and 3’ ends of the locus
following transposition within the host genome. Because of this, comparing mutations within the two LTRs flanking the inserted sequence can be used to attempt to ascribe an evolutionary chronology to the insertion event in support of studies directed to patterns of genome evolution (Alberts et al., 2008).

Identical tandem sequence repeats often serve to identify LTRs, where an analysis of the sequence between the repeats will confirm or refute the locus as an LTR. LTR-retrotransposons can vary in size from 1–15 kbp and express genes that encode at least two main proteins, gag and pol. Tandem repeat sequences that are separated by more than this sequence range, or that lack gag or pol genes can usually be excluded, although most algorithms will assign a score to the identification of each putative LTR annotation (Ellinghaus, Kurtz, & Willhoeft, 2008; Xu & Wang, 2007).

Other data sets important for genome annotation include those arising from RNA sequencing, commonly referred to as RNAseq. Given that genes encode functional RNAs (either structural or protein-coding) incorporating the study of RNA data sets can help identify expressed genes, including those that may be differentially expressed across developmental time and/or space. The process of generating RNA sequence data largely mimics DNA sequencing. The first step is isolation of RNA, for which exaction protocols can vary depending upon which type of RNA is sought. For instance, mRNA can be enriched from selected cells or tissues at defined stages of development using oligo-dT templates that bind to the 3’-poly-A tail commonly associated with most protein-encoding mRNAs. Once isolated the RNA is converted to DNA complimentary to the RNA, called cDNA, by subjecting it to a retroviral form of polymerase known as reverse transcriptase. cDNA for a gene will differ from the gene sequence itself (gDNA) as the mRNA may be a processed molecule; for instance, introns will have been spliced out and any post-transcriptional modifications will be represented in the cDNA, but not in the gDNA. The subsequent cDNA sequencing process is identical to that of gDNA (Z. Wang, Gerstein, & Snyder, 2009).

A key distinction between RNAseq and DNA sequencing is the importance of identical reads. gDNA input for purposes of genome assembly will filter identical reads in order to avoid computational loops, whereas RNAseq will use each copy of an identical read to represent the RNA abundance at the time of collection. RNA abundance can have
critical impact on the proper functioning of a cell and this indirect measure can be used to build an understanding of the mRNA pool, or “transcriptome”, for a given organism. Using this data one can interpret what genes are being expressed by an organism in specific cells or tissues, and/or under specific conditions, which can in turn lead to functional prediction for pathways/molecules (Mortazavi, Williams, McCue, Schaeffer, & Wold, 2008).

Other RNAs are similarly important to living systems as mRNA but are often underrepresented. For example, plants make extensive use of non-coding RNA (ncRNA), which are any RNAs that do not encode a protein. These RNAs are categorized into different types based on size and/or predicted function, such as extracellular (exRNA), long non-coding (lncRNA), micro (miRNA), ribosomal (rRNA), small interfering (siRNA), small nuclear (snRNA), small nucleolar (snoRNA), or transfer (tRNA), to name a few. Most of these RNAs serve regulatory roles in the cell, often by regulating gene expression or in some cases as a communication molecule between different tissues (Alberts et al., 2008; Lucas, Yoo, & Kragler, 2001).

Identifying the array of ncRNAs follows largely the same process as mRNA, with a modification to the purification step. The location of the ncRNAs within the genome are not always informative, however those with a regulatory role can be identified based on complementarity to known or predicted ‘target’ genes elsewhere in the genome. Given the regulatory roles many of them play, knowing which genomic positions are targeted can elucidate new layers of control for cellular processes. For example, knowing the target of piwi-interacting RNA (piRNA) has assisted in the identification of silenced TEs (Siomi, Sato, Pezic, & Aravin, 2011). Given the highly conserved nature of rRNAs they can be used to help refine phylogenetic analysis, or to indirectly measure the quality of a genome assembly (Woese & Fox, 1977).

When analyzing the relatedness of organisms at a genetic level synteny becomes an invaluable dataset. Defined as a conservation of order when two sequences are compared, synteny can be evaluated between primary sequence data sets (nucleotides/AAs) or the higher-order structures formed by those primary sequences (gene/motif order). This is a potentially valuable tool since closely related organisms will share a high synteny than those more distantly related. Similarly, when comparing
sequences of parent to offspring synteny can be used as a validation of the assembly process (Engström, Ho Sui, Drivenes, Becker, & Lenhard, 2007; Kurata et al., 1994).

When attempting to identify regions that have been incorporated through genetic introgression, the analysis of synteny can be useful for linking a sequence derived from one input parent versus the other. A challenge, however, is that synteny relies on a threshold of similarity: where the threshold is too low, non-syntenic areas will be identified as syntenic, whereas if it is set too high then potentially meaningful syntenic alignments will be obscured (D. Liu, Hunt, & Tsai, 2018). This issue of assignment of identity-thresholds in the analysis of genomic synteny is of importance to the current project, given the close relatedness of all three Phaseolus genomes analyzed in the conduct of this project.

**Genome Visualization**

Given the sheer volume of information contained within a single genome, organization and presentation of the data is an important and significant task. This also makes visualization of said data difficult, since the resolution can scale from one to millions of nucleotides for a single linkage group and no single level will clearly represent all the available information. To this end an interactive display medium has been developed known as a gene browser. Often offered as online tools, these browsers allow the user to investigate a sequence from a macro level that can cover entire chromosomes to a micro level where they display every individual nucleotide (J. Wang, Kong, Gao, & Luo, 2012; Wilkinson et al., 2002).

A repeated theme throughout the assembly and annotation processes is the alignment of pertinent data to a nucleotide sequence. By using this universal reference system, all annotations can be brought into a visual system whereby a shared location on an axis will give spatially organized information at any given genome position. From this co-visualization, deductions can be more easily identified and exploited vis-à-vis the annotated elements within a genome (Stein et al., 2002).

There are many options for gene browsers; some are specific to a single organism while others can be used generically for any sequence data and associated annotations.
The three best known are likely to be the Ensembl Genome Browser, NCBI’s Map Viewer, and the UCSC Genome Browser. Each has its own specific implementation methods and requirements, but another common option is the ‘GBrowse’ or ‘JBrowse’ genome browsers from the generic model organism database (GMOD) consortium (Hubbard et al., 2002; NCBI Resource Coordinators, 2016; Stein et al., 2002; Tyner et al., 2017).

Developed as a collaboration between several model organism databases, the GMOD consortium has worked to create software tools that would be theoretically needed by any large-scale sequencing project. These include genome editors/viewers, database management tools, and tools used for analysis. Originally a common gateway interface (CGI) tool, ‘GBrowse’ was re-engineered into the JavaScript-based ‘JBrowse’. Both browsers were made freely available as open source software and have extensive community support providing an array of modules (e.g., ‘BLAST’ functionality) that can be readily added to any installation as needed (Stein et al., 2002).

**Data Refinement and Management**

Given the vast assortment of tools that are available, a common issue for researchers is data formatting. Basically, program X will give output in format 1, but program Y will need an input of format 2. In some instances there are shared formats, such as bam/sam, or software available to convert the data appropriately, but far more often it is up to the researcher to correct this (Ahmed et al., 2017). Much of the code outlined in Appendix 2 was written for this purpose and applied in this study.

**Data Quality Assessment**

Given the importance of drawing sound scientific conclusions, the data upon which such conclusions are based must be of a high quality. Potential flaws in collection protocols may not manifest themselves until analysis begins, and faults in the data may lead to inaccurate conclusions despite proper analysis. The means to accomplish this data quality assessment will vary widely depending upon the type and format of data collected (Cover & Thomas, 1991).
One of the most difficult aspects of data management in scientific research is the distinction between noise and signal: what values in a dataset are truly indicative of a measured condition and which are the stochastic alignment of variables approximating a true signal. The importance of this distinction should be obvious; reading noise as signal will lead to false conclusions which, if encountered frequently lead to a situation where false conclusions and the hypothesis can neither be validated nor refuted. This becomes increasingly important in a strictly computational approach with large data such as genome assembly and annotation. Because the assembly is already a probabilistic outcome based on quality scores of reads and the annotations are probabilistic interpretations thereof, noise regarded as signal can have far reaching downstream impacts. High thresholds of accuracy and/or probability during each algorithmic step are essential to ensuring the final published assembly and associated annotations are as close to reality as possible (Cover & Thomas, 1991; Nakagawa & Cuthill, 2007).

An obvious solution to coping with the signal to noise ratio is to use conservative thresholds that will remove all but the greatest outliers of noise. The problem with such an approach is that the signal may inherently be a weak one, for example determining whether a nucleotide in a read is truly an error in sequencing (noise) or a point mutation (signal). In this context, setting thresholds becomes a balancing act that can often best be dealt with through trial and error and using some specific trait as an indirect measure of the overall assembly (Cover & Thomas, 1991; Nakagawa & Cuthill, 2007).

For this project, the critical step involving data quality assessment came during the initial read cleanup. Based on how ‘PHRED’ scores are calculated, a score of 28 is needed to reach ~99.9% likelihood of an accurate read. For that reason, and given the reach of analyses planned for OAC-Rex, the initial reads were filtered for a ‘PHRED’ score of 30 (Ewing & Green, 1998; Ewing et al., 1998). Any reads with a base quality scores below that threshold were trimmed, removing bases between the low quality read and the nearest end. Once trimmed, reads were then filtered such that any reads reduced to 80% of their initial size or less were filtered from the pool.

Trimmed reads were further filtered to remove any duplicate entries. Given the number of reads generated a direct alignment to identify identical reads would be impractical despite the availability of many filters that can be used to expedite the process
(e.g., a Bloom filter). Developed in the 1970s by Burton Howard Bloom, for whom they are named, Bloom filters are probabilistic data structures that will determine whether an element is a member of a set. This is done by creating a bit array of \( m \) bits, all set to 0, and using \( k \) hash functions, each of which map one of \( n \) elements to one of the bits within the array, changing it to 1. The design of the algorithm makes it such that while false negatives cannot occur, as any of the \( k \) hash functions that map to a bit in the array still 0 indicate that no element has produced this outcome from the hash functions. However, false positives can occur as the array is populated from the repeated testing of elements in the set. It is possible that elements X and Y will, between them, map to the same array positions as element Z, where the probability of such a false positive can be defined by: 

\[
\left(1 - \left[1 - \frac{1}{m}\right]^k\right)^k \approx \left(1 - e^{-kn/m}\right)^k.
\]

where \( m \) is the number of bits in the array, \( k \) is the number of hash functions, and \( n \) is the number of elements in the set. While this value can get extremely low it can never reach 0. Generally less than 10 bits per elements will achieve a false positive rate below 1%, regardless of the size of the set (Bloom, 1970; Cormen, 2009).

Bloom filtering can be used to identify unique reads from a sequencing run far more quickly and using much less computational space than other methods. The bit array does not need to interact with the actual data stored in each element beyond running the hash functions, thus providing a space advantage. With each element reduced to a series of \( k \) bits the memory needed to run a bloom filter will only be \( m + k + 8l \) bits; \( m \) for the array itself, \( k \) for the hash results of the element being tested, and 8 bits for each nucleotide in a read of length \( l \), since a character requires eight bits, or one byte. Once an element has been tested it is simply written to disk in a new file or skipped entirely. The time benefit arises from every element requiring only the amount of time required to run \( k \) hash functions then compare \( k \) bits in the array, regardless of the number of items already in the set. This is a unique feature to Bloom filters and keeps the entire procedure to a linear order of operations: \( O(k) \) (Cormen, 2009; Demetrescu, 2007).

With a single sequencing run generating millions or billions of reads, the advantages conferred by a Bloom filter become readily apparent. While the computational time and space saved for any single read is minute, the cumulative impact
is significant. Take for example a typical Illumina® short read library of 500 million reads each 100 bp long: to add or check any element a Bloom filter would use $O(16)$ and would require slightly more than 1 GB of memory to filter with a false positive rate below $1/1000$. For a binary search tree, possibly the most rudimentary and common of search filters, the same library would take $O(8.9)$, to lookup, plus $O(500,000,000)$, to insert/remove and enumerate the elements, and would require up to $\sim 46$ GB of memory (Demetrescu, 2007; Knuth, 1998).

The next step in data quality refinement involves assembly validation, which can be done after contig generation, scaffold generation, pseudochromosome assembly, and gap closing. There are essentially two variations on assembly validation: structural and functional. Structural validation of an assembly incorporates characteristics of the sequences themselves; read length, number and length of gaps, gaps/read, N50, % coverage, etc. The most often scrutinized values are perhaps number of reads, longest read, mean read size, N50, %N, and % coverage. Of these values, N50 requires some explanation. N50 values are calculated by ordering the scaffold lengths in decreasing order then summing the lengths starting from the longest. When the sum of the lengths is equal to or greater than 50% of the total length of all the reads, the length of the last read added is reported as the N50 value. While an odd value, this is instructive to a researcher when taken in conjunction with the longest read, mean read length, and number of reads, as it allows the researcher to quickly develop a distribution profile of the assembled reads. Knowing this profile helps compare different assembly strategies, since a better assembly will generally exhibit fewer, longer reads overall (Bradnam et al., 2013; Earl et al., 2011).

To assess these structural characteristics of an assembly, a common tool to use is the ‘Assemblathon_stats’ Perl script. Developed primarily at the UC Davis Genome Center, the assemblathon contests to assess the best methods available for genome assembly. To assess the assemblies generated, a standardized testing platform was developed that collects the information from the submitted assemblies and parses it for scaffolds and contigs, contigs within a scaffold being separated by a sequence of twenty five sequential Ns (Bradnam et al., 2013; Earl et al., 2011).
Functional validation undertakes to analyze a focussed or partial annotation of the reads to give an indirect measure of completeness. Using a known element that should be found within the assembly as an indirect measure, researchers can estimate how accurately the assembly reflects reality. This known element can be an organism-specific feature such as a unique gene, or it can be measured by looking for (a) universal genetic element(s), such as the 355 genes deemed to have been present in the last universal common ancestor. A eukaryotic variant of such a gene set includes 248 genes likely found with few, if any, in-paralogs across a range of eukaryota, referred to as Core Eukaryotic Genes (CEGs). In-paralogs are those wherein at lineages share a duplication, but each loses the reciprocal paralog, and using a dataset where they are minimized is important since they could be, falsely, identified as orthologs. Identification of CEGs in complete, partial, and orthologous forms via alignment of the gene components is the basis of the ‘CEGMA’ software’s assessment algorithm. This approach has since been further improved-upon by Benchmarking Universal Single-Copy Orthologs (BUSCO), which has datasets specific to different taxa (Parra, Bradnam, & Korf, 2007; Simão, Waterhouse, Ioannidis, Kriventseva, & Zdobnov, 2015).

This indirect measure of functional completeness is a useful verification between each stage of assembly since it reports on the consistency of the content of the reads. From contigs to scaffolds, or scaffolds to pseudochromosomes, the movement of reads containing elements of a CEG can alter the completeness profile of that CEG. Should the reads be incorrectly assembled they can be reported as such and used as an indirect indication that the assembly is incorrectly combining reads. The sampling cannot be performed on any subset of genes as the vast majority of genes are of unknown function, often only having been putatively predicted during annotation of a genome without any lab-based validation (Parra et al., 2007; Theobald, 2010).

Annotations are predictions based largely on the extrapolation of patterns found in the nucleotide sequence. Because they are inherently predictive, most annotations are not actually identified as being the predicted element, but simply match the pattern as identified in other organisms. This theoretical annotation may often have large amounts of evidence to support the prediction due to the highly conserved nature of the element, for instance rRNAs, although often the annotations are left tagged as putative gene calls.
until lab-based validation can be performed. The specifics of how this lab-based validation occurs varies widely depending on the element being validated (Dybas et al., 2008; Theobald, 2010).

Since genomic annotation is an exercise in pattern matching and given that our understanding of biological complexity is incomplete, it is all but guaranteed that mistakes will be made. Errors during sequencing, signal to noise ratio filtering, assembly, or annotation may give rise to a nucleotide sequence that exhibits all the patterns needed to be a gene, but which is not a gene due to some hitherto unknown biological regulatory mechanism. As research progressively reveals more of the patterns behind biological complexity, they will refine the annotation process and help to more accurately interpret genomic data. It is for these reasons that first round annotations are labelled as putative elements, to inform any future research that they are purely hypothetical interpretations (Yandell et al., 2005).

**Assembly Strategy**

For OAC-Rex and PI440795 genome assembly, the strategies involved varied, although there were common strategies for both. The different approaches were justified by a difference in the format of the base data: OAC-Rex had more PE read depth, PacBio® coverage, and two MP libraries, whereas the underlying PI440795 data lacked MP reads, although the lack of long read data was offset by deeper PacBio® read depth. PacBio® data for OAC-Rex was not generated until several later iterations of the genome assembly had been constructed.

As a first and obligatory step, all sequence data used in the assemblies was verified via ‘md5checksum’ to ensure no data was lost during transmission. Once verified the multiple smaller files were concatenated into a single file for each library using the ‘cat’ application common to all POSIX-compliant Linux operating systems including Ubuntu®. These raw reads were then analyzed using ‘FastQC’, where every file was manually curated to identify potential issues such as improper marker trimming, low quality base calling or overexpressed sequences (S. Andrews, 2010). Custom code was used to trim any sequencing marker contamination based on the Illumina® published PE
adapter sequences (‘Fasta_linker_trimmer’; this work). PE reads for both OAC-Rex and PI440795, along with OAC-Rex’s MP reads were submitted to read-trimming using the ‘fastq_quality_trimmer’ from ‘FASTX-Toolkit’, specifically for a base call minimum quality of 30 and a minimum length of 80 bp or 120 bp, respectively (Hannon, 2009). Custom code was then used to sort the trimmed data (‘Fastq_sorter’; this work), verify sorting (‘Fastq_sort_check’; this work), and each library made continuous using ‘Fastq_synchronizer’ to yield forward (R1) and reverse (R2) files of paired reads in identical order along with a file of singletons whose matching read was filtered. These trimmed reads were used as the basis for all further Illumina®-based assembly.

Multiple assembly strategies were developed and either retained or abandoned during assembly of a draft genome sequence for OAC-Rex, the goal being to generate the best possible assembly from the data available. The first assembly approach was shared by both OAC-Rex and PI440795; contig assembly using ‘Ray’ (Boisvert et al., 2010). Several ‘Ray’ assemblies were generated at varying $k$ values that were then assessed using ‘Assemblathon_stats’ before settling on the $k = 41$ assemblies as optimal. All other ‘Ray’ assemblies were later used to help train respective ‘MAKER’ annotations (Cantarel et al., 2008). As various assembly strategies were assessed the ‘-disable-scaffolder’ option was used for most ‘Ray’ assemblies because, while ‘Ray’ performs well for the generation of contigs, there are other software options that produced better scaffolds. In such instances where the scaffolder did not allow the submission of a separate contig assembly, custom code (‘Fasta_APE_generator’; this work) was developed to generate artificial PE reads (APEs) of 120 bp long, with a 40 bp overlap between forward and reverse reads, a 20 bp overlap between each read pair, and an arbitrarily high quality score of ‘PHRED’ 40.

The first scaffold set of OAC-Rex was assembled on the Mustang platform (McGill University) using ‘SOAPdenovo’ using exclusively Illumina® short read libraries for contiging, scaffolding, and gap closing (R. Li et al., 2010). The contigs from this run were compared to those from the k41 ‘Ray’ contigs using ‘Assemblathon_stats’ and found to be of lower quality as evidenced by lower overall length, shorter contig size, lower N50, and more contigs per assembly. Conversely the scaffolds from ‘SOAPdenovo’ showed higher quality than those from ‘Ray’. The k41 ‘Ray’ scaffolds
were then submitted along with all Illumina® short read libraries for scaffolding and gap closing. The scaffolds from this second ‘SOAPdenovo’ run exhibited higher quality than the first based on analysis via ‘Assemblathon_stats’ and ‘CEGMA’. Consequently, the outputs from the first approach along with both contigs and scaffolds, were abandoned and the output from the ‘Ray’ + ‘SOAPdenovo’ run was resubmitted for two more iterations of scaffolding and gap closing. The third iteration did not show a significant increase in assembly quality via ‘CEGMA’ or ‘Assemblathon_stats’, resulting in the second being kept as the final scaffold set (Bradnam et al., 2013; Parra et al., 2007).

The third and fourth scaffold set were assembled on Mustang using ‘MaSuRCA’ utilizing all Illumina® short read libraries, as well as an assembly using all Illumina® short read libraries and ‘Ray’ APEs (Zimin et al., 2013). Both assemblies were of much lower quality than the ‘Ray’ + ‘SOAPdenovo’ assembly based on total assembled size and number of contigs alone, and ‘MaSuRCA’ was abandoned as an alternative.

The fifth scaffold set was assembled on Mustang using ‘AllPaths-LG’ using all Illumina® short read libraries and ‘Ray’ APEs as input (Gnerre et al., 2011). The strategy to use ‘Ray’ to generate contigs that were then given to other assemblers was also used for other assembly projects unrelated to this one, using different software combinations, since this approach was found to deliver consistently superior results. As a result, ‘Ray’ contigs were adopted to be the base contig set for all further assemblies. The output from this approach greatly outperformed all other assemblies assessed across all aforementioned assembly projects, and so this assembly was kept as the scaffold set for OAC-Rex, which were subsequently submitted for annotation.

The assembled scaffolds and contigs of OAC-Rex lacked any indication of to which chromosome they belonged. This renders only the scaffolds large enough to contain all genomic data associated to an introgression event useful in downstream analysis. Such a limitation would vastly limit the potential scope of the investigation; thus, anchoring the scaffolds became a requirement, which in turn required some feature to act as the anchor for the scaffolds. The lack of information regarding \textit{P. vulgaris} genomics left no clear choice of feature; therefore, a SNP profile for G-19833 was used to generate a pseudochromosome set for OAC-Rex. A genomic interpretation of evolution suggests that increased homology, and thus synteny, indicates organisms are
more closely related to each other. Given the close relatedness of the two lines of *P. vulgaris*, these SNPs were hypothesized to exhibit a common synteny with respect to their approximate order and position. These SNPs were first aligned using ‘BLASTn’ to the G-19833 assembly and filtered for a minimum of 1e-50 alignment likelihood (Altschul, Gish, Myers, & Lipman, 1990; Song et al., 2015). The output was then sorted by match quality and the highest match for each SNP was selected. These best matches were used to rename the SNP based on their chromosomal coordinate information, and those SNPs with an insufficient quality match were removed from the pool. The newly labeled SNPs were subsequently aligned against the assembled OAC-Rex scaffolds, keeping the same minimum quality cut-off. In the case of multiple matches, the higher quality match was used as the anchor; in the case of an identical match score, the first match on the list was used. Using the location label as the base data, the reads were reorganized and combined when possible using custom code (‘Pseudochromosome_assembler’; this work) on the Banting platform. An issue arising from this assembly strategy was that the introgressed *P. acutifolius* DNA would generate larger reads between the SNP locations that incorporate an introgression event, which could in turn create an overlap between assembled reads when the read ‘tails’ were reorganized. In such instances the pseudochromosome was split into multiple entries and labelled to indicate which nucleotides the SNPs would have anchored them to. These ‘split’ pseudochromosomes were also combined into a single-entry set where these breaks were replaced by a 25N placeholder.

Due to an error in the first version of the pseudochromosome assembler, the first-generation OAC-Rex pseudochromosomes were re-run using the same SNP set and ‘AllPaths-LG’ scaffolds. The resulting second version pseudochromosomes were shared as the first draft genome assembly of OAC-Rex. This draft was submitted to the annotation pipeline previously described and the combined data was subsequently uploaded to a local ‘GBrowse’ installation (Figure 2.4).
Figure 2.4 – Screenshot of the ‘GBrowse’ Gene Browser

Displaying a contig from the OAC-Rex k41 ‘Ray’ based contig assembly with associated ‘BLASTp’ alignment and ‘MAKER’ gene annotation.

It was at this point (July of 2014) that long-read PacBio® sequence data became available for OAC-Rex, with PI440795 PacBio® data becoming available shortly afterwards. PacBio® raw reads were run through ‘FastQC’ to assess their quality. For OAC-Rex, this data was corrected using short read data via ‘PBcR’ on the Banting platform (Koren et al., 2012); however, when compared to the provided PacBio® subreads the filtered data exhibited a decrease in quality and were abandoned. The OAC-Rex v2 pseudochromosome data were subsequently subjected to gap closing using the PacBio® sequence data and ‘PBJelly’ on the Banting platform (English et al., 2012).

PI440795’s ‘Ray’ k41 assembly had been annotated on Banting while OAC-Rex was being refined. With the newly available PacBio® data, several attempts to further its assembly were performed. An initial attempt at gap closing of the contigs using ‘PBJelly’ did not significantly improve the contigs. In a second approach, custom code (‘Fastq_AMP_generator’; this work) was written to parse artificial MP reads from the PacBio® data. These reads, along with PI440795 ‘Ray’ APEs were input to ‘AllPaths-LG’ to assemble a scaffold set on the Beanblade platform. The validation of these scaffolds showed a drop in structural quality from the ‘Ray’ contigs as well as a significant drop in ‘CEGMA’ matches, and thus were abandoned.

With a functional assembly for all three input *Phaseolus* parental lines now in-hand, the synteny between all three was investigated in order to search for early patterns than may have been associated with introgression events in OAC-Rex. Individual
syntenic alignments were performed using ‘Mauve’ to examine the syntenic relatedness between OAC-Rex, PI440795, and G-19833 (Darling, Mau, Blattner, & Perna, 2004). During this procedure it was discovered that chromosome 08 showed minimal rearrangement and was largely unchanged across the 3 species, whereas chromosome 04 exhibited a complex pattern of rearrangements. Consultation with collaborators at U of Guelph confirmed that this finding was an expected representation of OAC-Rex’s pedigree (Figure 2.5).

![Figure 2.5 - G-19833 v1 Aligned to OAC-Rex Pseudochromosomes v2 via ‘Mauve’](image)

Colour coded segments represent syntenic blocks with a line connecting the corresponding blocks between G-19833 v1 (top) and OAC-Rex v2 (bottom). The red lines denote a new entry in the supplied fasta files.

Using the ‘Mauve’ alignments as a target, custom code (‘Mauve_title_parser’; this work) was written to parse out regions where elements of each assembly aligned to each other. Impossible alignments were generated from the approach, revealing an assumed concatenation in the way ‘Mauve’ incorporates multi-entry fastas. Custom code (‘Mauve_title_corrector’; this work) was written to correct the assumption before parsing. From this parsed data a ‘ClustalW’-format alignment was generated using custom code (‘Mauve_clustal_converter’; this work), which was then used to upload the alignment data to ‘GBrowse_syn’ to readily allow for visualization of synteny across the three organisms (Generic Model Organism Database, 2007a; Thompson, Higgins, & Gibson, 1994).
Figure 2.6 – ‘GBrowse_syn’ Screenshot

Displaying the terminus of the long arm of chromosome 8 from OAC-Rex pseudochromosomes v2 (blue square), with associated gene annotations, aligning to two PI440795 ‘Ray’ generated contigs (red squares)

From this syntenic view, shown in Figure 2.6, the location of the SU91 QTL marker, together with the associated annotations in those loci, it was revealed that one gene in G-19833 and two genes in OAC-Rex were the most likely candidates for a CBB-associated introgression. All three were identified as being similar to the NPC1-like gene (NPC1l1) found in mice. In humans, NPC1 is a gene associated with Niemann-Pick disease, type C (NPC), a lysosomal storage disease. Further investigation of these three genes showed that the two OAC-Rex predicted genes were comprised of an insert in a single gene annotated in G-19833, resulting in the disruption and presumed inactivation of the gene.

Since the PI440795 assembly was limited to a ‘Ray’ contig set, the available data was limited for further in-depth analysis. As well, the assembly strategy for OAC-Rex was deemed impractical given the differences between the read data sets. In an alternative approach, a second contig set for PI440795 was assembled on Beanblade using ‘CANU’ given all the available PacBio® long read libraries (Koren et al., 2017). This assembly was of a comparable quality to the ‘Ray’ k41 assembly, both structurally and functionally. The contig sets were then reconciled on Beanblade using ‘GARM’ given both the ‘Ray’ k41 and ‘CANU’ contigs (Luz Mayela, Karel, & Alejandro, 2014). The resulting assembled contig set showed a marked improvement over either of the substitutive sets, and this reconciled contig set was retained as the PI440795 contigs.
A second assembly of G-19833 was released in late 2016 that incorporated long-read data. As this assembly was the basis for the pseudochromosome anchoring, the assembly process was redone using this new information to generate OAC-Rex Pseudochromosomes v3. During this process several variants of the assembly protocol were used in an attempt to expedite the process, but each generated an assembly that was of lower quality than the earlier established protocol. As a result, v3.3 was derived using the long-form protocol, annotated and shared as OAC-Rex v3.

PI440795 pseudochromosomes were also assembled. A second pseudochromosome assembly was created where the alignment of the renamed SNPs to the ‘GARM’ contigs was conducted without the 1e-50 requirement, given that G-19833 and PI440795 are different (though related) species; a relaxing of the SNP alignment was thought to allow for evolutionary impact of the ancient speciation. The ‘relaxed’ assembly showed a noticeable improvement over its ‘stringent’ counterpart in every measure. As a result, the OAC-Rex pseudochromosome data set was also regenerated with the ‘relaxed’ protocol, which showed improvement over v3.3. The resulting data set was annotated and shared as OAC-Rex v4.

**Annotation**

The scaffolds produced by ‘AllPaths-LG’ were submitted to ‘MAKER’ for gene annotation on the Banting platform. The ‘MAKER’ outputs were compiled into a gff3 file using ‘gff3_merge’ and a fasta file generated using ‘fasta_merge’. The resulting MAKER.proteins.fasta file was used to annotate the identified protein-encoding genes via a ‘BLASTp’ against the UniPROT ‘Swiss-Prot’ database and MAKER_functional_gff”, in combination with ‘interproscan’ and ‘ipr_update_gff’ (Apweiler et al., 2000; Bairoch & Apweiler, 1996). The twice-updated gff file was kept as the gene annotation of OAC-Rex and this process was repeated for all future ‘MAKER’ annotations.

‘TRF’, ‘LTR_finder’, ‘LTRHarvest’, and ‘tRNAscan-SE’ were used to annotate non-coding genomic elements on Banting (Benson, 1999; Chan, Lin, Mak, & Lowe, 2019; Ellinghaus et al., 2008; Xu & Wang, 2007). Custom code was written to convert the outputs from each to a correctly formatted gff3 file (‘TRF_gff_generator’,

48
‘LTR_finder_gff_generator’, ‘LTRHarvest_gff_corrector’, and ‘tRNAscan-SE_gff_generator’, respectively; this work). These corrected gff3 files were concatenated into a single gff3 file and retained as the non-coding annotation of OAC-Rex. This same process was repeated for all future annotations.

Originally, all the data collected was loaded into a PostgreSQL database using GMOD’s chado schema and configured to be viewed via a local ‘GBrowse’ server implementation (Mungall, Emmert, & Consortium, 2007; Stein et al., 2002). Over the timespan of the assembly and annotation of OAC-Rex and PI440795, ‘JBrowse’ was created and implemented to replace the earlier ‘GBrowse’ implementation compared in Figure 2.7 (Skinner, Uzilov, Stein, Mungall, & Holmes, 2009). OAC-Rex pseudochromosomes v4, PI440795 ‘GARM’ pseudochromosomes, and G-19833 v2 assemblies and annotations were reordered and updated to match the new format before being uploaded to a local implementation of ‘JBrowse’.

The clustering of data of finer scale into many tracks allowed for a more useful review of the data by researchers. ‘BWA’ aligned contigs and scaffolds to the pseudochromosomes, and ‘BLASTn’ based alignment of ncRNA sets found from NCBI’s sequence read archive (SRA) were included to the consolidated data set (Leinonen et al., 2011; H. Li & Durbin, 2009).

Identification of Points of Introgression (POIs)

With a mature genome assembly and annotation available for all three parental input sources, attention could be turned to the objective of identifying sites in OAC-Rex
where conspecific DNA from *P. acutifolius* was incorporated into the genome resulting in a point of introgression (POI). CBB resistance markers lent themselves as prime candidates for modelling POIs, focusing particularly on those located to chromosome 8 due to its relatively low amount of rearrangement, and secondarily to those on chromosome four as the linkage group harbouring the PvCTT001 QTL marker and which exhibited a relatively high degree of rearrangement (Perry et al., 2013).

At its core, POI identification depends on determining the regions of the hybrid introgressed OAC-Rex that can be uniquely attributed to one input parent or the other. Once these locations with unique parentage are identified they can be sorted to parse out the locations where there is a switch in genome parentage. At some point between a position unique to parent A and a position unique to parent B, an introgression must have occurred. This approach is dependent on accurately identifying unique parentage in the hybrid, and therefore benefits from a more conservative/stringent interpretation of parentage.

The first attempt to identify parentage derived from straightforward ‘Mauve’ alignments like those in Figure 2.8. Syntenic blocks on chromosome 8 that uniquely mapped to either parent were grouped to find the smallest range between differing parentage. These so-called zones of introgression varied massively in size as well as nucleotide sequence, ranging from as little as 859 bp to over 880 kbp. However, when these zones were aligned against themselves, they exhibited very weak synteny overall, and no syntenic sequence was shared across a significant portion of the identified zones of introgression.

![Figure 2.8 – Chromosome 8 Syteny Between G-19833 v1 and OAC-Rex Pseudochromosomes v2 via ‘Mauve’](image)

Colour coded segments represent syntenic blocks with a line connecting the corresponding blocks. The general order of synteny is maintained from G-19833 v1 (top) but additional unmatching DNA, denoted by white sections within the coloured blocks, is expanding the size of the blocks in OAC-Rex v2 (bottom). The red lines denote a new entry in the
From this data it was believed that introgression was likely a process that incorporated large sections of DNA, akin to a meiotic recombination. For this reason, the approach was refined to instead use a multi-sequence alignment (MSA) of the entirety of chromosome 8 from each assembly using ‘BWA’, ‘ClustalO’, ‘MAFFT’, ‘Muscle’, and ‘T-COFFEE’ (Edgar, 2004; Katoh, Misawa, Kuma, & Miyata, 2002; Notredame, Higgins, & Heringa, 2000; Sievers et al., 2011). These alignments proved to be too large even for the impressive resources available on Beanblade. As a result, only the terminal 5 Mbp were parsed out to be aligned. This subset proved still too large, resulting in the generation of custom code (‘Fasta_range_subset_extractor’; this work) to parse out sections of the assemblies in a rolling window fashion with a defined overlap. Using a window of 1 Mbp, with 500 kbp overlap, the terminal 5 Mbp were extracted and aligned successfully. These MSA files were then visualized using NCBI’s ‘MSA viewer’ (National Center for Biotechnology Information, 2018) and seemed at first to reveal large sections of unique parentage. These sections were then further examined with the Recombination Detection Program (Martin, Murrell, Golden, Khoosal, & Muhire, 2015) to identify fingerprints left behind by a putative introgression event. This examination revealed a flaw in the large-scale approach in that none of the overlapping portions corresponded to those fingerprints revealed by ‘RDP4’.

A finer examination of the constituent reads revealed that the large sections that seemed to indicate introgression were in fact a high density of very small alignments, generally only 2–5 nucleotides in length. Another rolling-window extraction was therefore conducted using a 500 kbp window with 200 kbp overlap spanning the terminal 1 Mbp of chromosome 8. The resulting library showed the same issue as the previous. An alternative approach was undertaken that involved down-sizing the rolling-window size, an approach that had been used by others modeled on a subtractive RNA library methodology to identify all regions of OAC-Rex not found in G-19833 and which were assumed to be of PI440795 origin (Perry, G., Personal Communication). These segments were very often relatively small; only 100-200 bp in size, contradicting the presumption that introgression involved uptake of large-scale genomic segments.
To implement this new approach, the POI identification strategy was re-worked using custom applications (‘Fasta_title_assembler’, ‘Fasta_title_parser’, ‘Fasta_individualizer’, ‘Fasta_chunker’, and ‘Fasta_range_subset_extractor’; this work) that were expressly written or updated to do the following:

1. Take only the constituent scaffolds within each pseudochromosome of OAC-Rex
2. Chunk those regions into sections of 100 bp (the size of a single read)
3. Identify the region in G-19833 to which it matched based on the SNP used to anchor each scaffold
4. Parse those regions into their own fasta, align the OAC-Rex chunks against the G-19833 region to which it was mapped with ‘BLASTn’
5. Align the OAC-Rex chunks against the entirety of the PI440795 ‘GARM’ contigs with ‘BLASTn’
6. Combine the outputs of OAC-Rex against each parent into a single file for each
7. Manually curate each individual OAC-Rex scaffold to align each 100 bp chunk to its counterpart (if present)
8. Calculate a ratio of homology based on the bitscore values produced by ‘BLASTn’ (Parent A bitscore/Parent B bitscore)

These parentage ratios could then be used to filter out the matches based on a calculated threshold. A ratio of 1 indicates a shared, likely ancestral region across all three organisms, a DNA segment closer to parent A would have a limit of 0 from the right, while a segment closer to parent B would approach infinity. Thus, for those segments that only align to parent A the ratio can be arbitrarily set 0, while those only from parent B can be arbitrarily set to 2. Although these values can be exceeded by the actual scores, setting the 0-2 range allows the calculation of a standard deviation from 1 (+/- 0.464). Using this strategy, an entire chromosome can be mapped scaffold by scaffold to reveal its parentage and filtered to varying stringency in order to computationally assign unique parentage as shown in Figure 2.9.
These scores were tabulated then used to generate a gff of parentage via custom code (‘POI_gff_generator’; this work) which were then uploaded to ‘JBrowse’. With these identifying factors aligned to other annotated elements in Chr04, it was revealed that a SNP within a repeat region identified be as a CBB resistance marker (Perry et al., 2013) may be the only distinction between *P. vulgaris* and *P. acutifolius*. This same alignment strategy of data for Ch08 revealed a change in introgression parentage at the same Niemann-Pick gene earlier predicted to have been inactivated and associated with CBB resistance, suggesting the revised approach is able to accurately determine points of introgression in OAC-Rex.

From the newly identified small scale POIs, the OAC-Rex scaffolds containing the above CBB markers, along with corresponding sequences from each of the parental lines were extracted and aligned against each other with ‘MAFFT’. These MSAs were then submitted to ‘RDP4’ to attempt to elucidate which mechanism(s) were responsible for the recombination. Since ‘RDP4’ is a meta level analysis package employing multiple recombination programs to develop a consensus on calling for recombination events, it is able to produce a large amount of evidence regarding recombination events (Figure 3.11 and Figure 3.12).
CHAPTER 3: RESULTS
Sequencing

P. vulgaris ‘OAC-Rex’

Raw Reads

DNA was collected from OAC-Rex plants and submitted for sequencing to generate the ten libraries described in Table 3.1.

Table 3.1 – OAC-Rex Raw Short-read Statistics

*Illumina®* short-read libraries assessed by ‘FastQC’.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Total Sequence Reads</th>
<th>Sequence Length (bp)</th>
<th>Total Sample Size (bp)</th>
<th>%GC</th>
<th>Average Quality per Read Peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>S5_PE_R1</td>
<td>140754247</td>
<td>101</td>
<td>14216178947</td>
<td>33</td>
<td>37.5</td>
</tr>
<tr>
<td>S5_PE_R2</td>
<td>140754247</td>
<td>101</td>
<td>14216178947</td>
<td>33</td>
<td>37</td>
</tr>
<tr>
<td>S6_PE_R1</td>
<td>132970554</td>
<td>101</td>
<td>13430025954</td>
<td>33</td>
<td>37.5</td>
</tr>
<tr>
<td>S6_PE_R2</td>
<td>132970554</td>
<td>101</td>
<td>13430025954</td>
<td>33</td>
<td>37</td>
</tr>
<tr>
<td>S7_PE_R1</td>
<td>115459789</td>
<td>101</td>
<td>11661438689</td>
<td>33</td>
<td>37.5</td>
</tr>
<tr>
<td>S7_PE_R2</td>
<td>115459789</td>
<td>101</td>
<td>11661438689</td>
<td>33</td>
<td>38</td>
</tr>
<tr>
<td>2kbp_MP_R1</td>
<td>591487818</td>
<td>51</td>
<td>30165878718</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>2kbp_MP_R2</td>
<td>591487818</td>
<td>51</td>
<td>30165878718</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>5kbp_MP_R1</td>
<td>643153873</td>
<td>51</td>
<td>32800847523</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td>5kbp_MP_R2</td>
<td>643153873</td>
<td>51</td>
<td>32800847523</td>
<td>38</td>
<td>39</td>
</tr>
</tbody>
</table>

Trimmed Reads

The above libraries were subjected to Bloom filtering to remove duplicate entries and quality filtering to remove bases with quality scores below ‘PHRED’ 30 as well as a minimal length of 40 bp. These trimmed data sets were then sorted and made continuous using the ‘Fastq_sorter’ and ‘Fastq_synchronizer’ custom code (see Appendix II), with the resultant libraries described in Table 3.2.
Table 3.2 – OAC-Rex Trimmed Short-read Statistics

Illumina® short-read libraries assessed by ‘FastQC’ after trimming by FASTX toolkit’s ‘fastq_quality_trimmer’. All files were trimmed for a minimum quality of 30, PEs were further filtered for a minimum length of 40 bp.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Total Sequence Reads</th>
<th>% Reads Filtered</th>
<th>Sequence Length (bp)</th>
<th>Total Sample Size (bp)</th>
<th>%GC</th>
<th>Average Quality per Read Peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>S5_PE_R1</td>
<td>138358986</td>
<td>1.7017327</td>
<td>40-101</td>
<td>13761910235</td>
<td>33</td>
<td>37.5</td>
</tr>
<tr>
<td>S5_PE_R2</td>
<td>138358986</td>
<td>1.7017327</td>
<td>40-101</td>
<td>13586356946</td>
<td>33</td>
<td>37</td>
</tr>
<tr>
<td>S6_PE_R1</td>
<td>130940811</td>
<td>1.5264605</td>
<td>40-101</td>
<td>13030590682</td>
<td>33</td>
<td>37.5</td>
</tr>
<tr>
<td>S6_PE_R2</td>
<td>130940811</td>
<td>1.5264605</td>
<td>40-101</td>
<td>12892421656</td>
<td>33</td>
<td>37.5</td>
</tr>
<tr>
<td>S7_PE_R1</td>
<td>113516105</td>
<td>1.6834294</td>
<td>40-101</td>
<td>11300855509</td>
<td>33</td>
<td>37.5</td>
</tr>
<tr>
<td>S7_PE_R2</td>
<td>113516105</td>
<td>1.6834294</td>
<td>40-101</td>
<td>11158300482</td>
<td>36</td>
<td>38</td>
</tr>
<tr>
<td>2kbp_MP_R1</td>
<td>95281317</td>
<td>83.891246</td>
<td>2-51</td>
<td>4645338634</td>
<td>36</td>
<td>39</td>
</tr>
<tr>
<td>2kbp_MP_R2</td>
<td>95281317</td>
<td>83.891246</td>
<td>2-51</td>
<td>4506705129</td>
<td>36</td>
<td>39</td>
</tr>
<tr>
<td>5kbp_MP_R1</td>
<td>15462920</td>
<td>97.595767</td>
<td>2-51</td>
<td>750621003</td>
<td>36</td>
<td>39</td>
</tr>
<tr>
<td>5kbp_MP_R2</td>
<td>15462920</td>
<td>97.595767</td>
<td>2-51</td>
<td>724515152</td>
<td>36</td>
<td>39</td>
</tr>
</tbody>
</table>

OAC-Rex DNA was again collected and submitted for PacBio® sequencing to generate four long read libraries, described in Table 3.3.

Table 3.3 – OAC-Rex Raw Long-read Statistics

PacBio® long-reads assessed by ‘FastQC’.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Total Sequence Reads</th>
<th>Sequence Length (bp)</th>
<th>Total Sample Size (bp)</th>
<th>%GC</th>
<th>Average Quality per Read Peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>A05</td>
<td>289456</td>
<td>35-32257</td>
<td>1606574117</td>
<td>37</td>
<td>9</td>
</tr>
<tr>
<td>A07</td>
<td>304984</td>
<td>35-35742</td>
<td>1868372473</td>
<td>37</td>
<td>9</td>
</tr>
<tr>
<td>B07</td>
<td>279548</td>
<td>35-35410</td>
<td>1726502858</td>
<td>37</td>
<td>9</td>
</tr>
<tr>
<td>C01</td>
<td>118307</td>
<td>35-30245</td>
<td>795369712</td>
<td>36</td>
<td>8.5</td>
</tr>
</tbody>
</table>
Illumina®-based sequencing of OAC-Rex yielded 3247652562 short-reads of either 51 bp (MPs) or 101 bp (PEs) in length spanning a total of ~204 Gbp, an equivalent of ~284x coverage. After being filtered for duplicates and trimmed for a quality score no lower than 30 there were 987,120,278 (~70% filtered) reads whose length ranged from 2-101 bp to cover ~83 Gbp (~59% filtered), equating to ~115x coverage (~60% reduction in coverage). 992,295 PacBio® long-reads were later generated ranging in size from 35–35,742 bp totalling ~20 Gbp (~28x coverage). These long-reads were left unfiltered and untrimmed.

**P. acutifolius PI440795**

*Raw Reads*

The same collection and sequencing protocols for OAC-Rex were repeated for PI440795 to generate four short read and five long-read libraries, described in Table 3.4.

*Table 3.4 – PI440795 Raw Read Statistics*

*Illumina® short-reads and PacBio® long-reads assessed by ‘FastQC’.*

<table>
<thead>
<tr>
<th>Sample</th>
<th>Total Sequences</th>
<th>Sequence Length (bp)</th>
<th>Total Sample Size (bp)</th>
<th>%GC</th>
<th>Average Quality per Read Peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>440795_L1R1</td>
<td>86192402</td>
<td>151</td>
<td>13015052702</td>
<td>33</td>
<td>36</td>
</tr>
<tr>
<td>440795_L1R2</td>
<td>86192402</td>
<td>151</td>
<td>13015052702</td>
<td>33</td>
<td>36</td>
</tr>
<tr>
<td>440795_L2R1</td>
<td>69637719</td>
<td>151</td>
<td>10515295569</td>
<td>33</td>
<td>37</td>
</tr>
<tr>
<td>440795_L2R2</td>
<td>69637719</td>
<td>151</td>
<td>10515295569</td>
<td>33</td>
<td>37</td>
</tr>
<tr>
<td>A08</td>
<td>983913</td>
<td>35-45128</td>
<td>5576221543</td>
<td>35</td>
<td>10</td>
</tr>
<tr>
<td>B06</td>
<td>1040903</td>
<td>35-48099</td>
<td>6471756119</td>
<td>35</td>
<td>9</td>
</tr>
<tr>
<td>B08</td>
<td>804619</td>
<td>35-45917</td>
<td>4427333688</td>
<td>35</td>
<td>10</td>
</tr>
<tr>
<td>C08</td>
<td>273729</td>
<td>35-46375</td>
<td>1464647553</td>
<td>34</td>
<td>10</td>
</tr>
<tr>
<td>E04</td>
<td>943613</td>
<td>35-43579</td>
<td>5324452264</td>
<td>35</td>
<td>9</td>
</tr>
</tbody>
</table>
**Trimmed Reads**

PI440795 short read libraries were Bloom filtered, trimmed for base quality of ‘PHRED’ 32 and minimum length of 896 bp, sorted, and made continuous. These libraries are described in Table 3.5, below.

**Table 3.5 – PI440795 Trimmed Short-read Statistics**

Illumina® short read libraries assessed by ‘FastQC’ after trimming by FASTX toolkit’s ‘fastq_quality_trimmer’. All files were trimmed for a minimum quality of 32 and a minimum length of 96 bp.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Total Sequences</th>
<th>% Reads Filtered</th>
<th>Sequence Length (bp)</th>
<th>Total Sample Size (bp)</th>
<th>%GC</th>
<th>Average Quality per Read Peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>440795_L1R1</td>
<td>61466155</td>
<td>28.687270</td>
<td>96-151</td>
<td>9215490200</td>
<td>32</td>
<td>37</td>
</tr>
<tr>
<td>440795_L1R2</td>
<td>61466155</td>
<td>28.687270</td>
<td>96-151</td>
<td>9175363638</td>
<td>32</td>
<td>37</td>
</tr>
<tr>
<td>440795_L2R1</td>
<td>52049059</td>
<td>25.257375</td>
<td>96-151</td>
<td>7805547728</td>
<td>32</td>
<td>37</td>
</tr>
<tr>
<td>440795_L2R2</td>
<td>52049059</td>
<td>25.257375</td>
<td>96.151</td>
<td>7774341909</td>
<td>32</td>
<td>37</td>
</tr>
</tbody>
</table>

PI440795 was submitted for sequence generation of four short-read libraries. These libraries contained 311660242 reads, all of which were PEs and 151 bp in size, which spanned a combined ~47 Gbp (~72x coverage). These reads were trimmed and filtered to remove duplicates and ensure a minimal base call score of 32, as well as ensuring no read was shorter than 96 bp. The resultant dataset contained 337030428 reads, an ~27% decrease, totaling ~34 Gbp in length, ~28% less than unfiltered and ~52x coverage, with reads ranging in size from 96–151 bp in size. An additional five libraries long-read were generated spanning 35–48,099 bp and covering ~23 Gbp (~36x coverage) held within 4,046,777 reads.
Assembly

OAC-Rex

Contigs
The trimmed OAC-Rex PE libraries were used to generate multiple \textit{de novo} assemblies using ‘Velvet’, ‘Celera Assembler’, and ‘Ray’ assembly software. These assemblies were assessed using the ‘Assemblathon\_stats’ Perl package to gather structural statistics which were then compared. From these assemblies it was clear that ‘Ray’ was a superior contig assembler, but sub-optimal at scaffold assembly. The OAC-Rex reads were then run through ‘Ray’ using variable k-mer sizes, which were likewise compared via ‘Assemblathon\_stats’ before determining a k-mer of 41 as being optimal. The OAC-Rex ‘Ray’ k41 contigs are described in further detail in \textbf{Table 3.6}.

\textit{Table 3.6 – OAC-Rex ‘Ray’ Statistics}

\textit{‘Assemblathon\_stats’ was used to gather structural statistics, gap statistics were collected using the ‘Fasta\_gap\_analyzer’ custom code (described in Appendix II), and CEG statistics were collected using ‘CEGMA’ for ‘Ray’ generated contigs with a k-mer set to 41}

<table>
<thead>
<tr>
<th>Sample</th>
<th>‘Ray’ k41 Contigs</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of Assembly in Scaffolded Contigs</td>
<td>0.0</td>
</tr>
<tr>
<td>% of Assembly in Unscaffolded Contigs</td>
<td>100.0</td>
</tr>
<tr>
<td>Average Number of Contigs per Scaffold</td>
<td>1</td>
</tr>
<tr>
<td>Average Length of Break (&gt;25Ns) Between Contigs in Scaffold</td>
<td>0</td>
</tr>
<tr>
<td>Number of Contigs</td>
<td>15794</td>
</tr>
<tr>
<td>Number of Contigs in Scaffolds</td>
<td>0</td>
</tr>
<tr>
<td>Number of Contigs not in Scaffolds</td>
<td>15794</td>
</tr>
<tr>
<td>Total Size of Contigs</td>
<td>396395565</td>
</tr>
<tr>
<td>Longest Contig</td>
<td>216696</td>
</tr>
<tr>
<td>Shortest Contig</td>
<td>100</td>
</tr>
<tr>
<td>Number of Contigs &gt; 500 bp</td>
<td>44644</td>
</tr>
<tr>
<td>Number of Contigs &gt; 1k bp</td>
<td>40748</td>
</tr>
<tr>
<td>Metric</td>
<td>Value</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Number of Contigs &gt; 10k bp</td>
<td>11103</td>
</tr>
<tr>
<td>Number of Contigs &gt; 100k bp</td>
<td>64</td>
</tr>
<tr>
<td>Number of Contigs &gt; 1M bp</td>
<td>0</td>
</tr>
<tr>
<td>Mean Contig Size</td>
<td>2510</td>
</tr>
<tr>
<td>Median Contig Size</td>
<td>168</td>
</tr>
<tr>
<td>N50 Contig Length</td>
<td>16442</td>
</tr>
<tr>
<td>L50 Contig Count</td>
<td>6230</td>
</tr>
<tr>
<td>N50 Contig – NG50 Contig Length Difference</td>
<td>16268</td>
</tr>
<tr>
<td>Contig %A</td>
<td>33.59</td>
</tr>
<tr>
<td>Contig %C</td>
<td>16.52</td>
</tr>
<tr>
<td>Contig %G</td>
<td>16.50</td>
</tr>
<tr>
<td>Contig %T</td>
<td>33.39</td>
</tr>
<tr>
<td>Contig %N</td>
<td>0</td>
</tr>
<tr>
<td>Contig N bp</td>
<td></td>
</tr>
<tr>
<td>Contig % non-ACGTN</td>
<td>0</td>
</tr>
<tr>
<td>Number of Contig non-ACGTN bp</td>
<td>0</td>
</tr>
<tr>
<td>Number of Gaps (25+ N)</td>
<td>0</td>
</tr>
<tr>
<td>Longest Gap (bp)</td>
<td>0</td>
</tr>
<tr>
<td>Mean Gap Length (bp)</td>
<td>0</td>
</tr>
<tr>
<td>Mean Gaps per Gapped Scaffold</td>
<td>0</td>
</tr>
<tr>
<td>Number of CEGs Present in a Complete Form</td>
<td>204</td>
</tr>
<tr>
<td>Number of CEGs Present in at least a Partial Form</td>
<td>236</td>
</tr>
<tr>
<td>Mean Number of Orthologs per Complete CEG</td>
<td>1.89</td>
</tr>
<tr>
<td>Mean Number of Orthologs per Partial</td>
<td>2.26</td>
</tr>
</tbody>
</table>
CEG

| Percentage of Detected Complete CEGs with More than one Ortholog | 51.96 |
| Percentage of Detected Partial CEGs with More than one Ortholog   | 64.41 |

‘Ray’ assembled 15,794 contigs ranging from as little as 100 bp up to 216,696 bp. These contigs covered ~396 Mbp, approximately 55% of OAC-Rex’s estimated 720 Mbp and generated a 16,442 bp N50. None of the assembled sequences were N, thus there were no gaps to assess, though ‘CEGMA’ identified 204 CEGs in their complete form and another 32 in a partial form.

**Scaffolds**

Further assemblies were generated using ‘MaSuRCA’, ‘AllPaths-LG’, and ‘SOAP’. These assemblies were assessed, as described in Table 3.7, to determine the best assembly pipeline of ‘Ray’ for contig generation and ‘AllPaths-LG’ for scaffold generation. ‘AllPaths-LG’ only allows the submission of short- or long-read data for assembly and so the ‘Ray’ k41 contigs were used to generate artificial PEs via the ‘Fasta_APE_generator’ custom code and given an arbitrarily large quality score using the ‘Fasta_2_fastq’ custom code (see Appendix II). These APEs were submitted to ‘AllPaths-LG’ to reconstitute the ‘Ray’ contigs within the ‘AllPaths-LG’ algorithm before scaffold generation.

**Table 3.7 – OAC-Rex Scaffold Statistics**

*All scaffolds were generated using ‘Ray’ k41 contigs or artificial PEs based thereupon. ‘Assemblathon_stats’ was used to gather structural statistics, gap statistics were collected using the ‘Fasta_gap_analyzer’ custom code (described in Appendix II), and CEG statistics were collected using ‘CEGMA’.*

<table>
<thead>
<tr>
<th>Sample</th>
<th>‘Ray’ + ‘SOAPdenovo’</th>
<th>‘AllPaths-LG’</th>
<th>‘MaSuRCA’</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Scaffolds</td>
<td>112622</td>
<td>8532</td>
<td>57703</td>
</tr>
<tr>
<td>Total size of Scaffolds</td>
<td>424551928</td>
<td>423349915</td>
<td>459265843</td>
</tr>
<tr>
<td>Total Scaffold Length as % of</td>
<td>59.0</td>
<td>58.8</td>
<td>63.8</td>
</tr>
<tr>
<td></td>
<td>Estimated Genome</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td><strong>Longest Scaffold</strong></td>
<td>1736762</td>
<td>2063709</td>
<td>528573</td>
</tr>
<tr>
<td><strong>Shortest Scaffold</strong></td>
<td>102</td>
<td>890</td>
<td>101</td>
</tr>
<tr>
<td><strong>Number of Scaffolds &gt; 500 bp</strong></td>
<td>12309 (10.9%)</td>
<td>8532 (100.0%)</td>
<td>54330 (94.2%)</td>
</tr>
<tr>
<td><strong>Number of Scaffolds &gt; 1k bp</strong></td>
<td>10932 (9.7%)</td>
<td>8508 (99.7%)</td>
<td>44261 (76.7%)</td>
</tr>
<tr>
<td><strong>Number of Scaffolds &gt; 10k bp</strong></td>
<td>5535 (4.9%)</td>
<td>5147 (60.3%)</td>
<td>10962 (19.0%)</td>
</tr>
<tr>
<td><strong>Number of Scaffolds &gt; 100k bp</strong></td>
<td>1032 (0.9%)</td>
<td>1053 (12.3%)</td>
<td>332 (0.6%)</td>
</tr>
<tr>
<td><strong>Number of Scaffolds &gt; 1M bp</strong></td>
<td>2 (0.0%)</td>
<td>14 (0.2%)</td>
<td>0 (0.0%)</td>
</tr>
<tr>
<td><strong>Mean Scaffold Size</strong></td>
<td>3770</td>
<td>49619</td>
<td>7959</td>
</tr>
<tr>
<td><strong>Median Scaffold Size</strong></td>
<td>150</td>
<td>15099</td>
<td>2005</td>
</tr>
<tr>
<td><strong>N50 Scaffold Length</strong></td>
<td>119359</td>
<td>158891</td>
<td>28296</td>
</tr>
<tr>
<td><strong>L50 Scaffold Count</strong></td>
<td>844</td>
<td>595</td>
<td>4069</td>
</tr>
<tr>
<td><strong>N50 Scaffold – NG50 Scaffold Length Difference</strong></td>
<td>38852</td>
<td>55440</td>
<td>15490</td>
</tr>
<tr>
<td><strong>NG50 Scaffold Length</strong></td>
<td>2508</td>
<td>1805</td>
<td>12806</td>
</tr>
<tr>
<td><strong>LG50 Scaffold Count</strong></td>
<td>80507</td>
<td>103451</td>
<td>9054</td>
</tr>
<tr>
<td><strong>Scaffold %A</strong></td>
<td>32.47</td>
<td>31.02</td>
<td>33.00</td>
</tr>
<tr>
<td><strong>Scaffold %C</strong></td>
<td>16.15</td>
<td>15.30</td>
<td>16.97</td>
</tr>
<tr>
<td><strong>Scaffold %G</strong></td>
<td>16.13</td>
<td>15.30</td>
<td>16.97</td>
</tr>
<tr>
<td><strong>Scaffold %T</strong></td>
<td>32.38</td>
<td>31.02</td>
<td>32.94</td>
</tr>
<tr>
<td><strong>Scaffold %N</strong></td>
<td>2.87</td>
<td>7.36</td>
<td>0.12</td>
</tr>
<tr>
<td><strong>Scaffold N bp</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Scaffold % non-ACGTN bp</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Number of Scaffold non-ACGTN bp</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>% of Assembly in Scaffolded Contigs</strong></td>
<td>85.7</td>
<td>94.5</td>
<td>1.5</td>
</tr>
<tr>
<td><strong>% of Assembly in Unscaffolded Contigs</strong></td>
<td>14.3</td>
<td>5.5</td>
<td>98.5</td>
</tr>
<tr>
<td></td>
<td>1.1</td>
<td>3.4</td>
<td>1.0</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>Average Number of Contigs per Scaffold</td>
<td>1.1</td>
<td>3.4</td>
<td>1.0</td>
</tr>
<tr>
<td>Average Length of Break (&gt;25Ns) Between Contigs in Scaffold</td>
<td>108</td>
<td>3654</td>
<td>9</td>
</tr>
<tr>
<td>Number of Contigs</td>
<td>125489</td>
<td>28743</td>
<td>58196</td>
</tr>
<tr>
<td>Number of Contigs in Scaffolds</td>
<td>18963</td>
<td>25894</td>
<td>736</td>
</tr>
<tr>
<td>Number of Contigs not in Scaffolds</td>
<td>106526</td>
<td>2849</td>
<td>57460</td>
</tr>
<tr>
<td>Total Size of Contigs</td>
<td>412378991</td>
<td>382342356</td>
<td>458908203</td>
</tr>
<tr>
<td>Longest Contig</td>
<td>492789</td>
<td>551924</td>
<td>528573</td>
</tr>
<tr>
<td>Shortest Contig</td>
<td>71</td>
<td>6</td>
<td>64</td>
</tr>
<tr>
<td>Number of Contigs &gt; 500 bp</td>
<td>23614 (18.8%)</td>
<td>28135 (97.9%)</td>
<td>54488 (93.6%)</td>
</tr>
<tr>
<td>Number of Contigs &gt; 1k bp</td>
<td>21034 (16.8%)</td>
<td>26433 (92.0%)</td>
<td>44382 (76.3%)</td>
</tr>
<tr>
<td>Number of Contigs &gt; 10k bp</td>
<td>8916 (7.1%)</td>
<td>9127 (31.8%)</td>
<td>10961 (18.8%)</td>
</tr>
<tr>
<td>Number of Contigs &gt; 100k bp</td>
<td>580 (0.5%)</td>
<td>533 (1.9%)</td>
<td>326 (0.6%)</td>
</tr>
<tr>
<td>Number of Contigs &gt; 1M bp</td>
<td>0 (0.0%)</td>
<td>0 (0.0%)</td>
<td>0 (0.0%)</td>
</tr>
<tr>
<td>Mean Contig Size</td>
<td>3286</td>
<td>13650</td>
<td>7886</td>
</tr>
<tr>
<td>Median Contig Size</td>
<td>156</td>
<td>4622</td>
<td>1987</td>
</tr>
<tr>
<td>N50 Contig Length</td>
<td>4401</td>
<td>391756</td>
<td>28173</td>
</tr>
<tr>
<td>L50 Contig Count</td>
<td>2359</td>
<td>2444</td>
<td>4093</td>
</tr>
<tr>
<td>N50 Contig – NG50 Contig Length Difference</td>
<td>29883</td>
<td>28590</td>
<td>15472</td>
</tr>
<tr>
<td>Contig %A</td>
<td>33.43</td>
<td>33.47</td>
<td>33.03</td>
</tr>
<tr>
<td>Contig %C</td>
<td>16.62</td>
<td>16.50</td>
<td>16.98</td>
</tr>
<tr>
<td>Contig %G</td>
<td>16.61</td>
<td>16.51</td>
<td>16.99</td>
</tr>
<tr>
<td>Contig %T</td>
<td>33.37</td>
<td>33.48</td>
<td>32.96</td>
</tr>
<tr>
<td>Contig %N</td>
<td>0.00</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>Contig N bp</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contig % non-ACGTN</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>--------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Number of Contig non-ACGTN bp</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Number of Gaps (25+ N)</td>
<td>12867</td>
<td>20211</td>
<td>493</td>
</tr>
<tr>
<td>Longest Gap (bp)</td>
<td>4724</td>
<td>6985</td>
<td>15989</td>
</tr>
<tr>
<td>Mean Gap Length (bp)</td>
<td>946</td>
<td>1534</td>
<td>725</td>
</tr>
<tr>
<td>Mean Gaps per Gapped Scaffold</td>
<td>2.110728</td>
<td>3.556396</td>
<td>2.028807</td>
</tr>
<tr>
<td>Number of CEGs Present in a Complete Form</td>
<td>217</td>
<td>215</td>
<td>212</td>
</tr>
<tr>
<td>Number of CEGs Present in a Partial Form</td>
<td>240</td>
<td>240</td>
<td>240</td>
</tr>
<tr>
<td>Mean Number of Orthologs per Complete CEG</td>
<td>1.72</td>
<td>1.67</td>
<td>1.78</td>
</tr>
<tr>
<td>Mean Number of Orthologs per Partial CEG</td>
<td>1.98</td>
<td>1.89</td>
<td>2.09</td>
</tr>
<tr>
<td>Percentage of Detected Complete CEGs with More than one Ortholog</td>
<td>42.86</td>
<td>41.40</td>
<td>46.23</td>
</tr>
<tr>
<td>Percentage of Detected Partial CEGs with More than one Ortholog</td>
<td>52.92</td>
<td>53.33</td>
<td>58.75</td>
</tr>
</tbody>
</table>

Three scaffold sets were assembled from the ‘Ray’ k41 contigs. The first used ‘SOAPdenovo’ and ‘GapCloser’ in three iterations to generate 112,622 scaffolds spanning ~425 Mbp (~59.0% of OAC-Rex’s estimated genome size), 2.87% of which were N held in 12,867 gaps that were up to 4,724 bp in length. The scaffolds generated ranged from 102 bp to 1,736,762 bp, with a mean length of 3,770 bp and an N50 of 119,359 bp, there were 18,963 contigs (84.9%) which could not be incorporated into scaffolds.
The second assembly was via ‘AllPaths-LG’, which only allows the submission of short- or long-read data for assembly and so the ‘Ray’ k41 contigs were used to generate artificial PEs via the ‘Fasta_APE_generator’ custom code and given an arbitrarily large quality score using the ‘Fasta_2_fastq’ custom code (see Appendix II). These APEs were submitted to ‘AllPaths-LG’ to reconstitute the ‘Ray’ contigs within the ‘AllPaths-LG’ algorithm before scaffold generation. The output from ‘AllPaths-LG’ was 8,532 scaffolds spanning ~423 Mbp (~58.8%), 7.36% of which were N in stretches up to 6,985 bp long across 20,211 gaps. The longest scaffold was 2,063,709 bp, while the shortest was 890 bp, distributed to create an N50 of 158,891 bp and a mean size of 49,619 bp. 2,849 contigs (9.9%) could not be incorporated into scaffolds.

The third assembly, produced by ‘MaSuRCA’, generated 57,703 scaffolds spanning ~459 Mbp (~63.8%), 493 gaps held the 0.12% of bases which were N, ranging up to 15,989 bp in length. An N50 of 28,296 bp was found from the scaffolds that ranged from 101–528,573 bp, with a mean of 7,959 bp and 57,460 contigs (~98.7%) which could not be incorporated into scaffolds.

**Pseudochromosomes**

‘AllPaths-LG’ scaffolds were anchored using a *P. vulgaris* SNP dataset to assign a chromosomal location to each scaffold using the ‘Pseudochromosome_assembler’ custom code (see Appendix II). The ‘Pseudochromosome_gff_updater’ custom code (see Appendix II) was initially used to update the annotations of the ‘AllPaths-LG’ assembly to match the new pseudochromosome location. After OAC-Rex pseudochromosomes v2 this approach was replaced with de novo annotation of pseudochromosomes and any unanchored scaffolds. These pseudochromosomes are further detailed in Table 3.8 and were used as the basis for all further investigation.
Table 3.8 – OAC-Rex Pseudochromosome Statistics

Every version of pseudochromosomes was generated using ‘AllPaths-LG’ scaffolds. ‘Assemblathon_stats’ was used to gather structural statistics, gap statistics were collected using the ‘Fasta_gap_analyzer’ custom code (described in Appendix II), and CEG statistics were collected using ‘CEGMA’.

<table>
<thead>
<tr>
<th>Sample</th>
<th>PChr v2</th>
<th>PChr v3.3</th>
<th>PChr v4</th>
<th>v4 No Scaffs</th>
<th>v4 Unanchored Scaffolds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Scaffolds</td>
<td>6246</td>
<td>6315</td>
<td>5680</td>
<td>3396</td>
<td>2287</td>
</tr>
<tr>
<td>Total size of Scaffolds</td>
<td>746210109</td>
<td>747372962</td>
<td>782479212</td>
<td>761039192</td>
<td>21331939</td>
</tr>
<tr>
<td>Total Scaffold Length as % of Estimated Genome</td>
<td>103.6</td>
<td>103.8</td>
<td>108.7</td>
<td>105.7</td>
<td>3.0</td>
</tr>
<tr>
<td>Longest Scaffold</td>
<td>23081690</td>
<td>10711518</td>
<td>5794880</td>
<td>5794880</td>
<td>376423</td>
</tr>
<tr>
<td>Shortest Scaffold</td>
<td>890</td>
<td>890</td>
<td>927</td>
<td>1006</td>
<td>927</td>
</tr>
<tr>
<td>Number of Scaffolds &gt; 500 bp</td>
<td>6246</td>
<td>6315</td>
<td>5680</td>
<td>3396</td>
<td>2287</td>
</tr>
<tr>
<td>Number of Scaffolds &gt; 1k bp</td>
<td>6222</td>
<td>6291</td>
<td>5659</td>
<td>3396</td>
<td>2266</td>
</tr>
<tr>
<td>Number of Scaffolds &gt; 10k bp</td>
<td>3294</td>
<td>3311</td>
<td>3318</td>
<td>3318</td>
<td>490</td>
</tr>
<tr>
<td>Number of Scaffolds &gt; 100k bp</td>
<td>1031</td>
<td>969</td>
<td>1278</td>
<td>1257</td>
<td>20</td>
</tr>
<tr>
<td>Number of Scaffolds &gt; 1M bp</td>
<td>181</td>
<td>194</td>
<td>173</td>
<td>173</td>
<td>0</td>
</tr>
<tr>
<td>Mean Scaffold Size</td>
<td>119470</td>
<td>118349</td>
<td>137760</td>
<td>224099</td>
<td>9327</td>
</tr>
<tr>
<td>Median Scaffold Size</td>
<td>11271</td>
<td>11074</td>
<td>15163</td>
<td>48524</td>
<td>32761</td>
</tr>
<tr>
<td>N50 Scaffold Length</td>
<td>965264</td>
<td>1045589</td>
<td>794060</td>
<td>838530</td>
<td>23355</td>
</tr>
<tr>
<td>L50 Scaffold Count</td>
<td>192</td>
<td>181</td>
<td>242</td>
<td>229</td>
<td>196</td>
</tr>
<tr>
<td>N50 Scaffold – NG50 Scaffold</td>
<td>1139066</td>
<td>1445221</td>
<td>816720</td>
<td>816720</td>
<td>23355</td>
</tr>
<tr>
<td></td>
<td>Length Difference</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NG50 Scaffold Length</td>
<td>146 116 234 234</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LG50 Scaffold Count</td>
<td>173802 399632 22660 21810</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaffold %A</td>
<td>17.6 17.57 16.78 16.43 29.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaffold %C</td>
<td>8.67 8.66 8.28 8.08 15.14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaffold %G</td>
<td>8.69 8.68 8.58 8.09 15.07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaffold %T</td>
<td>17.6 17.57 16.79 16.45 29.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaffold %N</td>
<td>47.44 47.53 49.88 50.94 11.69</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaffold N bp</td>
<td>355201134 390307384 387704986 2494317</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scaffold % non-ACGTN bp</td>
<td>0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of Scaffold non-ACGTN bp</td>
<td>0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% of Assembly in Scaffolded Contigs</td>
<td>98.1 98.0 98.6 99.3 72.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>% of Assembly in Unscaffolded Contigs</td>
<td>1.9 2.0 1.4 0.7 27.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Number of Contigs per Scaffold</td>
<td>4.6 4.6 5.1 7.4 1.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Length of Break</td>
<td>56682 15777 16862 17878 1702</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(&gt;25Ns) Between Contigs in Scaffold</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>Number of Contigs</td>
<td>28743</td>
<td>28818</td>
<td>28817</td>
<td>25073</td>
<td>3744</td>
</tr>
<tr>
<td>Number of Contigs in Scaffolds</td>
<td>26417</td>
<td>26453</td>
<td>26961</td>
<td>24656</td>
<td>2303</td>
</tr>
<tr>
<td>Number of Contigs not in Scaffolds</td>
<td>2326</td>
<td>2365</td>
<td>1856</td>
<td>417</td>
<td>1441</td>
</tr>
<tr>
<td>Total Size of Contigs</td>
<td>392342356</td>
<td>392340764</td>
<td>392340765</td>
<td>373489557</td>
<td>18851208</td>
</tr>
<tr>
<td>Longest Contig</td>
<td>551924</td>
<td>551924</td>
<td>551924</td>
<td>551924</td>
<td>197784</td>
</tr>
<tr>
<td>Shortest Contig</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>111</td>
</tr>
<tr>
<td>Number of Contigs &gt; 500 bp</td>
<td>28135</td>
<td>28186</td>
<td>28185</td>
<td>24511</td>
<td>3674</td>
</tr>
<tr>
<td>Number of Contigs &gt; 1k bp</td>
<td>26433</td>
<td>26481</td>
<td>26480</td>
<td>23063</td>
<td>3417</td>
</tr>
<tr>
<td>Number of Contigs &gt; 10k bp</td>
<td>927</td>
<td>9134</td>
<td>91387</td>
<td>8737</td>
<td>397</td>
</tr>
<tr>
<td>Number of Contigs &gt; 100k bp</td>
<td>533</td>
<td>531</td>
<td>531</td>
<td>524</td>
<td>7</td>
</tr>
<tr>
<td>Number of Contigs &gt; 1M bp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean Contig Size</td>
<td>13650</td>
<td>13614</td>
<td>13615</td>
<td>14896</td>
<td>5035</td>
</tr>
<tr>
<td>Median Contig Size</td>
<td>4622</td>
<td>4615</td>
<td>4615</td>
<td>5409</td>
<td>1837</td>
</tr>
<tr>
<td>N50 Contig Length</td>
<td>39176</td>
<td>39020</td>
<td>390210</td>
<td>40575</td>
<td>12402</td>
</tr>
<tr>
<td>L50 Contig Count</td>
<td>2444</td>
<td>2453</td>
<td>2453</td>
<td>2256</td>
<td>315</td>
</tr>
<tr>
<td>N50 Contig – NG50 Contig</td>
<td>28590</td>
<td>23057</td>
<td>37499</td>
<td>40575</td>
<td>12402</td>
</tr>
<tr>
<td>Length Difference</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>Contig %A</td>
<td>33.47</td>
<td>33.47</td>
<td>33.46</td>
<td>33.49</td>
<td>32.88</td>
</tr>
<tr>
<td>Contig %C</td>
<td>16.49</td>
<td>16.49</td>
<td>16.51</td>
<td>16.47</td>
<td>17.13</td>
</tr>
<tr>
<td>Contig %G</td>
<td>16.53</td>
<td>16.53</td>
<td>16.51</td>
<td>16.48</td>
<td>17.05</td>
</tr>
<tr>
<td>Contig %T</td>
<td>33.47</td>
<td>33.47</td>
<td>33.49</td>
<td>33.52</td>
<td>32.87</td>
</tr>
<tr>
<td>Contig %N</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.07</td>
</tr>
<tr>
<td>Contig N bp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>168936</td>
<td>168937</td>
<td>155351</td>
<td>13586</td>
<td></td>
</tr>
<tr>
<td>Contig % non-ACGTN</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Number of Contig non-ACGTN bp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Number of Gaps (25+ N)</td>
<td>22497</td>
<td>22428</td>
<td>23057</td>
<td>21602</td>
<td>1455</td>
</tr>
<tr>
<td>Longest Gap (bp)</td>
<td>7025289</td>
<td>3183096</td>
<td>4876029</td>
<td>4876029</td>
<td>6537</td>
</tr>
<tr>
<td>Mean Gap Length (bp)</td>
<td>15730</td>
<td>15830</td>
<td>16916</td>
<td>17940</td>
<td>1705</td>
</tr>
<tr>
<td>Mean Gaps per Gapped Scaffold</td>
<td>5.739031</td>
<td>5.682291</td>
<td>6.035864</td>
<td>7.261177</td>
<td>1.721894</td>
</tr>
<tr>
<td>Number of CEGs Present in a Complete Form</td>
<td>218</td>
<td>217</td>
<td>218</td>
<td>215</td>
<td>9</td>
</tr>
<tr>
<td>Number of CEGs Present in a</td>
<td>240</td>
<td>240</td>
<td>240</td>
<td>237</td>
<td>10</td>
</tr>
<tr>
<td>Partial Form</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>Mean Number of Orthologs per Complete CEG</td>
<td>1.66</td>
<td>1.66</td>
<td>1.66</td>
<td>1.64</td>
<td>1.22</td>
</tr>
<tr>
<td>Mean Number of Orthologs per Partial CEG</td>
<td>1.88</td>
<td>1.87</td>
<td>1.88</td>
<td>1.86</td>
<td>1.2</td>
</tr>
<tr>
<td>Percentage of Detected Complete CEGs with More</td>
<td>39.45</td>
<td>39.17</td>
<td>39.91</td>
<td>40.00</td>
<td>22.22</td>
</tr>
<tr>
<td>than one Ortholog</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Percentage of Detected Partial CEGs with More</td>
<td>51.25</td>
<td>50.83</td>
<td>51.67</td>
<td>51.48</td>
<td>20.00</td>
</tr>
<tr>
<td>than one Ortholog</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The first OAC-Rex pseudochromosome (v2) dataset comprised 6246 scaffolds totalling ~746 Mbp (103.6% of the estimated size). The scaffolds measured 890–23,081,690 bp in length, with a mean of 119,470 bp and N50 of 965,264 bp. 47.44% of the bases were N as the set contained 22,479 gaps up to 7,025,289 bp in length. These scaffolds contained 26,417 contigs with another 2,326 (~9%) unused in scaffolds. There were 218 CEGs identified in a complete form and 240 in at least a partial form via ‘CEGMA’.

Version 3.3 increased the scaffold count to 6,315 while also increasing the total size by 0.2% of the estimated genome to ~747 Mbp. The longest scaffold was 10,711,518 bp (46% the length of version 2) and the shortest was still 890 bp, moving the mean scaffold size to 118,349 and the N50 to 1,045,589 bp. This version incorporated 36 more scaffolds than the previous, but there were 75 more scaffolds (2,8818) in total, of which 39 more remained unincorporated in scaffolds (2,365 in total). There were 22,428 gaps, the longest being 3183096 bp, reporting 47.53% of the assembly as N. 240 of the CEGs were detected as at least partially present 217 of which were complete.

Version 4 consisted of 5,680 scaffolds, 2,287 of which were unanchored, with ~761 Mbp anchored and another ~21 Mbp unanchored, it spans 108.7% of the estimated genome size. The longest scaffold, 5,794,880 bp, was anchored while the shortest, 927 bp, was unanchored leaving a N50 of 794,060 bp and mean of 137,760 bp. Of the 28,817 contigs 1,856 were unable to be used in scaffolds while ‘CEGMA’ was able to identify 218 complete CEGs and another 22 in a partial form. 49.88% of the assembly was N, collected in 23,057 gaps spanning up to 4,876,029 bp.

**PI440795**

The PI440795 assembly differed significantly from that of OAC-Rex. The first contig assembly was a ‘Ray’ k41 assembly, like OAC-Rex, but lacking any MP data prevented scaffolding. Unlike OAC-Rex, there was PacBio® long read data to generate a second contig set using the ‘CANU’ software. These two contig sets were submitted to the ‘GARM’ assembly reconciliation software to generate a merged contig set, which was in turn used with the same *P. vulgaris* SNP dataset as OAC-Rex to generate PI440795 Pseudochromosomes. Each stage of this assembly is detailed in Table 3.9.
Table 3.9 – Statistics of PI440795 Genome Assemblies

‘Assemblathon_stats’ was used to gather structural statistics, gap statistics were collected using the ‘Fasta_gap_analyzer’ custom code (described in Appendix II), and CEG statistics were collected using ‘CEGMA’ from multiple PI440795 sequence assemblies.

<table>
<thead>
<tr>
<th>Sample</th>
<th>‘Ray’ k41 Contigs</th>
<th>‘CANU’ Contigs</th>
<th>‘GARM’ Contigs</th>
<th>Pseudochromosomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Scaffolds</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7484</td>
</tr>
<tr>
<td>Total size of Scaffolds</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>819271488</td>
</tr>
<tr>
<td>Total Scaffold Length as % of Estimated Genome</td>
<td>69.7</td>
<td>63.2</td>
<td>58.9</td>
<td>126.6</td>
</tr>
<tr>
<td>Longest Scaffold</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>14790416</td>
</tr>
<tr>
<td>Shortest Scaffold</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1065</td>
</tr>
<tr>
<td>Number of Scaffolds &gt; 500 bp</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7484 (100.0%)</td>
</tr>
<tr>
<td>Number of Scaffolds &gt; 1k bp</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>7484 (100.0%)</td>
</tr>
<tr>
<td>Number of Scaffolds &gt; 10k bp</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>6578 (87.9%)</td>
</tr>
<tr>
<td>Number of Scaffolds &gt; 100k bp</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>781 (10.4%)</td>
</tr>
<tr>
<td>Number of Scaffolds &gt; 1M bp</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>142 (1.9%)</td>
</tr>
<tr>
<td>Mean Scaffold Size</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>109470</td>
</tr>
<tr>
<td>Median Scaffold Size</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>25945</td>
</tr>
<tr>
<td>N50 Scaffold Length</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1919016</td>
</tr>
<tr>
<td>L50 Scaffold Count</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>92</td>
</tr>
<tr>
<td>N50 Scaffold – NG50 Scaffold Length Difference</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>22042</td>
</tr>
<tr>
<td></td>
<td>NG50 Scaffold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NG50 Scaffold Length</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total Size of Contigs</td>
<td>Longest Contig</td>
<td>Shortest Contig</td>
<td>Number of Contigs &gt; 500 bp</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------</td>
<td>----------------</td>
<td>----------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Total Size of Contigs</td>
<td>451369444</td>
<td>409400810</td>
<td>381156496</td>
<td>381156496</td>
</tr>
<tr>
<td>Longest Contig</td>
<td>230244</td>
<td>300321</td>
<td>369713</td>
<td>639713</td>
</tr>
<tr>
<td>Shortest Contig</td>
<td>100</td>
<td>1002</td>
<td>1065</td>
<td>1065</td>
</tr>
<tr>
<td>Number of Contigs &gt; 500 bp</td>
<td>36159(7.7%)</td>
<td>18142(100.0%)</td>
<td>8341(100.0%)</td>
<td>8341(100.0%)</td>
</tr>
<tr>
<td>Number of Contigs &gt; 1k bp</td>
<td>33540(7.1%)</td>
<td>18142(100.0%)</td>
<td>8341(100.0%)</td>
<td>8341(100.0%)</td>
</tr>
<tr>
<td>Number of Contigs &gt; 10k bp</td>
<td>11423(2.4%)</td>
<td>12027(66.3%)</td>
<td>7421(89.0%)</td>
<td>7421(89.0%)</td>
</tr>
<tr>
<td>Number of Contigs &gt; 100k bp</td>
<td>104(0.0%)</td>
<td>298(1.6%)</td>
<td>893(10.7%)</td>
<td>893(10.7%)</td>
</tr>
<tr>
<td>Number of Contigs &gt; 1M bp</td>
<td>0(0.0%)</td>
<td>0(0.0%)</td>
<td>0(0.0%)</td>
<td>0(0.0%)</td>
</tr>
<tr>
<td>Mean Contig Size</td>
<td>962</td>
<td>22566</td>
<td>45697</td>
<td>45697</td>
</tr>
<tr>
<td>Median Contig Size</td>
<td>147</td>
<td>14415</td>
<td>28160</td>
<td>28160</td>
</tr>
<tr>
<td>N50 Contig Length</td>
<td>17987</td>
<td>34740</td>
<td>73226</td>
<td>73226</td>
</tr>
<tr>
<td>L50 Contig Count</td>
<td>6342</td>
<td>3346</td>
<td>1434</td>
<td>1434</td>
</tr>
<tr>
<td>N50 Contig – NG50 Contig Length Difference</td>
<td>11316</td>
<td>26682</td>
<td>73226</td>
<td>73226</td>
</tr>
<tr>
<td>Contig %A</td>
<td>33.71</td>
<td>33.19</td>
<td>33.46</td>
<td>33.45</td>
</tr>
<tr>
<td>Contig %C</td>
<td>16.37</td>
<td>16.79</td>
<td>16.54</td>
<td>16.55</td>
</tr>
<tr>
<td>Contig %G</td>
<td>16.34</td>
<td>16.82</td>
<td>16.55</td>
<td>16.55</td>
</tr>
<tr>
<td>Contig %T</td>
<td>33.58</td>
<td>33.20</td>
<td>33.45</td>
<td>33.46</td>
</tr>
<tr>
<td>Contig %N</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Contig N bp</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Contig % non-ACGTN</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Number of Contig</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>non-ACGTN bp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Number of Gaps (25+ N)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>857</td>
</tr>
<tr>
<td>Longest Gap (bp)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13015863</td>
</tr>
<tr>
<td>Mean Gap Length (bp)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>511219</td>
</tr>
<tr>
<td>Mean Gaps per Gapped Scaffold</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3.028269</td>
</tr>
<tr>
<td>Number of CEGs Present in a Complete Form</td>
<td>214</td>
<td>196</td>
<td>195</td>
<td>197</td>
</tr>
<tr>
<td>Number of CEGs Present in a Partial Form</td>
<td>240</td>
<td>227</td>
<td>223</td>
<td>221</td>
</tr>
<tr>
<td>Mean Number of Orthologs per Complete CEG</td>
<td>1.83</td>
<td>1.69</td>
<td>1.69</td>
<td>1.65</td>
</tr>
<tr>
<td>Mean Number of Orthologs per Partial CEG</td>
<td>2.14</td>
<td>2.01</td>
<td>1.91</td>
<td>1.86</td>
</tr>
<tr>
<td>Percentage of Detected Complete CEGs with More than one Ortholog</td>
<td>48.13</td>
<td>38.27</td>
<td>44.10</td>
<td>43.15</td>
</tr>
<tr>
<td>Percentage of Detected Partial CEGs with More than one Ortholog</td>
<td>59.58</td>
<td>54.63</td>
<td>53.81</td>
<td>51.13</td>
</tr>
</tbody>
</table>
‘Ray’ generated 469,188 contigs for PI440795 which spanned 100–230,244 bp, totalling ~451 Mbp or 69.7% of the *P. acutifolius* estimated genome size. The mean contig length was 962 bp with an N50 of 17,987 bp. They contained no N and thus no gaps, and ‘CEGMA’ found 240 CEGs, 214 of which were in a complete form.

The PI440795 long-reads assembled by ‘CANU’ also contained zero N and thus no gaps. These 18142 contigs covered ~409 Mbp (63.2%) ranging in size from 1,002 bp up to 300,321 bp within which 196 CEGs were identified as complete and an additional 31 as partial. The N50 was measured at 34,740 bp and 22,566 bp was reported as the mean length.

Once the two contig sets were reconciled via ‘GARM’ they still contained no gaps as there were no N bases. They had 195 complete CEGs with 223 detected in total. The ~381 Mbp assembly (58.9%) ranged from as long as 369,713 bp or as short as 1,065 bp, averaging 45,697 bp, across 8341 reads with an N50 of 73,226 bp.

**Synteny**

An understanding of synteny between the introgressed offspring and each parent is likely to suggest, or help identify, regions that may have been subjected to introgression. The ‘Mauve’ alignment software was given each matching chromosome/pseudochromosome pair to investigate the relatedness of the subjects. Once a new version of the Pseudochromosome dataset was constructed it was aligned against its predecessor to attempt to detect any major reorganizations that may have occurred.
In each, colour coded segments represent syntenic blocks with a line connecting the corresponding blocks. The general order of synteny is maintained from G-19833 v1 (top) but additional unmatching DNA, denoted by white sections within the coloured blocks, is expanding the size of the blocks in OAC-Rex v2 (bottom). The red lines denote a new entry in the supplied fasta files. A) Chromosome 4 alignments show smaller block size, though higher count with more rearrangement. B) Chromosomes 8 shows larger blocks with highly conserved syntenic order. C) All pseudochromosomes without unanchored scaffolds are shown, displaying a notable gap in synteny in the largest pseudochromosome and a large amount of apparent rearrangement across the genome.

Figure 3.1, along with Figure D.1 through Figure D.9 (Appendix D), display the syntenic alignment of G-19833 v1 and OAC-Rex Pseudochromosomes v2. Each represents a single chromosome, focussing on chromosome 4 (Figure 3.1(a)) and chromosome 8 (Figure 3.1(b)) represents a single chromosome, while Figure 3.1(c) shows all 11. In each individual chromosome the general order of the syntenic blocks is maintained, though there are some exceptions. The discontinuous blocks are distributed as follows: chromosome 1 has 7, chromosome 2 has 0, chromosome 3 has 13, chromosomes 4 has 12, chromosome 5 has 4, chromosome 6 has 10, chromosome 7 has 15, chromosome 8 and 9 each have 1, chromosome 10 has 10, and chromosome 11 has
13. While chromosome 11 shows the largest single syntenic block, chromosome 9 has the largest average block size and chromosome 4 has the smallest average, based on a visual inspection. There is also a notable gap positioned to the left end of chromosome 6 as well as a much larger gap central to chromosome 7 that corresponds to the largest single pseudochromosome of the v2 assembly. The scaffold 7 gap is large enough to be readily identifiable on the whole genome alignment of Figure 3.1(c). It is important to note that none of these alignments include unanchored scaffolds.

![Figure 3.2 – Syntenic Alignments involving PI440795 ‘Ray’ Generated Contigs via ‘Mauve’](image)

In each, colour coded segments represent syntenic blocks with a line connecting the corresponding blocks and red lines denoting a new entry in the supplied fasta files. **A** G-19833 v1 (top) and PI440795 ‘Ray’ generated contigs (bottom). **B** OAC-Rex pseudochromosomes v2 (top) and PI440795 ‘Ray’ generated contigs (bottom). **C** G-19833 v1 (top), OAC-Rex pseudochromosomes v2 (middle), or PI440795 ‘Ray’ generated contigs (bottom). In all three the number of blocks identified creates enough noise to visually mask any clear signal.
G-19833 v1 alignment to PI440795 ‘Ray’ generated contigs were generated with the aim of identifying synteny via ‘Mauve’. The produced graphic contained so many alignments of such small scale that no single match could be visually identified. Aligning PI440795 ‘Ray’ generated contigs to OAC-Rex v2 generated a high volume of low length reads, to the point that no single alignment could be detected, though a notable gap of alignment was apparent in the region that corresponded to the longest pseudochromosome, denoted by the largest space between red lines, found on chromosome 7. Attempting to align G-19833 v1, OAC-Rex v2, and PI440795 ‘Ray’ generated contigs revealed that any alignments would be of so small a scale and so high in count as to cover any further interpretable visible information.

Figure 3.3 – Syntenic Alignments of OAC-Rex Pseudochromosomes v2 and OAC-Rex Pseudochromosomes v3.3 via ‘Mauve’

In both, colour coded segments represent syntenic blocks along with a similarly coloured line connecting the corresponding blocks, with unmatching DNA denoted by white sections within the coloured blocks, while vertical red lines denote a new entry in the supplied fasta files. A) OAC-Rex Pseudochromosomes v3.3 (top) are aligned against OAC-Rex Pseudochromosomes v2 (bottom). Both were single-entry compressed for each chromosome and unanchored scaffolds were removed before alignment. While some rearrangement is detected, it is minimal across the genome, though there are multiple regions that do not display synteny. B) OAC-Rex Pseudochromosomes v2 (top) aligned to OAC-Rex Pseudochromosomes v3.3 (bottom) without single-entry compression and including unanchored scaffolds. Red break lines obscure much of the blocks themselves, but the connecting lines reveal a relatively large proportion of unanchored scaffolds matching to pseudochromosomes.

Upon generation of version 3.3 of OAC-Rex pseudochromosomes they were aligned against version 2. Initially without including unanchored scaffolds and after using
‘Fasta_single_entry_compressor’ to compress all pseudochromosomes into a single entry for each chromosome, inserting 25 x N (a detectable gap) between each read (Figure 3.3(a)). A visual inspection shows 23 discontinuous blocks between the assemblies and 17 blocks that align to the opposite strand. The general order is preserved, particularly the terminus of chromosome 8 and the entirety of chromosome 9 in the largest single syntenic block, every other chromosome has at least one region that was not aligned.

A second alignment included the unanchored scaffolds and did not concatenate the pseudochromosomes (Figure 3.3(b)). The syntenic blocks are largely hidden behind the red markers identifying a new entry in the supplied fasta. Even still, the connecting lines between the blocks still show a largely maintained arrangement of syntenic blocks with most of the discontinuous blocks appearing to align to unanchored reads in either assembly, though predominantly to those of version 3.3. Of note is the reduction in size of the largest pseudochromosome previously identified in Figure D.6 (see Appendix D) as being on chromosome 7.

Figure 3.4 – Syntenic Alignments of OAC-Rex Pseudochromosomes v3.3 and OAC-Rex Pseudochromosomes v4 via ‘Mauve’

In both, blocks of shared synteny between OAC-Rex pseudochromosomes v4 (bottom) and OAC-Rex pseudochromosomes v3.3 (top) are denoted by colour coded segments with a line connecting the corresponding blocks, with unmatched DNA denoted by white sections within the coloured blocks. The red lines denote a new entry in the supplied fasta files. In a) pseudochromosomes have been single-entry compressed per chromosome and unanchored scaffolds have been removed before alignment, while b) is unfiltered and not compressed.

OAC-Rex pseudochromosomes v4 had a drastically reduced stringency for
alignment of the anchoring SNPs to the ‘AllPaths-LG’ scaffolds. These were aligned against v3.3 to examine the impact of this reduction on the arrangement of the scaffolds within the pseudochromosomes. Examining the ‘Fasta_single_entry_compressor’ compressed assemblies (Figure 3.4(a)) revealed the syntenic blocks to be much smaller than those between version 3.3 and version 2 (Figure 3.3(a)). These blocks also show much greater discontinuity than any other alignment not involving ‘Ray’ contigs except perhaps that of the whole genomes of OAC-Rex v2 and G-19833 v1 (Figure 3.1(c)). Many regions of version 4 do not appear to have a corresponding alignment in v3.3, which coincides with the much larger size of v4; the last two chromosomes of v4 extend beyond the last chromosome of v3.3.

When the assemblies are compared including their unanchored scaffolds the degree to which v4 extends beyond v3 is greatly reduced, though v4 remains larger. The discontinuity intensifies as well, though most of the discontinuous matches appear to be coming from the v3.3 unanchored scaffolds to the assembled v4 pseudochromosomes.

‘SyMAP’

While ‘Mauve’ detects synteny between submitted read sets, it does not distinguish between reads. Multi-read fastas have their reads concatenated to each other (denoted by a red line in the output image), which can create a syntenic block that spans across different chromosomes. Individual chromosomes can be run, but will require the generation of one alignment for each chromosome, as shown in Figure 3.1(a), Figure 3.1(b), and Figure D.1 through Figure D.9. When combined for a whole genome the earlier identified blocks are not faithfully recreated, Figure 3.1(c). Upon generation of the third version of OAC-Rex’s pseudochromosomes a different syntenic mapping tool, ‘SyMAP’, was used to examine the genome-wide synteny. This tool parses out every chromosome, scaffold, or contig (as defined by the user) before aligning the second assembly against each parsed segment of the first. It then recombines the alignments into a whole genome representation of the region specific synteny identified. ‘SyMAP’ was used to align OAC-Rex to G-19833, PI440795, and to itself for an analysis of chromosome-scale synteny.
Figure 3.5 – Genome-wide Syntenic Alignments of Single-entry Compressed OAC-Rex Pseudochromosomes and G-19833 v2 via ‘SyMAP’

Each colour denotes a single entry supplied to SyMAP, in this instance a chromosome from G-19833 v2 (bottom) and single-entry compressed chromosomes from OAC-Rex Pseudochromosomes v2 (a), OAC-Rex Pseudochromosomes v3.3 (b), and OAC-Rex Pseudochromosomes v4 (c), consistently on top. The bands represent what regions of the sequences showed synteny.

Single-entry compressed OAC-Rex pseudochromosomes v2 were aligned against G-19833 v2 to establish a baseline synteny profile against which subsequent OAC-Rex pseudochromosome assemblies could be compared (Figure 3.5(a)). For every chromosome the general synteny is conserved between G-19833 and OAC-Rex with 7 identified regions that align with different chromosomes than the anchoring assigned
them to. While chromosome 6 initially appears to be missing any alignment, save near its terminal end, a change in shading near the central overlap both above and below indicated there is a white band that connects the two. Importantly, the terminus of chromosome 8 in OAC-Rex shows a short match to chromosome 6, this corresponds to a known duplication at those locations, however, rearrangements are entirely lacking in chromosome 4, a region known to have been largely reorganized (Perry, G., Personal Communication).

Next, OAC-Rex pseudochromosomes v3.3 were single-entry compressed then submitted to SyMAP for alignment against G-19833 v2 (Figure 3.5(b)). Overall there is more synteny detected between the chromosomes than those of OAC-Rex Pseudochromosomes v2 (Figure 3.5(a)) and for each chromosome the general synteny between chromosomes is clearly shown. However, chromosome 4 shows only a small region at its terminus that aligns elsewhere to G-19833 and a region in chromosome 6 again aligns to the terminus of chromosome 8.

**Figure 3.5(c)** shows reduced stringency of OAC-Rex pseudochromosomes v4 generated a much more widespread synteny detected in G-19833 v2. Every chromosome showed a larger amount of synteny than previous versions of OAC-Rex pseudochromosomes (Figure 3.5(a) and Figure 3.5(b)). While chromosomes 6, 7, and 8 showed proportionately less synteny than their counterparts, chromosome 7 also showed a gap in alignment to a central region of the single-entry pseudochromosome, and chromosome 6 had a section that aligned to the terminus of chromosome 8, all of which correlates to the established understanding of OAC-Rex’s genome (Figure D.6 and Perry, G., Personal Communication).

**Annotation**

Assembly optimization focussed primarily on OAC-Rex. To gain an understanding of the functional content of each assembly generated an annotation was generated to identify an array of genome content (Table 3.10). These annotations were of greatest use when co-visualized via a gene browser. The same annotation pipeline was used for PI440795.
Table 3.10 – Annotation Statistics of OAC-Rex Assemblies and PI440795 Pseudochromosomes

Annotations were done via 'MAKER', 'LTRHarvest', 'LTR_finder', 'TRF', and 'tRNAscan-SE', while values were collected using the 'Gff_stat_parser' custom code (see Appendix II)

<table>
<thead>
<tr>
<th>Source</th>
<th>OAC-Rex ‘Ray’ k41 Contigs</th>
<th>OAC-Rex ‘Ray’ + ‘SOAPdenovo’</th>
<th>OAC-Rex ‘AllPaths-LG’</th>
<th>OAC-Rex PChr v2</th>
<th>OAC-Rex PChr v3.3</th>
<th>OAC-Rex PChr v4</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘MAKER’ Consensus Genes</td>
<td>28470</td>
<td>25317</td>
<td>26435</td>
<td>26435</td>
<td>27249</td>
<td>28706</td>
</tr>
<tr>
<td>‘Augustus’ Matches</td>
<td>78520</td>
<td>71487</td>
<td>63750</td>
<td>63750</td>
<td>64116</td>
<td>63371</td>
</tr>
<tr>
<td>‘SNAP’ Matches</td>
<td>22507</td>
<td>26296</td>
<td>25782</td>
<td>25782</td>
<td>27763</td>
<td>25482</td>
</tr>
<tr>
<td>‘Genemark’ Matches</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>‘BLASTn’ Matches</td>
<td>11002</td>
<td>110629</td>
<td>51380</td>
<td>51380</td>
<td>11399</td>
<td>11583</td>
</tr>
<tr>
<td>‘BLASTx’ Matches</td>
<td>368544</td>
<td>373307</td>
<td>365261</td>
<td>365261</td>
<td>352705</td>
<td>371128</td>
</tr>
<tr>
<td>‘tBLASTx’ Matches</td>
<td>902425</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>930749</td>
<td>1041978</td>
</tr>
<tr>
<td>‘est2genome’ Matches</td>
<td>11553</td>
<td>136229</td>
<td>62928</td>
<td>62928</td>
<td>12080</td>
<td>12288</td>
</tr>
<tr>
<td>‘protein2genome’ Matches</td>
<td>272250</td>
<td>268071</td>
<td>264076</td>
<td>264076</td>
<td>261999</td>
<td>266388</td>
</tr>
<tr>
<td>‘cdna2genome’ Matches</td>
<td>889763</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>976340</td>
<td>1102700</td>
</tr>
<tr>
<td>‘RepeatMasker’ Matches</td>
<td>380603</td>
<td>506103</td>
<td>268623</td>
<td>268623</td>
<td>287070</td>
<td>217200</td>
</tr>
<tr>
<td>‘RepeatRunner’ Matches</td>
<td>1442</td>
<td>13808</td>
<td>11229</td>
<td>11229</td>
<td>1741</td>
<td>30073</td>
</tr>
<tr>
<td>Matches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>‘LTRHarvest’ Matches</td>
<td>-</td>
<td>8574</td>
<td>7825</td>
<td>7186</td>
<td>-</td>
<td>7918</td>
</tr>
<tr>
<td>‘LTR_finder’ Matches</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>894</td>
<td>-</td>
<td>883</td>
</tr>
<tr>
<td>‘TRF’ Matches</td>
<td>-</td>
<td>221215</td>
<td>197373</td>
<td>176369</td>
<td>-</td>
<td>171054</td>
</tr>
<tr>
<td>‘tRNAscan-SE’ Matches</td>
<td>-</td>
<td>773</td>
<td>732</td>
<td>711</td>
<td>-</td>
<td>710</td>
</tr>
<tr>
<td>Genes</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Transposable Elements</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Annotation of every assembly was standardized since the annotations made were going to be used for a comparison of the functionality of the output from each assembler. The pseudochromosomes v2 dataset was not annotated independently via ‘MAKER’, it instead used the values from ‘AllPaths-LG’ with updated locations to correspond to the anchored positions of the scaffolds. Notably, ‘Genemark’ failed to return a single match to any assembly and the ‘Ray’ + ‘SOAPdenovo’ and ‘AllPaths-LG’ assemblies, and thus ‘OAC-Rex v2’, had no hits for ‘tBLASTx’ but did record results for ‘BLASTn’ and ‘BLASTx’, which were components of the same installation package. The same is true for ‘cdna2genome’ and ‘protein2genome’; they came from the same installation package, but one failed to report any hits. Overall, there was no apparent patterning to the number of hits from the component algorithms and the consensus genes identified by ‘MAKER’, some of the values for a single algorithm varied by as much as an order of magnitude.

The non-‘MAKER’ algorithms did report a general pattern of recording more hits from the ‘Ray’ + ‘SOAPdenovo’ assembly and decreasing with each new assembly annotated, except for ‘LTRHarvest’ which reported the second most on pseudochromosomes v4. These algorithms were not in place when the ‘Ray’ contigs were first generated and once they had been implemented the k41 contigs had been decided on as the base. Pseudochromosomes v3.3 are also lacking in any values from these algorithms since the generation of pseudochromosomes v4 had begun while their predecessor was being annotated by ‘MAKER’.

The final two entries on this table account for generic entries in the gffs analysed. Should a source not be identified the feature may be given a broad descriptor. These two values act as verification that all elements are being appropriately associated to their source.

POI Identification

POI Analysis

An in-depth analysis of introgression was performed once the OAC-Rex and PI440795 assemblies were at a mature and annotated stage. Chromosomes 8 and 4 were used given the presence of known CBB resistance markers; however, the focus was on
chromosome 8 as it exhibited minimal identifiable rearrangement following synteny analyses using ‘Mauve’ or ‘SyMAP’.

‘BLASTn’ was used to align chromosomes 8 from OAC-Rex pseudochromosomes v2 against chromosome 8 from G-19833 v1 and PI440795 ‘Ray’ k41 contigs. Unique matches between OAC-Rex and one parent were identified and the shortest distance between sequence of differing unique parentage were extracted to identify nine zones of introgression. To analyse primary sequence synteny these nine were submitted to ‘Mauve’.

![Figure 3.6 – ‘Mauve’ Alignment of Initial Points of Introgression in Chromosome 8 from OAC-Rex Pseudochromosomes v2](image)

Blocks of shared synteny between putative POIs in chromosome 8 of OAC-Rex pseudochromosomes v2 are denoted by colour coded segments with a line connecting the corresponding blocks, with unmatched DNA denoted by white sections within the coloured blocks. The red lines denote a new entry in the supplied fasta files.

The length of the identified putative POIs ranged drastically; the shortest was less than 1,000 bp and the longest was ~90 kbp. Following the lines connecting the syntenic blocks one can see there is no block shared by more than 3 putative POIs, though the second putative POI appears to have the most syntenic blocks. These blocks are notably smaller than those identified in chromosome-wide alignments of OAC-Rex pseudochromosomes v2 shown in the various images of Appendix D. It is also
noteworthy that the blocks identified are far more often white than coloured, denoting unmatched DNA within the block.

A finer scale analysis was performed on the terminal 5 Mbp of Chr08 via RDP, where investigation of gene annotations identified as Niemann-Pick genes had identified a strong candidate for an introgressed CBB resistance marker. A rolling window approach of 1 Mbp with a 500 kbp overlap was used, with the ‘Fasta_range_subset_extractor’ custom code (see Appendix II) to generate the corresponding sequences from OAC-Rex, G-19833, and PI440795. These sequences were paired across organisms and aligned against each other using ‘MAFFT’, which identified clear signals in every individual window. These windows were then aligned to each other (**Figure 3.7**).

**Figure 3.7** – Alignment of the Terminal 2 Mbp Sequence Ranges of Chromosome 8 in OAC-Rex Pseudochromosomes v4 via ‘RDP4’

Gathered from a rolling window alignment of 1 Mbp stretches with 500 kbp overlap. The three windows are aligned such that the X-axis represents position along the chromosome with the right-most position of the bottom graph being the terminus of chromosome 8. A purple background denotes a lack of informative signal, as opposed to a grey background, the yellow line indicated the pairwise alignment of G-19833 to PI440795, a purple line the pairwise alignment of OAC-Rex to G-19833, and a teal line the pairwise alignment of OAC-Rex to PI440795.

From this alignment to each other in ‘RDP4’ the recombination signals detected by the RDP method could be seen. The clearest signal was seen in the middle window,
with two regions that showed changing signals. The left such region aligned clearly with
the only region in the top window identified as having informative signal, however the
signals did not match between windows. The right region of the middle window did not
have a match in the bottom window, whose signals did not have any corresponding
matches in the middle window.

POI Dot Plots

‘AllPaths-LG’ scaffolds 600, 570, and 1970, along with the corresponding regions
of G-19833 and PI440795, were used for a fine-scale study of introgression, since these
scaffolds were known to carry CBB-associated markers (PvCTT001) and Niemann-Pick
like genes, respectively. These ranges were broken into 100-bp chunks using
‘Fasta_chunker’ (see Appendix II), then aligned against each other using ‘BLASTn’. The
resulting alignment scores were organized based on the nucleotide position covered, with
an alignment to only one parent assigned a ratio of either a 0 or a 2 depending on the
lacking parent, and the ratios of OAC-Rex/G-19833 to OAC-Rex/PI440795 were plotted
for each scaffold. These ratios were then filtered to remove any scores that varied less
than 1 standard deviation from a ratio of 1.
**Scaffold_600 Parentage**

*Figure 3.8 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Parentage*

Dot plots showing parentage scores of 100 bp regions of scaffold_600 from the ‘AllPaths-LG’ generated scaffold set. Shown in *a*) is the unfiltered distribution of parentage, *b*) shows the same dataset filtered to remove any points that vary by less than 1 stdev from 1. The X-axis represents the nucleotide position along the chromosome and the Y-axis shows the parentage score.

Scaffold_600, anchored to chromosome 4, was identified as matching the PvCTT001 marker. This scaffold showed a majority of the parentage scores near the 1 ratio, though many were also assigned 2 and some a 0. Once filtered, however, the bulk of the remaining points have a score of 2, often in large clusters, though six points above 1.464 were not at 2. The distribution of points below 0.536 (1 – 1 stdev) more often sit at 0, sometimes in small clusters, though many sit visibly above the 0 line.
Figure 3.9 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Parentage

Dot plots showing parentage scores of 100 bp regions of scaffold_570 from the ‘AllPaths-LG’ generated scaffold set. Shown in a) is the unfiltered distribution of parentage, while b) shows the same dataset filtered to remove any points that vary by less than 1 stdev from 1. The X-axis represents the nucleotide position along the chromosome and the Y-axis shows the parentage score.

Plotting parentage of scaffold_570 showed more than half of the scaffold uniquely matched to PI440795, with several clusters sitting near the 1 line in the last quarter of the graph. There are also two sharp gaps in alignments to G-19833, the first is again uniquely PI440795, the second lack any parental scores at all. Removing the points of ‘shared’ parentage it becomes apparent only four points have a score of 2, the rest above 1 are loosely distributed. No points below 1 are not at 0 after filtering.
**Scaffold_1971 Parentage**

*Figure 3.10 – OAC-Rex ‘AllPaths-LG’ Scaffold_1971 Parentage*

Dot plots showing parentage scores of 100 bp regions of scaffold_1971 from the ‘AllPaths-LG’ generated scaffold set. The unfiltered distribution of parentage is shown in a), while b) shows the same dataset filtered to remove any points that vary by less than 1 stdev from 1. The X-axis represents the nucleotide position along the chromosome and the Y-axis shows the parentage score.

The majority of the points from scaffold_1971 cluster slightly above 1, though there are regions of breaks in the clustering that correspond to clusters on of 0, denoting uniquely PI440795 parentage. Of the G-19833 favouring points, none of them sit on the 2 line, and after filtering it becomes clear that the majority of the parentage is uniquely PI440795. Nearly half of the of the filtered G-19833 points show scores greater than 2.

Across the three scaffolds the filtered datapoints appeared to mostly cluster to the 0 or 2 line, indicating a single parent match is the most common indicator of an introgression. While all three still had values between 0/2 and 1, scaffold_600 showed far more between 0 and 1 than between 1 and 2, while scaffold_570 and scaffold_1971 both
showed the inverse. Of further note is that, even unfiltered, scaffold_570 revealed a section near the right end that had a sudden lack of matching to G-19833.

‘RDP4’ graphs

Scaffold_600, scaffold_570, and scaffold_1971 were submitted to ‘RDP4’ in an attempt to identify an indication of recombination at or near the known markers. ‘RDP4’ was selected for this task as it runs a suite of algorithms to generate a profile of recombination based on an MSA. Several of these algorithms detect breakpoints (‘BURT’, ‘Chimaera’, ‘MAXChi’, ‘TOPAL’), some are based on alignment between the component sequences (‘3Seq’, ‘Bootscan’, ‘GENECONV’, the RDP method, and ‘SiScan’), and others (‘PhylPro’, ‘VisRD’) aim to directly detect recombination. The resultant graphs, shown below, describe the output from each algorithm independently.

Scaffold_600
quality of the alignment on the y. Teal denoted G-19833 against OAC-Rex, purple denoted PI440795 against OAC-Rex, and yellow denoted G-19833 against PI440795. G) MAXChi; Potential breakpoints were indicated by a peak similar to ‘Chimaera’ (d), while using the teal (G-19833 against OAC-Rex), purple (PI440795 against OAC-Rex), and yellow (G-19833 against PI440795) colour scheme. P-values on the graph denote identified peaks. H) PhylPro; Recombinations are identified by calculating the p-distance for the alignment of each subject to every sequence on the left and right side of a sliding window. The corresponding left and right alignments are regressed against each other and a regression coefficient is calculated for each nucleotide. Plotting the coefficients generates a line graph showing correlation of each target against all others, where negative spikes indicate breakpoints and the lowest spike indicates the recombinants. Green represents G-19833, blue represents PI440795 and red represents OAC-Rex. I) RDP method; This method aligns every combination of three sequences from those submitted and generates a percent identity between each pair within the triplet. These are then plotted, and recombination events are detected whenever the percent identity of the most alike pair is reduced below that of any other pair. Teal, purple, and yellow denote G-19833 aligned to OAC-Rex, PI440795 align to OAC-Rex, and G-19833 aligned to PI440795, respectively. J) SiScan; The Sister scanning method determines a Z-score likelihood of alignment between the submitted sequences (yellow for G-19833 against PI440795, teal of G-19833 against OAC-Rex, and purple for PI440795 against OAC-Rex) against a randomization based on those sequences. Like the RDP method it will show a recombination by having the plotted line with the highest Z-score drop while another Z-score overtakes it. The dotted lines denote the bounds of multiple testing to establish a z-score cut-off whose mean is shown by the solid black line. K) TOPAL; This graph identifies breakpoints by comparing the difference of a sum of squares of real sequences and of bootstrap replicates. The light grey lines are those of the replicates and the black line is that of the real sequences. The dashed lines indicate 95% and 99% confidence of expected scores and where the black line raises above these marks is where the recombination breakpoints are predicted to potentially occur. L) VisRD; The green, yellow, and red zones denote different interaction outcomes, however, no signal was recorded.

Even with the ‘breaks’ of non-informative signal, the trend became clear in Figure 3.11(a) (‘3Seq’) that the match to G-19833 continuously increased as the algorithm progresses along the sequence. Importantly, no recombination was detected as evidenced by the absence of an inversion of the score line.

The results from ‘Bootscan’ (Figure 3.11(b)) showed three alignment scores as coloured lines. G-19833 against OAC-Rex was almost entirely above the dashed cut-off threshold, with very few times that it dropped below. When it did drop below the cut-off it was just as often a better match between the parents as it was PI440795.

Of note, the ‘BURT’ output did not differentiate between G-19833/OAC-Rex and PI440795/OAC-Rex lines (both are teal), though the G-19833/PI440795 line remained yellow (Figure 3.11(c)).

Figure 3.11(d) identified only five instances where the measurements spiked above the cut-off, three of which identified breakpoints in OAC-Rex. The two largest spikes, however, identified a seemingly shared breakpoint in both the parents, though G-19833 showed a larger, narrower spike while PI440795 showed a wider range, shortly after the 50910 mark.

The beginning of the distance plot of scaffold_600 (Figure 3.11(e)) showed that PI440795 found no alignment to the start of scaffold_600, though G-19833 did. There were several other instances where the alignment identity for PI440795 to OAC-Rex was
significantly reduced. Some of these instances showed a shared decrease in signal with that of G-19833 to OAC-Rex, although one notably lacked a decrease in G-19833 to PI440795. This patterning was very different for the right most region, where the teal line was almost always above the purple and yellow lines, both of which showed a similar match. There was also a larger section where the only match appeared to be that between the parents and several sections where there was no shared alignment across any of the three sequences.

‘GENECONV’ revealed a relatively straightforward matching set (Figure 3.11(f)). Notably, there were almost no regions identified that showed recombination except in the third section, where G-19833 matched very well

The first two regions of Figure 3.11(g) (‘MAXChi’) showed no detectable peak, though the third region identified two, both when plotting OAC-Rex against G-19833, indicated by the marked P-values.

Scaffold_600 showed only correlation coefficients of 1 or -1 in Figure 3.11(h), with each measured sequence having a different pattern. There was only one transition from 1 to -1 near the far-right that was not strictly vertical. Regardless, the data presented in this figure is essentially uninterpretable since the algorithm identifies breakpoints from signal peaks.

Figure 3.11(i) showed in scaffold_600 there were six locations where the greatest identity score changed, all of them only spikes located in the first two regions. The most alike pair was clearly G-19833 and OAC-Rex, but the first two spikes seen in the first area show a drop in G-19833/OAC-Rex identity that coincides with an increase in PI440795/OAC-Rex identity. The next two recombination spikes had G-19833/OAC-Rex dropping below G-19833/PI440795, as did the last two spikes. The remaining spike showed another short recombination with PI440795.

There appeared to only be spikes showing recombination in Figure 3.11(j) following the same pattern and position as Figure 3.11(i).

Figure 3.11(k), representing ‘TOPAL’ differs greatly from the rest generated using ‘RDP4’, many breakpoints were identified across all three regions.

The last graph from ‘RDP4’ covered the output from ‘VisRD’ (Figure 3.11(l)). This algorithm directly detects recombinant sequences and normally displays them within
one of three tracks, representing the potential ways the sequences can be mapped to each other. This program requires at least four sequences to be run, however, since this study only has OAC-Rex, G-19833, and PI440795 nothing could be reported for this project.

**Scaffold_570**

![Graphical representation of Scaffold_570](image)

**Figure 3.12 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signals Generated via ‘RDP4’**

Each graph within this figure can be observed at a higher resolution in Appendix F. **A)** 3Seq: Each line depicts the pairwise alignment of each nucleotide when assuming a different source is the recombinant; green of G-19833, red for PI440795, and blue for OAC-Rex. Starting from 0 each mapped triplet was given a +1 for a match to 1 parent or a -1 for a match to the other parent. Recombination is indicated by an inversion of the score line. **B)** Bootscan; The lines depict the alignment scores between different combinations of sequences; yellow being G-19833 against OAC-Rex, teal being G-19833 against PI440795, and purple being PI440795 against OAC-Rex. The dashed line represents a minimum cut-off threshold. **C)** BURT; Alignment between OAC-Rex and G-19833 is indicated by yellow and that of G-19833 to PI440795 is teal. **D)** Chimaera; Breakpoints are represented as a series of ratio values of the match of either parent to an assumed recombinant; green for G-19833, blue for OAC-Rex, and red for PI440795. When plotted across an alignment peaks would represent likely breakpoints, provided they cross a minimum p-value cut-off determined during the testing of each assumed recombinant shown by a dashed line. **E)** Distance Plot; A teal/purple/yellow colour scheme is to identify the sequences being measured to pairwise distance between G-19833 and PI440795, PI440795 and OAC-Rex, and G-19833 and OAC-Rex, respectively. **F)** GENECONV; Alignments of sequence are shown as squares to depict the length of the alignment along x and the quality of the alignment on the y. Teal denoted G-19833 against PI440795, purple denoted PI440795 against OAC-Rex, and yellow denoted G-19833 against OAC-Rex. **G)** MAXChi; Potential breakpoints were indicated by a peak similar to ‘Chimaera’ (d), while using the teal (G-19833 against PI440795), purple (PI440795 against OAC-Rex), and yellow (G-19833 against OAC-Rex) colour scheme. P-values on the graph denote identified peaks. **H)** PhylPro; Recombinations are identified by calculating the p-distance for the alignment of each subject to every sequence on the left and right side of a sliding window. The corresponding left and right alignments are regressed against each other and a regression coefficient is calculated for each nucleotide. Plotting the coefficients generates a line graph showing correlation of each target against all others, where negative spikes indicate breakpoints and the lowest spike indicates the recombinants. Green represents G-19833, red represents PI440795 and blue represents OAC-Rex. **I)** RDP method; This method aligns every combination of three sequences from those submitted and generates a percent identity between each pair within the triplet. These are then plotted, and recombination events are detected whenever the percent identity of the most alike pair is reduced below that of any other pair. Teal, purple, and yellow denote G-19833 aligned to PI440795, PI440795 align to OAC-Rex, and G-19833 aligned to OAC-Rex, respectively. **J)** SiScan; The Sister scanning method determines a Z-score likelihood of alignment between the submitted sequences (yellow for G-19833 against OAC-Rex, teal of G-19833 against PI440795, and purple for PI440795 against OAC-Rex) against a randomization based on those sequences. Like the RDP method it will show a recombination by having the plotted line with the highest Z-score drop while another Z-score overtakes it. **K)** TOPAL; This graph identifies breakpoints by comparing the difference of a sum of squares of real sequences and of bootstrap replicates. The light grey lines are those of the replicates and the black line is that of the real sequences. The
dashed lines indicate 95% and 99% confidence of expected scores and where the black line raises above these marks is where the recombination breakpoints are predicted to potentially occur. L) VisRD; The green, yellow, and red zones denote different interaction outcomes, however, no signal was recorded.

‘3Seq’ identified a shared increase in score with very similar increasing trend lines for G-19833 and PI440795 that spike in the right region before abruptly ending, but detected a low, gradually diminishing, score for OAC-Rex even through the same spike early in the right region (Figure 3.12(a)).

The strongest alignments shown in Figure 3.12(b) (‘Bootscan’) correspond to those between G-19833 and PI440795. There are 19 instances where the alignment between PI440795 and OAC-Rex crosses the minimum cut-off, but only two of them appear to coincide with a drop in the alignment score between the parents. While the total count of times the yellow (G-19833 aligned to OAC-Rex) crosses the threshold is largely masked by the colour density of the other two lines, eight instances thereof are visible of it being the highest score. The right region, however, shows no dynamic signal. Instead, it shows a perfect match between G-19833 and PI440795 for a brief period before the signal ends, while the other two alignments are both at 0 for the length of the scaffold.

‘BURT’ returned a uniform signal of perfect match between G-19833 and PI440795, perfect mismatch between OAC-Rex and G-19833, and no apparent signal between PI440795 and OAC-Rex (Figure 3.12(c)), though all signals end shortly into the right region.

‘Chimaera’ was able to detect one peak in Figure 3.12(d) in the far-left end of the right region for G-19833 and PI440795 before a steady line continues through the remainder of the scaffold. OAC-Rex was never above the cut-off, but PI440795 and G-19833 shared similar score line profiles.

In the left region of Figure 3.12(e) the pairwise alignments between OAC-Rex and either parent appeared to be very similar, though both were generally lower than that between the two parents. There were ~35 instances where OAC-Rex displayed lower distance than the parents, mostly to PI440795. The right region only has a signal above 0 for G-19833 associated lines for the very left end, where the parental distance is near 0 and the distance between OAC-Rex and PI440795 is comparable to that between OAC-
Rex and PI440795. The match between OAC-Rex and PI440795 persists to the end of the scaffold, with 5 notable spikes down to 0 and 1 region where the drop is extended.

The only signal reported by ‘GENECONV’ (Figure 3.12(f)) is between G-19833 and PI440795. There is a brief spike at the right end of the left region, but the entirety of the right region shows alignments that persist until the terminus of the scaffold.

Similar to Figure 3.12(d) (‘Chimaera’), ‘MAXChi’ reported a peak in the second region between G-19833 and PI440795 before a sudden leveling off that persisted to the end of the scaffold. However, a second peak was detected at the very start of the left region. Further, the dominant score was always between the parents with OAC-Rex sharing a similar distribution to either parent, though favouring G-19833.

‘PhylPro’ once again only reported correlation coefficients of 1 or -1 in Figure 3.12(h). Unlike for scaffold_600 (Figure 3.11(h)), the only signal to change for scaffold_570 was that of OAC-Rex. In the left region it began at -1 and shortly after jumped to 1, while in the right region it began at -1, jumped to 1, and dropped again to -1. Each transition was instantaneous, and the signal was reported across the length of the scaffold.

The RDP method, shown in Figure 3.12(i) indicated that the alignment between the parents was at all but two instances the better match. The two instances are both in the left region and are both spikes, with the first being between OAC-Rex and G-19833 and the second aligning to PI440795. The right region only has signal detected in the left end which shows a near perfect alignment between the parents and a near 0 alignment for either OAC-Rex alignment.

The pattern from ‘SiScan’ (Figure 3.12(j)) is similar to that from (Figure 3.12(i)); the alignment between the parents scores higher across the board than OAC-Rex does to either parent. That said, there appeared to be more spikes where OAC-Rex does overtake the parental alignment, mostly favouring PI440795. Oddly, in the right region there appeared to be two signals for each alignment, all of which end early in the region. They score the same general trend of parental match being very high and OAC-Rex not crossing the cut-off thresholds.

Figure 3.12(k) showed a signal for bootstrap replicates but failed to produce a signal for the real sequences. Even among the replicates only five spikes crossed the 95%
confidence line, while only 2 crossed the 99% confidence threshold. The signal did fill the entire range of scaffold_570, however.

*Scaffold_1971*

‘RDP4’ failed to produce a single graph for scaffold_1971 as no region of the scaffold was identified as having an informative signal by any of the algorithms.

**Data Co-Visualization**

With the many disparate approaches used to generate data associated to the assembled genomes, the value of each dataset can better be realized through the co-presentation of their features in a way that is helpful to the human observer. Accordingly, the ‘JBrowse’ server ([http://www.bioinfo.uwindsor.ca/~JBrowse](http://www.bioinfo.uwindsor.ca/~JBrowse)) was adopted and established to collate and co-present these data sets. Below, the same three scaffolds are used as examples to show the volume of information readily available and how it can be used to scale from a pseudochromosome down to the nucleotide for targeted genomic elements.
Figure 3.13 – JBrowse Screenshots of Chr04:383593–1396784 from OAC-Rex Pseudochromosomes v4
In each, the top pink and purple bars represent the BWA alignments of the constituent scaffolds and contigs. The bottom two tracks are genomic features annotated in a gff. The left-hand menu shows the available tracks. A) The entirety of pseudochromosome Chr04:383593-1396784, where the blue bars denote the stacked 100 bp regions that aligned to G-19833 (top) or PI440795 (bottom). B) Zoomed-in on scaffold_600 from the ‘AllPaths-LG’ assembly, the blue bars from a) have been broken into columns of short, directionless alignments. C) The PvCTT001 marker, this zoom has revealed the colour coded nucleotide sequence track; red for T, blue for C, green for A, and orange for G. Flanking that track is start and stop codons in each reading frame for either strand.

Using ‘JBrowse’, pseudochromosome Chr04:3383593-1396784 could be investigated (Figure 3.13(a)) and the macro level view was partially informative; the blue bar graphs did give an idea of the distribution of parentage. The browser was used to zoom in on a target region in order to more clearly display data associated to that region. Figure 3.13(b) shows only scaffold_600 within Chr04:383593-1396784 and, while the parentage tracks are less unified, the general trend was still readily apparent among them. At this scale ‘MAKER’ annotations became visible, and, most notably, the position of the marker alignment became clear. Zooming-in one step further (Figure 3.13(c)) to focus on the marker alignment itself, the underlying nucleotide sequence was shown, and the parentage tracks show the individual chunks. A ‘TRF’ identified tandem repeat was seen bridging a gap between the marker alignments, and this gap also coincided with a change in parentage from *P. vulgaris* to *P. acutifolius*. Examining the nucleotide sequence in this repeat region revealed a CTT repeat on the reverse strand repeating 11 times, however, the 5th iteration of the repeat appears to have a C that had been converted to a T.
SU91 CBB Marker

Niemann-Pick-like Gene 1
The top pink and purple bars represent the BWA alignments of the constituent scaffolds and contigs. The bottom two tracks are genomic features annotated in a gff. The left-hand menu shows the available tracks. A) The entirety of pseudochromosome Chr08:62819995-63166454, where the blue bars denote the stacked 100 bp regions that aligned to G-19833 (top) or PI440795 (bottom). B) Zoomed-in on scaffold_570 from the ‘AllPaths-LG’ assembly, the blue bars from a) have been broken into columns of short, directionless alignments for G-19833, but remain too densely packed for PI440795. C) Zoomed to a Niemann-Pick-like gene.

The macro level view of Chr08:62819995-63166454 (Figure 3.14(a)) revealed a pseudochromosome with many gene annotations, seen in the bottom track. Comparing the blue bars that represent parentage the juxtaposition clearly showed how the increase in G-19833 parentage corresponds with a drop in PI440795 parentage, this pattern was maintained near the right end of the region covered by scaffold_570 when there was a sudden lack of any G-19833 alignment; PI440795 parentage tops out the range of the track. The middle view of this pseudochromosome (Figure 3.14(b)), focussing on scaffold_570, did not readily reveal any new patterns, but it added a finer resolution to those established in the macro view. The micro view brought into focus an identified Niemann-Pick like gene with a large interruption (Figure 3.14(c)). This alignment lined up nicely with a ‘MAKER’ annotation, while the gap itself appeared to coincide with an inversion in parentage from G-19833 to PI440795. The end of the gap was also in proximity to several tandem repeats.
Niemann-Pick-like Gene 2

Figure 3.15 – JBrowse Screenshots of Chr08:61349612-63304788 from OAC-Rex Pseudochromosomes v4
The top pink and purple bars represent the BWA alignments of the constituent scaffolds and contigs. The bottom two tracks are genomic features annotated in a gff. The left-hand menu shows the available tracks. A) The entirety of pseudochromosome Chr08:61349612-63304788, where the blue bars denote the stacked 100 bp regions that aligned to G-19833 while those matching PI440795 did not compress to the bar graph. B) Zoomed-in on scaffold_1971 from the 'AllPaths-LG' assembly, the blue bars from a) have been broken into columns of short, directionless alignments. C) Zoomed-in to a Niemann-Pick-like gene

Strikingly obvious when observing Figure 3.15(a) is that there are not enough matches for PI440795 parentage to compress the results into the blue bars like those of G-19833. Despite this, the general trend in parentage can be vaguely seen that the ‘spikes’ in one parentage coincide with the ‘troughs’ of the other. The label for the track obscures the identified Niemann-Pick-like gene at this scale. Zooming in to Figure 3.15(b) is an insufficient change in detail to detect anything new as the track label still obscures too much of the track. Figure 3.15(c) reveals the Niemann-Pick-like gene coinciding with a ‘MAKER’ annotation and, interestingly, a change in parentage from PI440795 to G-19833, however there appears to be an overlap in the parentage, unlike what was observed in Figure 3.15(c).
Phaseolus Genome Structure and Organization

OAC-Rex

Contigs

From OAC-Rex’s ~204 Gbp of short-read sequences 15,794 contigs were assembled via ‘Ray’ (Table 3.1). Covering ~330 Mbp, ~45.8% of the estimated genome size of 720 Mbp, these contigs contained no gaps nor any non-GACT bases and had 204 complete CEGs with another 32 in at least a partial form (Table 3.6). Other contig assemblies generated before settling on ‘Ray’ with a k-mer size of 41 were not kept as the k41 assembly showed higher quality statistics across the board (data not shown).

Scaffolds

Despite having the largest assembly in terms of base pairs, the ‘MaSuRCA’ assembly was the least favourable since ~98.5% of the contigs could not be incorporated into scaffolds, making it nearly identical to the ‘Ray’ k41 contig set. Further, the longest scaffold was less than half the length of the longest scaffold from the ‘SOAPdenovo’ assembly and slightly more than one quarter the length of the ‘AllPaths-LG’ assembly’s longest scaffold (Table 3.7). These two factors taken together were sufficient to abandon using the ‘MaSuRCA’ assembly approach.

Table 3.7 also shows the ‘SOAPdenovo’ assembly had approximately the same coverage as the ‘AllPaths-LG’ assembly, however, with ~1/14th the number of scaffolds, nearly twice the length of the longest scaffold, a ~30% larger N50, and ~1/3rd of the contigs remaining unincorporated into scaffolds, the ‘AllPaths-LG’ assembly is clearly the higher quality assembly. None of the contigs assembled demonstrated this extent of a measurable higher quality in relation to each other and so the ‘AllPaths-LG’ scaffold set was quickly settled upon to be used for all further assembly.

Pseudoautosomes

7017 SNPs were identified in P. vulgaris from the BARCBean6K BeadChips. These were aligned to G-19833 in ‘BLASTn’ to produce 6946 alignments with a minimum e-value of 1e-50. These alignments were renamed to identify their location in G-19833 (Chr###:<start>-<finish>) and aligned against the scaffolds generated by ‘AllPaths-LG’ via ‘BLASTn’. The resulting 6,245 alignments were used as anchor points
to organize and orient their respective scaffolds and generate a chromosome level assembly. Because OAC-Rex is classified as a *P. vulgaris* it was believed that these SNPs would be a more accurate representation than a *P. acutifolius* set. The added genomic data meant that the specific locations would not hold completely accurately, and anchoring may produce extensive overlaps, which should be an impossibility when assembling two genomes of the same species though some minor overlap may be possible. When this overlap occurred, the chromosome being assembled was split and the location of the anchors was kept unchanged generating a set of scaffold-like sequences that represented a chromosome, dubbed pseudochromosomes.

The initial assembly of pseudochromosomes used already annotated ‘AllPaths-LG’ derived scaffolds. Using the identified anchor points of the *P. vulgaris* SNPs, these annotations were updated to reflect their new chromosomal locations. In so doing, errors in the original code were revealed and subsequently corrected, generating OAC-Rex pseudochromosomes v2. This second version was used for much of the initial POI research but was eventually replaced by a third version. The process for creating the third version was the same as that used for the second, however phytozome had released v2.1 of their G-19833 *P. vulgaris* assembly that incorporated long-read data, this new phytozome release was used in lieu of the original short-read based version. While creating the third version of the pseudochromosomes, different changes were made to try and automate the process and/or improve the output assembly such as tweaking the ‘BLASTn’ settings when aligning the SNPs to G-19833, this generated three variations of version 3. Comparing the ‘Assemblathon_stats’ and ‘CEGMA’ outputs (data not shown) demonstrated that the third variation was superior in that the ‘CEGMA’ outputs showed a greater number of complete and partial CEGs.

Comparing OAC-Rex pseudochromosomes v2 to OAC-Rex pseudochromosomes v3.3 revealed that there were fewer scaffolds, slightly less coverage of the estimated genome size, and a comparable N composition indicating that v3.3 did not likely fill in gaps from v2 but redistributed the scaffolds across the chromosomes. Because the third version had a larger N50, though shorter largest scaffold and mean, it was deduced that the longest scaffolds were split into multiple larger than mean scaffolds. Coinciding with this conclusion were fewer, and shorter, gaps in v3.3, all while keeping a comparable
measure of CEGs (Table 3.8). This suggested that v3.3 was of comparable accuracy to v2, but with its scaffolds being less clustered, as indicated by the small gap and scaffold sizes. Version 3.3 covers a broader landscape of the OAC-Rex genome, potentially revealing genomic elements not annotated to the appropriate chromosomal location in v2.

Given the unknown outcome this anchoring approach would produce, a conservative approach was used requiring near-perfect alignments (minimum e-value 1e-50) of the SNPs at every stage. Between v1 and v3.3 other investigations using the assembly had revealed the process to be largely accurate. As denoted in Figure 3.1(a), chromosome 4 showed large amounts of rearrangement while Figure 3.1(b) also showed a minimal amount of rearrangement in chromosome 8, both of which were consistent with earlier understandings of OAC-Rex (Perry, G., Pauls, K.P., Personal Correspondence). These two chromosomes became the focus of further investigations since the terminus of chromosome 8 held an SU91 QTL marker while chromosome 4 matched the PvCTT001 marker, both of which were linked to CBB-associated traits. The stringent requirements placed upon the pseudochromosome assembly served to limit the possible scaffolds that could be anchored. However, given the apparently conserved ordering of the scaffolds between versions 2 and 3.3 (Figure 3.3), a pseudochromosome assembly was attempted with a relaxed minimum e-value.

This assembly, denoted as version 4, retained much of the same apparent organization shown in Figure 3.3(a) between its predecessors (Figure 3.4(a)). Comprising only 5,680 scaffolds (a ~10% decrease from version 3.3) but by a total size ~35 Mbp larger, the data suggested that many of the unanchored scaffolds had been added to a pseudochromosome, possibly using anchors previously ignored. The longest scaffold and N50 were shorter than that of v3.3 or v2; however, the shortest scaffold changed from 890 bp to 927 bp, and the number of gaps, longest gap, and mean gap size were all larger in v4. This suggested that the extra anchored scaffolds generated more overlap causing breaks in the pseudochromosomes, but incorporating multiple scaffolds left previously unanchored. From a content perspective the %N was only ~2% higher at 49.88%, with 218 CEGs complete and another 22 in partial form which lead to the conclusion that, despite the broader distribution of scaffolds into pseudochromosomes,
the content of these scaffolds was not being interrupted in any significant way (Table 3.8).

Unanchored scaffolds were parsed out from the pseudochromosomes and the two data sets were independently analyzed to elucidate the difference between the scaffolds and the pseudochromosomes. 3,396 of the scaffolds identified by ‘Assemblathon_stats’ composed the pseudochromosomes which were ~761 Mbp leaving ~2.8% of the estimated genome left in the unanchored scaffolds. The largest scaffold was, expectedly, found within the pseudochromosomes while the shortest was among the unanchored scaffolds, again as expected. N50, mean scaffold size, and gap associated statistics were not as useful in this analysis as they represent only a subset of the assembly. Of interest were those statistics that overlap, particularly those associated to the CEGs. The pseudochromosomes contained 215 complete CEGs and 237 partial CEGs, while the unanchored scaffolds had 9 and 10, respectively. This means at least 6 complete and 7 partial CEGs were found in both sets. This information can be useful for troubleshooting incorrect anchoring should errors or inconsistencies appear during analysis.

The context established for this project was one that required a high-quality assembly for OAC-Rex, as the known introgressed offspring. To be sure that the final assembly analyzed was of the highest quality every stage from sequencing to analysis needed to be optimized. To this end the search for the best contig, scaffold, and anchoring assembly algorithm was of vital importance. Being able to utilize the best possible end assembly maximized the ability of the subsequent analyses to identify POIs and potential patterns associated thereto.

PI440795

*De novo Assemblies*

Assembling OAC-Rex revealed optimal protocols to follow to generate a draft genome assembly, which were largely followed when assembling PI440795. The difference in depth coverage of both short- and long-reads allowed for more options in assembling PI440795, most notably the lack of MP data. Akin to OAC-Rex, the ‘Ray’ assembly generated the most contigs and covered ~69.7% of the estimated genome size of ~647 Mbp (Gujaria-Verma et al., 2016). There was sufficient PacBio long-read
sequence data to assemble a second contig set via ‘CANU’. This assembly covered ~63.2% of the estimated genome in 18142 contigs. These two contig sets were both functional in their own rights according to the ‘CEGMA’ outputs, which reported 214 complete and another 26 partial CEGs for the ‘Ray’ contigs compared to ‘CANU’ contigs’ 196 complete and another 31 partial CEGs. The reason for the lower functional coverage in the ‘CANU’ based assembly can be ascribed to the lower coverage overall (Table 3.4 and Table 3.5).

Reconciliation

Before anchoring was possible these datasets needed to be unified, a process known as reconciliation. There exist several options for reconciling assemblies, but there is no distinct advantage inherent to one method over another, as such, ‘GARM’ was selected, mostly due to its ease of use (Alhakami, Mirebrahim, & Lonardi, 2017). These reconciled contigs covered ~58.8% of the estimated genome in 8341 contigs, but across all three the %N remained 0 and no gaps were present. The N50 from ‘Ray’, to ‘CANU’, to ‘GARM’ approximately doubled with each interval, while the longest scaffold increased by ~30% and ~113%, respectively, and the mean scaffold size changed from 962 bp to ~23 kbp to ~47 kbp, respectively, all indicating a significant combinatorial effect from ‘GARM’. The ‘CEGMA’ based functionality assessment is the only position that showed a decrease in quality, when compared to the ‘CANU’ generated contigs the reconciled contigs had 1 fewer complete and 4 fewer partial CEGs, an acceptable loss for such major gains in structural features.

Pseudochromosomes

Anchoring PI440795’s reconciled contigs was a helpful step in being able to investigate introgression, since being able to compare only the same chromosomes would allow for an increased resolution in analysis. Unfortunately, there was not a P. acutifolius specific SNP dataset generated to use as anchors, however the high collinearity and synteny among Phaseolus species suggests the same BARCBean6K SNPs aligned to G-19833 should still be a viable anchoring approach, especially since v4 of OAC-Rex pseudochromosomes demonstrated lowering stringency improved anchoring (Bitocchi et al., 2017). Thus, the same pseudochromosome assembly process from v4 was followed using the ‘GARM’ contigs to generate 7,484 scaffolds covering ~819 Mbp (~126.6% of
P. acutifolius’ estimated genome). The largest scaffold grew from ~640 kbp to ~14.8 Mbp while the mean scaffold size more than doubled to 109 kbp and the N50 increased drastically to ~1.92 Mbp. This anchoring introduced 53.48% N in 857 gaps, but 7201 of the ‘GARM’ contigs were left unanchored. Functionally speaking, the anchoring improved the assembly as 2 of the partial CEGs were now near enough in proximity to count as complete.

The differences between the composition of the sequence read datasets meant different approaches had to be used, however, the same goal persisted for both; a high-quality end assembly to analyze. PI440795 could be left significantly less organized than OAC-Rex, provided it was functionally sound since the location of the sequence on a chromosome is not explicitly needed to identify patterns within aligned sequence, though it did require a larger aligning time to filter through all possible sequences. This was not a limiting factor though, since OAC-Rex and G-19833 could be refined to relatively small windows for comparison. The performed POI analysis, detailed later in the discussion, demonstrated that the functional aspects of this assembly approach were sufficient to draw conclusions.

**Sequencing & Assembly – Merits of the Approach**

**Sequencing**

When sequencing first began for OAC-Rex it was based entirely on the use of data sets derived from Illumina short-read technology. There were several reasons for this, the first being availability; Illumina had made a business model focussed largely on accessibility of their product and was in turn widely adopted at the time. Second was cost: this was where Illumina was again at the forefront with a product that was extremely affordable. Third, there was a temporal coincidence: long-read sequencing was not commercially available before 2011, and the diversity of options only made it to market in 2014. The fourth reason was ease-of-use: widespread adoption has generated a large support network of tools and expertise associated to Illumina short-read data. Taken together, these four reasons made Illumina the best option of those available to us when
sequencing. As new technologies became available, they too, were incorporated in the assembly, for instance PacBio long-reads.

With the goal of the project in mind, the higher resolution available from the finer scale of Illumina protocols became more valuable than the broad view of long-read sequencing approaches. Identification of mutations as short as a single nucleotide permitted an investigation into any genomic permutations that may have contributed to introgression. Given the issues that arose from a large-scale approach to POI analysis, the high resolution was apparently essential.

Assembly

With the known introgression in OAC-Rex it was essential that a de novo assembly approach was used in its assembly. The unknown molecular mechanisms would need to be preserved in an assembly without potential influence from a reference, especially given the high level of synteny among Phaseolus species. This required a minimum depth of coverage that exceeded what most assemblies would need, such as that of the PI440795 sequencing.

The cost of a de novo assembly is the large amount of unknown sequence within OAC-Rex, represented by the high N percentage in every version of the pseudochromosomes (Table 3.8). However, as explained in greater detail later in the chapter, the fine scale required to accurately detect POIs would likely not have been possible without a de novo base assembly. That said, the assembly is not entirely de novo since pseudochromosome anchoring is based on the chromosomal organizations established by the G-19833 assembly.

With millions of base-pairs across thousands of scaffolds manual curation of the data was impractical. Figure 3.2 clearly demonstrates this concept; there are alignments, but the data is all but useless for drawing conclusions on a genome wide scale. When the range is adjusted to an individual chromosome and the smaller parts are clustered based on location (i.e. anchored into pseudochromosomes), as in Figure D.1 through Figure D.9, the patterns become understandable, trends become apparent, and interpretations can be made. Most notably examining Figure 3.1 we see that chromosome 4 had smaller syntenic blocks and more rearrangement while chromosome 8 kept the order of syntenic
blocks but found more genetic material in OAC-Rex, an accurate representation of the known status of chromosomes 4 and 8 (Perry, G., Personal Communication). This clarity is scale specific, however, and Figure 3.1(c) and Figure 3.5(c) demonstrate how quickly the data can become noise.

Annotation

At each stage of its assembly, the protocols used to generate that stage of OAC-Rex were assessed and the best was determined. Part of this determination process was the functionality of the assembly, quickly and indirectly measured by ‘CEGMA’, but also more deeply elucidated via thorough annotation. Gene content was assessed using ‘MAKER’ and Table 3.10 describes the outcomes from each stage of OAC-Rex assembly.

Most notable was the similarity between the ‘AllPaths-LG’ scaffolds and version 2 of the pseudochromosomes, this was because the ‘MAKER’ results were originally transposed on to the pseudochromosomes via the ‘Pseudochromosome_gff_updater’ custom script (Appendix II). Second most obvious was the consistent 0 for ‘Genemark’ matches, this was most likely due to a consistently incorrectly installed version of ‘Genemark’. However, this error was not discovered until OAC-Rex pseudochromosomes version 3.3, since ‘MAKER’ was able to draw on many other sources and form consensus it was left unrepaired to be consistent with all other annotation work to date. Another oddity was the zeros for ‘tBLASTx’ and ‘cdna2genome’ for the scaffolding stages, the specific reason for this is unknown and by the time it was discovered, post 3.3, it was already corrected. Furthering the peculiarity of these zeros is that the associated programs (‘BLASTx’ and ‘protein2genome’, respectively) generated results without issue, undermining the idea that the program was incorrectly installed, as well as the fact that the ‘Ray’ contigs generated results in each, suggesting it was not an issue of scaffold size. Beyond these issues there was nothing within the table that stands out. While the numbers vary, sometimes greatly, the consensus output from ‘MAKER’ differs by an average of 2% indicating the annotations are likely very close to representative.
While extensive, ‘MAKER’ annotates only genes. To examine beyond the expressed elements there are many options available and the ‘correct’ tools to be used depend almost entirely on the goal(s) of the project (e.g. https://omictools.com-genome-annotation2-category). Often this can be as simple as getting a sequence for a specific element and aligning it to the genome via ‘BLAST’. For OAC-Rex it was believed that a means of managing the excess introgressed DNA from \textit{P. acutifolius} may mimic extant processes used for silencing expression, such as that of TEs. To this end, several software tools were selected to identify LTRs, characteristically associated to TEs, such that when examined in an integrated way via ‘JBrowse’ they may have hinted at the involvement of TE-like expression control.

With so many unknowns associated with introgression, it was believed that any annotation may significantly contribute to understanding the means by which an organism copes with the introgression event. While a broad view of annotation was deemed beneficial, limited resources restricted the potential tools considered to those not requiring too extensive a specialization or skillset to be implemented, widespread support for assistance with debugging and/or interpretation, and a bias towards more generalist processes.

**Epigenetic elements**

A recipient genome subjected to an introgression event is potentially expanded during introgression by a significant amount. For instance, OAC-Rex’s estimated genome of 720 Mbp is \sim23\% larger than that of G-19833 (587 Mbp) and \sim11\% larger than that of PI440795 (647 Mbp). This expansion was reflected in the ‘Mauve’ based analyses. In each chromosome the OAC-Rex track showed a larger total sequence, and the individual syntenic blocks were widened by regions that did not match to G-19833. Given this influx of new genetic information, it is highly likely that there is some method of regulation on the expression of said information. An obvious candidate for such regulatory system would be an epigenetic silencing mechanism for deleterious genetic elements. Such a hypothesis can be examined by methods such as bisulfite sequencing to identify methylated CpG islands, known to silence associated genes. The use of epigenetic silencing in plants has been well established for a variety of functions
including photoperiodism, vernalization, and aspects of germination (Heo & Sung, 2011; Pikaard & Mittelsten Scheid, 2014). One could predict that examining such a data set would reveal an increase in silencing at loci associated with introgression, particularly if the introgressed DNA contained a normally strictly controlled genetic element. The study’s original intent was to investigate OAC-Rex epigenetics; however, the requirements associated with such a consideration revealed it to be beyond the logistical scope of this inquiry.

POI Identification

Large-Scale Approach

Syntenic Patterns

Once OAC-Rex and PI440795 reads had generated mature pseudochromosome-level assemblies it became possible to search for putative points of introgression. Using ‘Mauve’ to identify synteny between the assemblies demonstrated large blocks of synteny that were conserved between OAC-Rex and G-19833 (Figure 3.1 and Figure D.1 through Figure D.9). The general trend in each chromosome was nearly every syntenic block maintaining relative sequence order but expanding to a larger size with large stretches of OAC-Rex that did not match strictly to G-19833, denoted by the white ‘gaps’ within the coloured blocks. The order of the syntenic blocks was based on the anchoring of scaffolds, which were in turn based on the order of the SNPs as detected in G-19833. Thus, it follows that the general order of the assembled blocks should have followed the general pattern established in G-19833

This pattern was largely maintained by ‘SyMAP’ analysis, shown in Figure 3.5. With each subsequent version of OAC-Rex the amount of synteny detected increases, as shown by the smaller strands linking the OAC-Rex segments to different chromosomes than their G-19833 counterparts. Since this is most notable in OAC-Rex pseudochromosomes v4, which had the least stringent minimum requirement to align a SNP to a scaffold, the question was raised regarding whether or not the lax anchoring was truly reflective of the chromosomal position of OAC-Rex’s scaffolds. The shift in OAC-Rex pseudochromosomes v4 to a lower stringency than its predecessors was an increase
in accuracy. This may be somewhat misleading, as Figure 3.3(a) shows a clear, if imperfect, conservation of the order of syntenic blocks that is not as clearly maintained in Figure 3.4(a). This implied that v4 introduced rearrangement in OAC-Rex, a conclusion that was apparently supported by a brief examination with Figure 3.5(c). However, comparing these to Figure 3.5(b) it became apparent that v3.3 and earlier were missing many already understood aspects of OAC-Rex’s genome structure. While they still possessed some markers, such as the known duplication between the terminus of chromosome 8 and chromosome 6, the most notable missing elements were the lack of any potential disturbance to chromosome 4 and the full scale synteny identified across every chromosome. Figure 3.5(c) clearly shows a high degree of detected synteny throughout the genome and chromosome 7 showed a gap in synteny. This gap was a defining trait of Figure D.6 (Appendix D) and was clearly maintained in Figure 3.3(a), but is entirely absent in Figure 3.5(a) and Figure 3.5(b). With these in mind, OAC-Rex pseudochromosomes v4 were understood to be the most accurate representation of the introgressed genome. Regardless of the arrangement and given the scale of the analysis, both the ‘Mauve’ syntenic blocks and the small ‘SyMAP’ bands were thousands of nucleotides long, supporting the interpretation that introgression works via a large-scale rearrangement/recombination.

Initial POI identification

The involvement of rearrangement/recombination on a large scale was the presumption used when first approaching the objective of identifying POIs. Areas of unique parentage were identified in OAC-Rex by using ‘BLASTn’ to align scaffolds in OAC-Rex anchored to chromosome 8 against chromosome 8 from G-19833, and again aligning these against the entirety of the PI440795 ‘GARM’ contig set since anchoring had not yet been completed. These ‘BLASTn’ alignments used a low stringency e-value of 5e-2 in order to allow for very low score matches to still be registered, ensuring that those matches aligning to only one parent had a high likelihood of being unique in parentage. The resulting matches were tabulated and regions of the OAC-Rex genome that matched exclusively to one parent or another were identified. The shortest sequence between any two uniquely matching regions of opposing parentage were extracted using ‘Fasta_range_subset_extractor’. The resulting nine sequences ranged from < 1 kbp up to
~875 kbp and were submitted to ‘Mauve’ as an approach to identify any common primary sequence-based pattern. Figure 3.6 shows a clear lack of any pattern shared by more than three potential POIs syntenic blocks, all of which are notably short spans.

These nine sequences were also aligned against each other using ‘BLASTn’, to assess whether some pattern could be identified within the isolated sequences. While this approach did reveal additional matches than using ‘Mauve’ alone the results from ‘BLASTn’ failed to highlight any region of alignment shared between more than three POIs (data not shown). Since there was no discernable pattern detected across these POIs a new approach was devised and undertaken.

**MSA Analysis**

Chromosome 8 remained the focus of the investigation since the terminus of the long arm was known to contain the CBB resistance-associated marker SU91. Instead of searching for alignment based on primary sequence, a recombination detection program was used. ‘RDP4’ uses a suite of recombination detection algorithms to build a consensus of where recombination has occurred in a given genomic region. The application requires a multi-sequence alignment, or MSA, as input and ‘MAFFT’ was used to align the terminal end of OAC-Rex, G-19833, and PI440795. ‘MAFFT’ was chosen due to its ease of use, its stability, reduced processing time required, and functionality of its output for further analysis. Attempting to align the terminal 3 Mbp of chromosome 8, enough to cover from the chromosome terminus back to the SU91 marker and approximately an additional 1 Mbp beyond that, failed as Beanblade did not possess enough RAM. Through trial and error, this hardware constraint was found to limit the size of the sequence that could be aligned to 1 Mbp. To overcome potential limitations associated with this scale, a sliding window approach was used, starting from the terminal 1 Mbp and moving 500 kbp per iteration. Figure 3.7 shows the terminal 2 Mbp of chromosome 8 in three sliding windows that had been aligned horizontally such that any point along a vertical line represented the same nucleotide position in the genome. The right end of the top window (highlighted in green) shows a recombination signal, denoted by the purple and teal lines, the location of this signal was also mapped in the second window, but the signal itself did not match the first analysis in that the major parent is inverted. The data revealed a second recombination signal nearer the terminal end, highlighted in red, but
none of the pattern from the second window was found in the third window. The results strongly suggested that the signal being detected was likely a false positive and, combined with the negative result of the earlier POI analyses, that the large-scale approach was not producing viable results. Investigations outside this study confirmed this issue with a large-scale approach, however, it became apparent from the shared data that as the scale was reduced the accuracy, though still low, was increased (Perry, G., Personal Communication). Another new approach was implemented to test the assumption that as the scale of the frame of reference shrank, the accuracy of the detected signal would increase.

**Small-Scale Approach**

CBB resistance has already been linked to the SU91, SAP6, UBC420, and PvCTT001 markers. The target sequence for each of these markers was extracted and combined into a fasta which was aligned against the entirety of OAC-Rex’s ‘AllPaths-LG’ generated scaffolds, as well as version 4 of the pseudochromosomes, to confirm locations. SAP6 and UBC420 returned no hits, which was expected for UBC420 as it is lacking from OAC-Rex’s pedigree, but not for SAP6, which showed similarity on chromosome 10 of G-19833 (Perry et al., 2013). PvCTT001 was matched to scaffold_600 anchored to chromosome 4, and SU91 matched scaffold_570 and scaffold_1971, both anchored to chromosome 8.

**The Protocol**

All the scaffolds anchored to each chromosome were extracted from the ‘AllPaths-LG’ assembly via ‘Fasta_title_assembler’ custom code. These scaffolds were then divided into chunks using ‘Fasta_chunker’ custom code, several chunk sizes were tested which revealed the smaller the chunk the clearer the signal became. Given the read size of OAC-Rex PEs was 100 bp, this was set as the lower limit. The identified region that these scaffolds corresponded to based on SNP alignment that generated the pseudochromosomes was parsed from the G-19833 assembly using ‘Fasta_range_subset_extractor’ custom code. Each 100-bp chunk from OAC-Rex was aligned, using ‘BLASTn’, against the region for the corresponding scaffold extracted from G-19833, thus allowing for a variance in strict positioning of nucleotides due to
introgression altering primary sequence relative to the SNP position, and aligned again against the entirety of the PI440795 reconciled contigs from ‘GARM’, as the anchoring of PI440795 was based on the position of SNPs within G-19833 and would not anchor uniquely *P. acutifolius* specific sequences. The resulting alignments were tabulated and manually curated to show how well each 100 bp chunk from the three identified scaffolds (600, 570, and 1971) aligned to either parent.

Using the bitscores of the alignments, it was reasoned that a ‘parentage score’ could be calculated by dividing the value of score of the alignment to G-19833 over the score of the alignment to PI440795. In this schema, the better the alignment matches PI440795 over G-19833, the more the score would approach 0. Alternatively, the more common the chunk was to *Phaseolus*, regardless of species, the closer the score approaches 1, since identical bitscores would divide to equal 1. The better the alignment matches G-19833 over PI440795, the more the score would approach infinity. In the case of a chunk matching only PI440795 or G-19833 the score was set to 0 or 2, respectively. The results of this approach to parentage scoring and mapping are shown in Figure 3.8(a), Figure 3.9(a), and Figure 3.10(a) for scaffolds 600, 570, and 1971, respectively.

**Variable Resolution**

This revised approach allows setting the desired scoring threshold to define assigning unique parentage to either parent. For the purposes of this work, scores greater than 1 standard deviation from the perfectly common score of 1 were used. To determine this value, 28691 ESTs for PI440795 that were aligned against an earlier OAC-Rex assembly were analyzed to reveal a mean identity between the two parental sequence sets of 92.96% with a standard deviation of 3.69 which, using the coefficient of variation, was translated to a ratio range of ±0.464. Accordingly, this filter rule was applied to all three scaffolds (Figure 3.8(b), Figure 3.9(b), and Figure 3.10(b)). These figures revealed large segments of the genome match to a single parent, suggesting that introgression may also use a medium-scale approach.

**Integration of Genetic and Genomic Data**

The bulk of the data assembled for this project consisted small disparate portions that, independently, did not reveal the impacts or mechanisms of introgression. Much of
this data did present a coordinated view towards the goal of a deeper understanding of OAC-Rex, introgression, and bean genomic organization as a whole. To take advantage of this data synergy, disparate parts were first combined into an informative whole. A particularly poignant example was that of the PvCTT001 marker.

Using ‘JBrowse’, the pseudochromosome was investigated (Figure 3.13) and the macro level view (Figure 3.13(a)) is partially informative; the blue bar graphs do give an idea of the distribution of parentage. The browser can be used to zoom in on a target region in order to more clearly display data associated to that region. Figure 3.13(b) shows only scaffold_600 within Chr04:383593-1396784 and, while the parentage tracks were less unified, the general trend was still readily apparent among them. At this scale ‘MAKER’ annotations became visible, and, most notably, the position of the marker alignment became clear. Zooming-in one step further (Figure 3.13(c)) to focus on the marker alignment itself, the underlying nucleotide sequence was shown, and the parentage tracks showed the individual chunks. A ‘TRF’ identified tandem repeat was seen bridging a gap between the marker alignments, and this gap also coincided with a change in parentage from *P. vulgaris* to *P. acutifolius*. Examining the nucleotide sequence in this repeat region revealed a CTT repeat on the reverse strand repeating 11 times, however, the 5th iteration of the repeat appeared to have a SNP wherein the C had been converted to a T.

Considering that PvCTT001 is a marker for CBB resistance, that the apparent SNP falls in a chunk of *P. acutifolius* parentage, and that the repeat is interrupted all suggest that introgression may act on as little as 1 bp, which may confer trait variance to the introgressed organism. The exact impact of this SNP would require further experimental study to determine its impact on the CBB phenotype, but this case is an example of the sensitivity of the small-scale approach and the synergistic benefits of an integrated display for genomic data. This information is freely available and searchable at [http://bioinfo.uwindsor.ca/jbrowse/](http://bioinfo.uwindsor.ca/jbrowse/).
POI Analysis

Patterns of Introgression at the Genome Level

The three identified scaffolds covered a small fragment of the genome of OAC-Rex (~317 kbp). Once filtered, each showed stretches of consecutive chunks with unique parentage, for instance scaffold_600 (Figure 3.8(b)) has more than fifteen regions where consecutive chunks sit with a score of 2. These consecutive chunks clustered in long spans, such as the lengths on the 0 line in scaffold_570 (Figure 3.9(b)), which span more than half the scaffold (~80 kbp), moderately short spans such as the first on the 0 line in scaffold_1971 (~3 kbp; see Figure 3.10(b)), or spans less than 10 chunks (<1 kbp) such as those one the 0 line before the 520000 mark in scaffold_600.Chunks were tightly packed like scaffold_570, with little space between them, or widely spread like scaffold_1971, where there is ample white space between them. In both scaffold_570 and scaffold_1971 there were only unique matches to PI440795, yet in scaffold_600 there were unique matches to both, though it clearly favoured G-19833.

These chunks were also seen individually and isolated or scoring differently but still consecutively clustered. In scaffold_570, approximately above the 62780000 mark, there was a cluster that differs in parentage score from ~1.55 – ~2.75, but the chunks clustered closely on the x-axis, while scaffold_600 has a much more tightly grouped cluster at ~0.45 just before the 440000 mark. Scaffold_1971, on the other hand, had chunks so dispersed they were truly independent, such as the four chunks in the 62940000–62950000 region.

The only potential pattern that became somewhat apparent was best illustrated when looking at Figure 3.9(b), the 0 line is highly populated, but points are absent between 0 and 0.536 or 1 standard deviation below the mean of 1. While this pattern is maintained in Figure 3.10(b), it was abandoned in Figure 3.8(b) where, while the ‘2’ score is more solidly populated and the vast bulk of the partial matches favour >1 stdev below 1, there are clusters on the 0 line and six data points do not reside on the 2 line.

These repeated contradictions in arrangement; proximal and distal, clustered and dispersed, similar and varied, large and small, consecutive and isolated, revealed a tumultuous landscape. Every trend established in one context was broken elsewhere, no
pattern was maintained across even a small sample (N=3), not even the scale of these arrangements was conserved. Yet the accuracy of the approach was supported when examining genome-wide synteny; **Figure 3.5(c)** identified the chromosome 7 gap, chromosome 4’s genome-wide similarity, and the chromosome 6 duplication. The ability of the protocol to identify potential targets was validated as demonstrated by the ‘JBrowse’ based inspection of the PvCTT001 marker, and **Figure 3.3(a)** and **Figure 3.4(a)** show the general order of synteny was maintained between OAC-Rex pseudochromosome iterations. All told, there do not appear to be any distinct patterns to the form of distribution of parentage during introgression.

**Deduced Mechanisms of Introgression**

In approach to the goal of understanding the mechanisms behind introgression, the three scaffolds were used for a targeted analysis. This analysis aimed to take advantage of the ‘RDP4’ software package’s in-built use of multiple recombination detection algorithms. Unfortunately, this aim was an oversight. ‘VisRD’ and ‘PhylPro’ both directly detect recombinant sequences, but ‘VisRD’ requires at least four sequences to be able to run and thus was unable to report anything, whereas ‘PhylPro’ reported only 1 or -1 for correlation coefficients, but was not static at those values implying that introgression-based recombination happened instantly and that not a single nucleotide was common between the parents, it also made identifying a peak impossible rendering the output uninterpretable. Beyond that, almost none of the algorithms agreed on what they were identifying, as described below.

**Scaffold_600**

The results of each algorithm applied to scaffold_600 are summarized in **Figure 3.11**, where three regions were identified that revealed an informative signal located on both ends plus a third significant signal near the left end of the region depicted. Scaffold_600 was marked as the recombinant, a deduction we can ignore since we know OAC-Rex is the recombined offspring, G-19833 was believed to be the major parent and PI440795 was believed to be the minor parent, though inversions of major/minor parent were not filtered.
The twelve graphs combine to create a detailed image of the recombination landscape of scaffold_600. Unfortunately, this landscape was nonsensical. Beyond ‘VisRD’ and ‘PhylPro’, ‘TOPAL’, ‘MAXChi’, ‘BURT’, and ‘Chimaera’ all aim to identify breakpoints, but where ‘TOPAL’ identified 15+, ‘Chimaera’ identified 1. ‘MAXChi’ failed to detect breakpoints anywhere but the third region, which is the only region in which ‘BURT’ did not detect any breakpoints at all. Alignment based analyses from ‘3Seq’ showed an unchanging match to G-19833, which ‘GENECONV’ supported, but only in the last region. ‘SiScan’, ‘Bootscan’, and the RDP method all agreed that the recombination was occurring in seven localized spikes restricted to the first two regions but disagreed on which organism was better aligned in said spikes.

Importantly, none of the putative recombination sites revealed coincide to the position of the PvCTT001 marker at position 45840 (Figure 3.13(c)). The distance plot is the only graph that has a feature approximately at this position: a span in the second region where all three measurements dropped to the bottom of the graph implying none of the sequences matched each other. Agreement within these figures is restricted to alignment-based recombination spikes which map to none of the identified introgression markers, thus no deductions can be made from this scaffold concerning mechanism of introgression.

**Scaffold_570**

Scaffold_570 showed two broad ranges identified as possessing informative signals for recombination. PI440795 was identified as the recombinant, an inaccurate conclusion, as OAC-Rex is known to be the introgressed offspring, while G-19833 maintained its ranking as major parent and OAC-Rex took PI440795’s place as minor parent. ‘VisRD’ was expectedly blank and ‘PhylPro’ again showed only 1 or -1, however only OAC-Rex was detected as switching between extremes, G-19833 and PI440795 both stayed strictly at 1. ‘MAXChi’ and ‘Chimaera’ showed a similar trend in their graphs and both suggested G-19833 and PI440795 were more likely to be the recombinant sequence than OAC-Rex. Both ‘TOPAL’ and ‘BURT’ showed no signals. ‘3Seq’ showed a stronger signal between G-19833 and PI440795, again with the agreement of ‘GENECONV’ for the latter region. ‘SiScan’ showed three spikes that aligned PI440795/OAC-Rex better than G-19833/PI440795, all in the first region. These
spikes were supported by the RDP method; however, it identified the second spike as a better match between G-19833/OAC-Rex. ‘Bootscan’ identified 10+ spikes above the threshold that corresponded to G-19833/OAC-Rex and 20+ spikes for PI440795/OAC-Rex.

The known introgression on this scaffold corresponds to the Niemann-Pick-like gene and was located ~139 kbp from the start of the scaffold, as seen in Figure 3.14(c), a locus which lay in the center region deemed to have no informative signal. Importantly, none of the ‘RDP4’ algorithms were able to identify OAC-Rex as being the most likely recombinant, nor did the identified likely breakpoints coincide with alignment based recombinations. Regardless, no conclusions can clearly be drawn vis-à-vis recombination in scaffold_570 from these analyses as they do not agree with each other.

**Scaffold 1971**

No recombination was detected in scaffold_1971 from any of the ‘RDP4’ associated algorithms. However, the small-scale POI identification approach revealed a distinct change in parentage at a position ~2,100 bp from the start of the scaffold that coincides with a region annotated as a Niemann-Pick-like gene (Figure 3.15(c)).

**Potential Mechanisms**

The results arising from analyses of the three test scaffolds revealed no shared patterns of genome rearrangement that would suggest a possible mechanism behind introgression. The RDP method and ‘SiScan’ were most consistent in revealing evidence to suggest short stretches of the genome were being briefly recombined, and that these regions may warrant closer analysis at a finer scale. However, the identified positions for recombination did not align with the potential breakpoints that were expected. With no fingerprints identifying the means for introgression to exploit recombinatorial pathway(s) to incorporate new DNA from the minor parent to the major parent, the ability of the small-scale POI identification protocol to isolate potential introgressed variants down to a nucleotide, and the volume of new DNA incorporated into OAC-Rex (~73 Mbp), the logical conclusion is that introgression is likely relying on a genome wide application of very small scale reorganization of the parental genomes.
Implications of this Work

Genome-Assisted Breeding & Development

Aside from the desirability of generating a high-quality genome assembly for OAC-Rex and *P. acutifolius*, a major motivation for this project was to develop a better understanding of introgression mechanisms as it applies it to breeding efforts, particularly those relevant to the Ontario bean breeding enterprise. It was thought that even a rudimentary pattern would have some benefit to plant breeding, such as to identify introgression hotspots, to predict likelihood of introgression of certain elements, or discover a syntenic threshold that could be correlated with success by the introgression process. While this project fell short on providing a directly applicable tool for breeding and development, there were promising and informative results.

The lack of any clear pattern in parentage of the scaffolds denies the opportunity for predictive interpretation; nonetheless, a proposed gene conversion-like mechanism accounts for the fine-scale changes, while allowing for larger stretches of sequence to still be incorporated. Depending on the parent strand used, it can even account for the large stretches of unique parentage. Knowing how to specifically target the impact of such a mechanism may give context or insight to the skilled breeder that would otherwise be left unconsidered. Alternatively, this may raise questions concerning our understanding of gene conversion since the steps of the process are oft admittedly poorly understood. Identifying introgression in organisms that undergo gene conversion as experimentally understood may reveal a mechanism that has eluded researchers. Investigation of the association of POIs to known gene conversion elements may aid in further refining our understanding of the process by more clearly defining boundaries, or the lack thereof, creating an appropriate context to interpret the data.

It may be possible that a wider analysis of parentage may reveal some informative patterning, but without a pattern within a region with known introgression it is unlikely that a pattern will be detected between such regions. While the current body of knowledge cannot identify a mechanism, if such were to be deduced, it may be able to provide essential context for refined analysis.
**For Phaseolus**

While the goal of improving breeding practices from an understanding of introgression may not be apparent, data gained from this project still has an agricultural impact. It was learned that introgression may be conferring resistance by an interruption of the Niemann-Pick gene by splitting the gene product into two pieces (Perry, G., Personal Communication).

Given the benefit of the resistance trait, OAC-Rex has been used repeatedly for further breeding. Having a completed genome for OAC-Rex allows a platform by which research may further identify resistance traits and establish marker sets specific to the resistance variant of said traits. A completed draft genome assembly of *P. acutifolius*, a plant grown regularly for its resistance traits, has yet to be published. These platforms offer new avenues for a wide range of further agronomically and agriculturally relevant research.

To further add to the broadening of the Phaseolus platform is the unique heritage of OAC-Rex. Being of Mesoamerican lineage it allows for the further combined study with G-19833, an Andean bean. The juxtaposition of these genomes allows a potentially unique insight into the impacts of domestication on the evolution of a crop genome. It may reveal details about the history of the domestication process itself and may even be useful for identifying the cultural impact(s) of new world domestication practices compared to their old-world counterparts.

While *Arabidopsis thaliana* has been the epitome of the plant model species since the mid 20th century, it is not a plant cultivated for purposes beyond research and has a relatively small genome. Recent advances in genomics and biotechnology, such as second- and third-generation sequencing, have reduced the need for a few well-established model species with extensive resources and now support the rapid expansion of a new organism to act as a model. As species falling within the genus Phaseolus are largely diploid, are extensively cultivated around the world in diverse cropping regions, exhibit self-pollination and offer a rapid life-cycle, this genus offers significant advantages as a model agricultural species.
For Other Plant Species

Introgression has been recognized for some time, particularly in plants, but the evidence has always been collected well after the introgression has occurred. *A. thaliana* studies have identified introgressed lines for over ten years, but the introgression has only ever been inferred from a QTL analysis (Keurentjes et al., 2007; Törjék et al., 2008). Having a protocol to identify POIs with the genetic resources of an established model like *A. thaliana* may reveal a whole host of details about both the plant and the process that were hitherto only inferred.

An understanding of *A. thaliana* recombination has already established that the recombination rate in *A. thaliana* is low, which would make an interesting case study in introgression (Wijnker et al., 2013). While the recombination rate of OAC-Rex is not known, based on values reported for *P. vulgaris* (Blair et al., 2018), the recombination rate is estimated to be some 15 times higher. If introgression is exploiting gene conversion-like mechanisms it would be expected that the small-scale approach would likely identify far fewer changes in parentage. This investigation can be expanded on as the rate of gene conversion varied in *A. thaliana* for crossover vs non-crossover events, if that holds true for introgression it may be a useful milieu for inquiry into a calculation for rate of introgression. Should the POI examination reveal more frequent changes in parentage it would serve as a prime refutation of the hypothesis that introgression is gene conversion-like in mechanism. Also identified in *A. thaliana* is an enrichment of poly-A and CTT sequences at ancestral recombination hotspots (Choi et al., 2013; Horton et al., 2012), an intriguing association based on the analysis of PvCTT001. If this pattern holds to *P. vulgaris* as well, which would be dependant on just how ancestral the hotspot is, it would appear to support the hypothesis that gene conversion is the foremost mechanism to investigate. Lastly, recombination appeared to cluster in *A. thaliana* around nucleosome-free regions and those with low levels of DNA methylation of the genome, furthering the potential benefit of epigenetic investigation into introgression (Wijnker et al., 2013).

In 2016 the 1001 Genome Consortium published a detailed analysis of their initiative of whole-genome sequencing of 1,000 *A. thaliana* accessions. These data sets are publicly available and will be updated as new information is gathered. While they did
not directly comment on introgression, they identified two groups within *A. thaliana* that they called relics and non-relics, as well several admixtures between the two. They concluded that the relics were accessions that had been isolated during the last glaciation and survived the expansion of non-relics into their environments (Genomes Project, 2016). Those accessions in the admixture group would be perfect candidates for introgression analysis; knowing which relict accessions introgressed with which non-relics could help clarify the timeline of spread of *A. thaliana*. Inversely, these admixed accessions would provide another data set against which to test the approach. The detailed information available already describes much of the evolutionary history of the specific accessions which may also reveal flaws in the small-scale approach by which it can be refined to improve accuracy or elucidate the extent to which such small-scale genome reorganizations occur.

Another plant-based avenue for investigation would be bryophytes (avascular plants, e.g., mosses) as these plants exhibit significantly different patterns in recombination usage. For instance, *Physcomitrella patens* integrates homologous DNA far more frequently than other plantae, where some DNA integrons have a 100% insertion rate. While the insertion appears to use homologous recombination, sometimes in conjunction with NHEJ, the rate is high enough that it has been postulated that a different mechanism is truly at play (Kamisugi et al., 2006). Parentage patterning from the small-scale approach should coincide with the homologous regions showing low levels of unique parentage in a wild-type organism. This model could also be used to explicitly test the small-scale approach by designing a unique insertion, next transforming *P. patens*, then sequencing the offspring and comparing it to the reference genome and a second copy of the reference, modified to contain the insertion at the targeted locus. As the only difference between the provided genomes it should immediately apparent and perfectly identified as introgressed.

**Non-Plant Organisms**

Mating type switching via cassettes in yeast is often used as a textbook example of gene conversion. Wild haploid yeast are one of two mating types denoted as ‘a’ or ‘α’, even if only one cell is the founder of the colony both mating types will be found therein.
This is possible because yeast mating type is controlled by the allele found at the MAT locus, but a silenced version of either allele (MATa or MATα) is found in the HMR or HML loci, respectively. Under appropriate conditions a yeast cell will excise the DNA at the MAT locus and undergo gene conversion using the HMR or HML ‘cassette’ as the homologous base for the new DNA (Hicks & Herskowitz, 1977). Observing the output of the small-scale approach on these loci could serve to refine the parameters by which the unique parentage is defined. It is important to note that while the ‘cassette’ model is novel and fairly well understood, it represents a recombinase mediated form of recombination. General homologous recombination is also well studied in yeast, with many of the pathways so well understood the impact on recombination based on what stage of cycle is a known attribute (Eckert-Boulet, Rothstein, & Lisby, 2011). Generating an introgressed line in yeast can be done similarly to how it was done in P. vulgaris, doing so while observing activity of known recombination-associated genes, such as RAD52, Tel1, or Mre11, may reveal a candidate mechanism (Dujon & Louis, 2017).

Bacteria would provide an entirely different landscape of recombination. While introgression does occur in bacteria, the many ways in which they share DNA in and with their environment complicates identification. Horizontal gene transfer is widespread among bacteria and as such the ‘parentage’ of plasmids is difficult to ascertain. This uptake of new DNA does not necessarily alter of the chromosomal DNA in the bacteria. Hypothetically, the small-scale approach would identify such a non-parental sequence as not aligning to either parent, which could in turn be beneficial to tracking plasmid movement across bacteria. This is unlikely, however, since the initial source of the plasmid would be unknown, and its spread would only be detected if the target gained the plasmid from one of the ‘parents’ investigated. If the focus was on chromosomal DNA only, the small-scale approach should still be able to identify introgressed spans within the genome and a bacterial model for managing the uptake of new may inform further investigations into eukaryotic mechanisms. However, the already understood differences between plasmid regulation likely define hard limits to the applicability of such a model.

A part of the human condition is an unending fascination with ourselves; who we are, where we come from, and what makes us unique are common questions in biology. Oft overlooked are the possible ways understanding biology of one taxon can improve...
the understanding of another taxa. Within the last 200,000 years both Neanderthals and Denisovans became extinct, with some evidence that both may have bred with modern Humans (Huerta-Sánchez et al., 2014; Wills, 2011). While far beyond the scope of this project, this recorded hybridization may have truly been an introgression. A protocol to identify parentage with high precision and resolution would be of obvious benefit to further examine the impacts that ancient interactions with these two species may have had and better understand our own genetic heritage.

**Future Directions**

**Exploring Mechanism(s)**

With the protocol established from this project, the locations of introgressed material has become detectable on a very fine scale; however, evidence of how these fine-scale insertions are occurring has eluded discovery. Understanding of such a mechanism would be a valuable tool for many, even beyond the geneticist. Knowing how the process by which massive amounts of new genetic information can be regulated has the immediate benefit of understanding genome regulation which may be abstracted out to agriculture, to improve yields or reduce stress impact, medicine, as a potential means to treat infection or illness, or conservation, by better understanding how environmental factors are leading to decline in fitness.

In a research context, the lack of a detectable pattern could be evidence of a hitherto unknown molecular mechanism involved in DNA maintenance and control. This seems unlikely, if a mechanism was repeatedly being employed to cope with the impact of introgression it would still present some form of a consistent pattern, even if that pattern was largely nonsensical to the current understanding it would be repeated nonsense. The near-absolute lack of pattern suggests instead that there is no single mechanism being employed and that a diversified examination is needed to identify which of the regulatory pathways is/are being activated during introgression. In line with this thought is that the possible trend of a favoured parent having predominant clustering on the assigned ‘unique’ score and the ‘minor’ parent being more dispersed may have some significance to introgression and merits further investigation. As the only potential
pattern revealed, it is enticing to believe this was hinting at some feature being regularly exercised in the addressing of an introgression event. To explore this, a more widespread (potentially genome-wide) parentage analysis would need to be undertaken whereby every scaffold would be examined in the context of every other scaffold examined. If there is a ‘favoured’ parent it would likely become obvious though the ‘minor’ parent being more dispersed may not be a readily apparent, in Figure 3.8(b) scaffold_600 displayed a potential for both contested ‘favourite’ and the dispersion isolated to the minor parent. Implications of this pattern holding true are varied but include the ability to potentially generate some form of introgression mapping based on density of unique parental sequence, revealing potential foci for further investigation into introgression as some pattern may become apparent within the distribution of the ‘minor’ parent, or even a means towards elucidating mechanism(s) involved. Should it not hold true, however, then there is truly no pattern to how parentage is distributed during introgression, which serves to complicate efforts to identify associated mechanism(s).

The data discovered also raises a critical question concerning the current bioinformatic methods to identify recombination; OAC-Rex is known to be the recombined organism, but the algorithms used often did not identify it as such, nor did they identify the regions where an introgression is known to have occurred. The scale of the examination of these events was small (n = 3), and as such the conclusions should not be considered definitive by any means without further investigation. The fact remains, however, that current tools were unable to accurately identify known elements for which they are made to detect, a problem with potentially far reaching impacts.

**Extending the Current Work**

The scope of the analysis may have hidden a clear signal; the ‘spikes’ identified by the RDP method and almost always supported by ‘SiScan’ initially appear small, but the scale of the scaffolds is still hundreds of thousands of base pairs. Given the improvement garnered by reducing the scale in detecting introgression, it is posited that adjusting the scale of multisequence alignment to match would likewise improve results. This would require a dedicated effort to anchor PI440795 and OAC-Rex to minimize the
number of unanchored scaffolds and clarify the genomic organization of the relevant regions across all three plants.

**Insights on Evolutionary Impact of Domestication and/or Cultivation**

As mentioned earlier, OAC-Rex is of purely Mesoamerican lineage while G-19833 is Andean. An initial objective of the project was to investigate the variance between these two landraces by a phylogenetic analysis of OAC-Rex and G-19833. The dual domestication history of *P. vulgaris* is well documented and the greater cultivation of the Mesoamerican landraces has created a disproportionate amount of agricultural selective pressure on the two landraces. With newly assembled genomes an investigation into the genomic impact of post-domestication cultivation could be performed. The results of such an investigation could be useful for refining our understanding of the evolution of crop species. A comparative analysis would be uniquely positioned to potentially find an impact associated to extensive cultivation. There may also be distinct patterns from the process of domestication as practiced in the New World compared to those of the Old World elucidated by the differences between the two land races.

That said, concerns have been raised that traditional phylogenetic analyses assume an old-world approach to farming, where plots were assigned to a single crop and edge effects were isolated to the rows between plots. New world farming, however, alternated crops by row within a single plot. The impacts of such a difference on resource availability to the crop are potentially not properly accounted for and thus the selective pressures on the crops may not be accurately represented in bioinformatic pipelines. Such an analysis should be delayed until the potential impacts of these differences can be fully assessed and appreciated.


identification and fine mapping of yield-associated QTL. *Genetics, 141*(3), 1147-1162.


treatment. *DNA Repair, 6*(7), 923-935. doi:https://doi.org/10.1016/j.dnarep.2007.02.006


https://www.nature.com/articles/nbt.2280#supplementary-information


https://www.nature.com/articles/nmeth.1226#supplementary-information


https://www.nature.com/articles/nature09014#supplementary-information


### Appendix A – Software Tools Utilized

**Table A.1 – Software Tools Used**

All descriptions are quoted directly from source websites or man pages associated to each software and edited for continuity or to correct spelling and/or grammatical errors.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllPaths-LG</td>
<td>‘AllPaths-LG’ is a whole-genome shotgun assembler that can generate high-quality genome assemblies using short reads (~100 bp) such as those produced by the new generation of sequencers. The significant difference between ‘AllPaths-LG’ and traditional assemblers such as ‘Arachne’ is that ‘AllPaths-LG’ assemblies are not necessarily linear, but instead are presented in the form of a graph. This graph representation retains ambiguities, such as those arising from polymorphism, uncorrected read errors, and unresolved repeats, thereby providing information that has been absent from previous genome assemblies. Source: <a href="https://software.broadinstitute.org/allpaths-lg/blog/">https://software.broadinstitute.org/allpaths-lg/blog/</a></td>
</tr>
<tr>
<td>Assemblathon_stats</td>
<td>An offshoot of the Genome 10K project, and primarily organized by the UC Davis Genome Center, Assemblathons are contests to assess state-of-the-art methods in the field of genome assembly. Source: <a href="https://assemblathon.org/">https://assemblathon.org/</a></td>
</tr>
<tr>
<td>Augustus</td>
<td>‘Augustus’ is a program that predicts genes in eukaryotic genomic sequences. Source: <a href="http://bioinf.uni-greifswald.de/augustus/">http://bioinf.uni-greifswald.de/augustus/</a></td>
</tr>
<tr>
<td>BLAST</td>
<td>The Basic Local Alignment Search Tool (BLAST) finds regions of local similarity between sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches. BLAST can be</td>
</tr>
<tr>
<td>Tool</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>BWA</strong></td>
<td>‘BWA’ is a software package for mapping low-divergent sequences against a large reference genome, such as the human genome. It consists of three algorithms: ‘BWA-backtrack’, ‘BWA-SW’ and ‘BWA-MEM’.</td>
</tr>
<tr>
<td><strong>Canu</strong></td>
<td>‘Canu’ is a fork of the ‘Celera Assembler’ designed for high-noise single-molecule sequencing (such as the PacBio® RSII or Oxford Nanopore® MinION).</td>
</tr>
<tr>
<td><strong>cat</strong></td>
<td>Concatenate FILE(s), or standard input, to standard output.</td>
</tr>
<tr>
<td><strong>CEGMA</strong></td>
<td>‘CEGMA’ (Core Eukaryotic Genes Mapping Approach) is a computational method for building a highly reliable set of gene annotations in the absence of experimental data. Using a set of 458 core proteins that are present in a wide range of taxa. Since these proteins are highly conserved, sequence alignment methods can reliably identify their exon-intron structures in genomic sequences. The resulting dataset can be used to train a gene finder or to assess the completeness of the genome or annotations.</td>
</tr>
</tbody>
</table>

---

used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families.

| **Celera Assembler** | ‘Celera Assembler’ is a *de novo* whole-genome shotgun (WGS) DNA sequence assembler. It reconstructs long sequences of genomic DNA from fragmentary data produced by whole-genome shotgun sequencing.  
| **ClustalΩ** | A multiple sequence alignment program that uses seeded guide trees and HMM profile-profile techniques to generate alignments between three or more sequences. It produces biologically meaningful multiple sequence alignments of divergent sequences. Evolutionary relationships can be seen via viewing cladograms or phylograms.  
Source: [https://www.ebi.ac.uk/Tools/msa/clustalo/](https://www.ebi.ac.uk/Tools/msa/clustalo/) |
| **ClustalW** | A general-purpose DNA or protein multiple sequence alignment program for three or more sequences  
Source: [https://www.ebi.ac.uk/Tools/msa/clustalw2/](https://www.ebi.ac.uk/Tools/msa/clustalw2/) |
| **Exonerate** | ‘Exonerate’ is a generic tool for pairwise sequence comparison. It allows you to align sequences using a many alignment models, either exhaustive dynamic programming or a variety of heuristics.  
Source: [https://www.ebi.ac.uk/about/vertebrate-genomics/software/exonerate](https://www.ebi.ac.uk/about/vertebrate-genomics/software/exonerate) |
| **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

Source:
https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

| FASTX-Toolkit | The ‘FASTX-Toolkit’ is a collection of command line tools for
Short-Reads fasta/fastq files preprocessing. Next-Generation
sequencing machines usually produce fasta or fastq files,
containing multiple short-reads sequences (possibly with quality
information). The main processing of such fasta/fastq files is
mapping (aka aligning) the sequences to reference genomes or
other databases using specialized programs. However, it is
sometimes more productive to preprocess the fasta/fastq files
before mapping the sequences to the genome – manipulating the
sequences to produce better mapping results. The ‘FASTX-
Toolkit’ tools perform some of these preprocessing tasks.

Source: http://hannonlab.cshl.edu/fastx_toolkit/ |
| GARM | ‘GARM’ (Genome Assembler, Reconciliation and Merging) is a
software pipeline to merge and reconcile assemblies from
different algorithms or sequencing technologies. The pipeline is
based mainly implemented using Perl scripts and modules and
third-party open source software like the ‘AMOS’ (Myers et al.,
2000) and ‘MUMmer’ (Kurtz et al., 2004) packages.

Source: http://garm-meta-assem.sourceforge.net/ |
| GBrowse | The Generic Genome Browser (‘GBrowse’) is a simple but
highly configurable web-based genome browser. It is a
component of the Generic Model Organism Systems Database
project (GMOD). |
<table>
<thead>
<tr>
<th><strong>Source:</strong> <a href="https://github.com/GMOD/GBrowse">https://github.com/GMOD/GBrowse</a></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GeneMark</strong></td>
</tr>
<tr>
<td><strong>Source:</strong> <a href="http://exon.gatech.edu/GeneMark/">http://exon.gatech.edu/GeneMark/</a></td>
</tr>
<tr>
<td><strong>HMMER</strong></td>
</tr>
<tr>
<td><strong>Source:</strong> <a href="http://hmmer.org/">http://hmmer.org/</a></td>
</tr>
<tr>
<td><strong>InterProScan</strong></td>
</tr>
<tr>
<td><strong>Source:</strong> <a href="http://www.ebi.ac.uk/interpro/interproscan.html">http://www.ebi.ac.uk/interpro/interproscan.html</a></td>
</tr>
</tbody>
</table>
| JBrowse | ‘JBrowse’ is a fast, scalable genome browser built completely with JavaScript and HTML5. It can run on your desktop, or be embedded in your website. 
Source: https://jbrowse.org/ |
| --- | --- |
| LncTar | ‘LncTar’ is a software for predicting lncRNA-RNA interactions by means of free energy minimization. ‘LncTar’ utilized a variation on the standard "sliding" algorithm approach to calculate the normalized binding free energy (ndG) and found the minimum free energy joint structure. The ndG was regard as a cut-off which determining the paired RNAs as either interacting or not. 
Source: http://www.cuilab.cn/lnctar |
| LongTarget | ‘LongTarget’ was developed to predict a lncRNA’s DNA binding motifs and binding sites in a genomic region based on potential base pairing rules between an RNA sequence and a DNA duplex. 
Source: http://lncrna.smu.edu.cn/ |
| LTR_finder | The program first constructs all exact match pairs by a suffix-array based algorithm and extends them to long highly similar pairs. Then Smith-Waterman algorithm is used to adjust the ends of LTR pair candidates to get alignment boundaries. These boundaries are subject to re-adjustment using supporting information of TG..CA box and TSRs and reliable LTRs are selected. Next, ‘LTR_finder’ tries to identify PBS, PPT and RT inside LTR pairs by build-in aligning and counting modules. RT identification includes a dynamic programming to process frame shift. For other protein domains, ‘LTR_finder’ calls ps_scan to |
locate cores of important enzymes if they occur. Then possible ORFs are constructed based on that. At last, the program reports possible LTR retrotransposon models in different confidence levels according to how many signals and domains they hit.

Source: https://github.com/xzhub/LTR_Finder

| LTRHarvest | A software tool for de novo predictions of LTR retrotransposons in genomic sequences. LTRharvest computes boundary positions of potential LTR retrotransposons on a persistent index structure of the genomic target sequence, the enhanced suffix array. For the prediction, ‘LTRharvest’ implements several filters. These are consecutively applied on the sequence data to reject candidates, which are not conform with sequence, length or distance features of LTR retrotransposons. Since these features are mostly species-specific, every filter can be switched on or switched off and is free for parameterisation of a certain LTR retrotransposon model.

Source: http://genometools.org/index.html

| Mafft | ‘MAFFT’ is a multiple sequence alignment program for UNIX-like operating systems. It offers a range of multiple alignment methods, L-INS-i (accurate; for alignment of <~200 sequences), FFT-NS-2 (fast; for alignment of <~30,000 sequences), etc.

Source: https://mafft.cbrc.jp/alignment/software/

<p>| Maker | ‘MAKER’ is a portable and easily configurable genome annotation pipeline. Its purpose is to allow smaller eukaryotic and prokaryotic genome projects to independently annotate their genomes and to create genome databases. ‘MAKER’ identifies repeats, aligns ESTs and proteins to a genome, produces ab- |</p>
<table>
<thead>
<tr>
<th>Software</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>initio</td>
<td>Gene predictions and automatically synthesizes these data into gene annotations having evidence-based quality values. Source: <a href="https://www.yandell-lab.org/software/maker.html">https://www.yandell-lab.org/software/maker.html</a></td>
</tr>
<tr>
<td>MaSuRCA</td>
<td>‘MaSuRCA’ is whole genome assembly software. It combines the efficiency of the de Bruijn graph and Overlap-Layout-Consensus (OLC) approaches. ‘MaSuRCA’ can assemble data sets containing only short reads from Illumina® sequencing or a mixture of short reads and long reads (Sanger, 454, PacBio® and Nanopore®). Source: <a href="https://github.com/alekseyzimin/masurca">https://github.com/alekseyzimin/masurca</a></td>
</tr>
<tr>
<td>Mauve</td>
<td>Mauve is a system for constructing multiple genome alignments in the presence of large-scale evolutionary events such as rearrangement and inversion. Multiple genome alignments provide a basis for research into comparative genomics and the study of genome-wide evolutionary dynamics. Source: <a href="http://darlinglab.org/mauve/mauve.html">http://darlinglab.org/mauve/mauve.html</a></td>
</tr>
<tr>
<td>md5checksum</td>
<td>Print or check MD5 (128-bit) checksums. With no FILE, or when FILE is -, read standard input. Source: Any POSIX compliant OS</td>
</tr>
<tr>
<td>Muscle</td>
<td>‘MUSCLE’ stands for MUltiple Sequence Comparison by Log-Expectation. ‘MUSCLE’ is claimed to achieve both better average accuracy and better speed than ‘ClustalW2’ or ‘T-Coffee’, depending on the chosen options. ‘MUSCLE’ enables high-throughput applications to achieve average accuracy comparable to the most accurate tools previously available,</td>
</tr>
</tbody>
</table>
which is expected to be increasingly important in view of the continuing rapid growth in sequence data.

Source: [https://www.ebi.ac.uk/Tools/msa/muscle/](https://www.ebi.ac.uk/Tools/msa/muscle/)

| **PBJelly** | ‘PBJelly’ is a highly automated pipeline that aligns long sequencing reads (such as PacBio® RS reads or long 454 reads in fasta format) to high-confidence draft assembles. ‘PBJelly’ fills or reduces as many captured gaps as possible to produce upgraded draft genomes. Each step in ‘PBJelly’s’ workflow can be run on a cluster, thus parallelizing the gap filling process for rapid turn around, even for very large eukaryotic genomes.  
Source: [https://sourceforge.net/p/pbjelly/wiki/Home/](https://sourceforge.net/p/pbjelly/wiki/Home/) |
| **Ray** | ‘Ray’ is a parallel software that computes de novo genome assemblies with next-generation sequencing data. The ‘Ray’ genome assembler is built on top of the ‘Ray’ platform, a generic plugin-based distributed and parallel compute engine that uses the message-passing interface for passing messages.  
| **RDP4** | ‘RDP4’ (Recombination Detection Program version 4) is a Windows XP/VISTA/7/8 program for detecting and analysing recombination and/or genomic reassortment signals in a set of aligned DNA sequences. While a number of other programs have been written to carry out the same task, the motivation for writing ‘RDP4’ has been to produce an analysis tool that is both accessible to users who are uncomfortable with the use of UNIX/DOS command lines and permits a more interactive role in the analysis of recombination. |

160
<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RepeatMasker</td>
<td>‘RepeatMasker’ is a program that screens DNA sequences for interspersed repeats and low complexity DNA sequences. The output of the program is a detailed annotation of the repeats that are present in the query sequence as well as a modified version of the query sequence in which all the annotated repeats have been masked (default: replaced by Ns).</td>
</tr>
<tr>
<td>RepeatRunner</td>
<td>‘RepeatRunner’ is a CGL-based program that integrates ‘RepeatMasker’ with ‘BLASTx’ to provide a comprehensive means of identifying repetitive elements. Because RepeatMasker identifies repeats by means of similarity to a nucleotide library of known repeats, it often fails to identify highly divergent repeats and divergent portions of repeats, especially near repeat edges. To remedy this problem, ‘RepeatRunner’ uses ‘BLASTx’ to search a database of repeat encoded proteins (reverse transcriptases, gag, env, etc...).</td>
</tr>
<tr>
<td>SNAP</td>
<td>‘SNAP’ is a general-purpose gene finding program suitable for both eukaryotic and prokaryotic genomes. ‘SNAP’ is an acronym for Semi-HMM-based Nucleic Acid Parser.</td>
</tr>
<tr>
<td>snoscan</td>
<td>Search for C/D box methylation guide snoRNA genes in a genomic sequence</td>
</tr>
<tr>
<td>Tool</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SOAPdenovo</td>
<td>‘SOAPdenovo’ is a novel short-read assembly method that can build a de novo draft assembly for the human-sized genomes. The program is specially designed to assemble Illumina® GA short reads. It creates new opportunities for building reference sequences and carrying out accurate analyses of unexplored genomes in a cost-effective way.</td>
</tr>
<tr>
<td>SRA Toolkit</td>
<td>The NCBI ‘SRA Toolkit’ enables reading (&quot;dumping&quot;) of sequencing files from the SRA database and writing (&quot;loading&quot;) files into the .sra format</td>
</tr>
<tr>
<td>SyMAP</td>
<td>‘SyMAP’ (Synteny Mapping and Analysis Program) is a software package for detecting, displaying, and querying syntenic relationships between sequenced chromosomes and/or FPC physical maps. It is designed for medium-to-high divergent eukaryotic genomes (not bacteria). It can align a draft genome to a fully sequenced genome, but not draft-to-draft. It can perform self-synteny.</td>
</tr>
<tr>
<td>TRF</td>
<td>A tandem repeat in DNA is two or more adjacent, approximate copies of a pattern of nucleotides. Tandem Repeats Finder is a program to locate and display tandem repeats in DNA sequences. In order to use the program, the user submits a sequence in fasta format. There is no need to specify the pattern, the size of the pattern or any other parameter.</td>
</tr>
<tr>
<td>Tool</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Trimmomatic</td>
<td>Trimmomatic performs a variety of useful trimming tasks for Illumina paired-end and single ended data. The selection of trimming steps and their associated parameters are supplied on the command line.</td>
</tr>
<tr>
<td>‘tRNAscan-SE’-SE</td>
<td>Transfer RNAs are the largest, most complex non-coding RNA family, universal to all living organisms. tRNAscan-SE has been the de facto tool for predicting tRNA genes in whole genomes. The newly developed version 2.0 has incorporated advanced methodologies with improved probabilistic search software and a suite of new gene models, enabling better functional classification of predicted genes.</td>
</tr>
<tr>
<td>Velvet</td>
<td>A de novo genomic assembler specially designed for short read sequencing technologies. Velvet currently takes in short read sequences, removes errors then produces high quality unique contigs. It then uses paired-end read and long read information, when available, to retrieve the repeated areas between contigs.</td>
</tr>
</tbody>
</table>
Appendix B – Custom Code Generated for this Work

Table B.1 – Custom Code Generated for this Work

All custom code is written in Cas self-contained programs and can be found at https://github.com/Pallieguy/BCB-programs. C was used since it allowed for optimal management of limited computational resources. Each piece of code written was tested with a sample dataset relevant to the objective of the code that was manually processed. This manually processed sample was used as a rubric to assess the accuracy of the code.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAST_2_gff3</td>
<td>This program parses out a default tab delimited BLAST output file into a gff3 file. It takes a tabular (-outfmt 6) BLAST output file as input.</td>
</tr>
<tr>
<td>BLAST_parser</td>
<td>This program parses entries in a tab delimited BLAST output file into a new file based on a match in a user defined column to a user defined parameter. It takes any tabular (-outfmt 6) BLAST output file, an integer (column number), and a string (match parameter) as input.</td>
</tr>
<tr>
<td>BLAST_best_output_parser</td>
<td>This program filters a tab delimited BLAST output file based on highest bitscore and e-value, keeping ties, into a new file. It takes a tabular (-outfmt 6) BLAST output file as input, the bitscore must be the last column and e-value the second last</td>
</tr>
<tr>
<td>BLAST_title_corrector</td>
<td>This program generates a file with the titles from a tab delimited BLAST output file changed to match those of the fasta input file. It takes the Fasta_title_parser output file from the fasta file used as the index and the tabular (-outfmt 6) BLAST output file as inputs.</td>
</tr>
<tr>
<td>Fasta_2_fastq</td>
<td>This program takes a fasta file and generates a fastq file with a PHRED score of 40 from it. It takes a fasta file as input.</td>
</tr>
<tr>
<td>Program Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Fasta_2_gff3</td>
<td>This program makes a contig gff3 file from a fasta file. It takes a fasta file as input.</td>
</tr>
<tr>
<td>Fasta_AMP_generator</td>
<td>This program parses entries in a fasta file to generate a fastq file of interleaved artificial 100 bp mate pair reads of user defined distance with a 60% overlap and a PHRED score of 40. It takes a fasta file and an integer (distance) as input.</td>
</tr>
<tr>
<td>Fasta_APE_generator</td>
<td>This program parses entries in a fasta file to generate a fastq file of interleaved artificial 100 bp paired end reads with a 60% overlap and a PHRED score of 40. It takes a fasta file as input.</td>
</tr>
<tr>
<td>Fasta_chunker</td>
<td>This program breaks entries a fasta file into chunks of user defined size in bp, saved in a new fasta file. It takes a fasta file and an integer (chunk size) as input.</td>
</tr>
<tr>
<td>Fasta_file_combiner</td>
<td>This program combines a title file and a sequences file into a fasta file. It takes a file of titles and a file of single-line sequences as input.</td>
</tr>
<tr>
<td>Fasta_G2N</td>
<td>This program corrects the output file from an assembler that changed all 'N' to 'G'. It takes a fasta file as input.</td>
</tr>
<tr>
<td>Fasta_gap_analyzer</td>
<td>This program analyzes a fasta file and calculates several stats concerning the gap (a sequence of Ns of at least a user defined length) content thereof, reporting to stdout. It takes a fasta file and an integer (minimum gap size) as input.</td>
</tr>
<tr>
<td>Fasta_gap_compressor</td>
<td>This program generates a fasta where all N repeats in each entry a fasta file longer than a user defined length are shortened to that length. It takes a fasta file and an integer (maximum gap length) as input.</td>
</tr>
<tr>
<td><strong>Program</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Fasta_individualizer</td>
<td>This program takes a multi-entry fasta file and generates a series of single entry fasta files. It takes a fasta file as input.</td>
</tr>
<tr>
<td>Fasta_interleaver</td>
<td>This program interleaves two fasta files into a single fasta file. It takes the shared part of the R1 and R2 file names as input.</td>
</tr>
<tr>
<td>Fasta_range_subset_extractor</td>
<td>This program parses entries from a fasta file that match a provided range list in the format <code>&lt;fasta entry title&gt;:&lt;start position&gt;-&lt;stop position&gt;</code> into a new fasta. It takes a fasta file and a file of ranges as input.</td>
</tr>
<tr>
<td>Fasta_read_extractor</td>
<td>This program parses entries from a fasta file to generate a new fasta with up to a user defined number of reads of user defined length and overlap from the user defined terminus of the entries. It takes a fasta file, &quot;Start&quot; or &quot;End&quot;, and three integers (extract length, overlap size, and extract count) as inputs.</td>
</tr>
<tr>
<td>Fasta_single_entry_compressor</td>
<td>This program generates a single-entry fasta file from a multi-entry fasta file, separating entries by 25N. It takes a fasta file as input.</td>
</tr>
<tr>
<td>Fasta_sorter</td>
<td>This program generates a fasta file of sorts the entries in a fasta file into a new fasta file. It takes a fasta file as input.</td>
</tr>
<tr>
<td>Fasta_title_assembler</td>
<td>This program generates a fasta file from a source fasta file and a list of fasta titles. It takes a fasta file and a title file as input.</td>
</tr>
<tr>
<td>Fasta_title_parser</td>
<td>This program parses out the titles of each entry in a fasta file, calculates the length of each entry, and a cumulative length of the entries into a new file and</td>
</tr>
<tr>
<td>Program Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>prints the final statistics to stdout. It takes a fasta file as input.</td>
<td></td>
</tr>
<tr>
<td>Fastq_2_fasta</td>
<td>This program generates a fasta file from a fastq file. It takes a fastq file as input.</td>
</tr>
<tr>
<td>Fastq_AMP_generator</td>
<td>This program parses entries in a long-read fastq file to generate a fastq file of artificial mate pair reads of user defined distance. It takes a fastq file and a long (distance) as input.</td>
</tr>
<tr>
<td>Fastq_coverage_parser</td>
<td>This program copies entries in a fastq file into a new fastq file until a user defined Kbp length has been reached. It takes a fastq file and an integer as input.</td>
</tr>
<tr>
<td>Fastq_interleaver</td>
<td>This program interleaves two fastq files generating a single fastq file. It takes the shared part of the R1 and R2 file names as input.</td>
</tr>
<tr>
<td>Fastq_interleave_separator</td>
<td>This program separates an interleaved fastq file into two matching fastq files. It takes a fastq file as input.</td>
</tr>
<tr>
<td>Fastq_length_check</td>
<td>This program generates a fastq file by filtering entries below a user defined length from a fastq file. It takes a fastq file and a long as input.</td>
</tr>
<tr>
<td>Fastq_linker_trimmer</td>
<td>This program generates a fastq file by removing a user defined sequence and its palindrome from the end of each entry (if found) in a fastq file. It takes a fastq file and a string as input.</td>
</tr>
<tr>
<td>Fastq_size_sorter</td>
<td>This program sorts the entries in a fastq file from largest to smallest into a new fastq file. It takes a fastq file as input.</td>
</tr>
<tr>
<td>Fastq_sort_check</td>
<td>This program checks that the entries in two fastq files are in the same order, reporting any that do not</td>
</tr>
<tr>
<td>Program</td>
<td>Function</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Fastq_sorter</td>
<td>This program generates a fastq file by sorting the entries in a fastq file. It takes a fastq file as input.</td>
</tr>
<tr>
<td>Fastq_synchronizer</td>
<td>This program generates two fastq files by reordering the entries of two fastq files, so they are identical. It takes the shared part of the R1 and R2 file names as input.</td>
</tr>
<tr>
<td>Fastq_title_parser</td>
<td>This program generates a file listing the titles and sizes of all the entries of a fastq file, it reports final statistics to stdout. It takes a fastq file as input.</td>
</tr>
<tr>
<td>Gff_contig_title_parser</td>
<td>This program generates a file listing contig titles and sizes from a gff3 file. It takes a contig gff3 file as input.</td>
</tr>
<tr>
<td>Gff_fasta_compiler</td>
<td>This program generates a fasta file from the ‘gene’ or ‘transcript’ entries in a gff3. It takes a gff3 file and a fasta file as input.</td>
</tr>
<tr>
<td>Gff_sorter</td>
<td>This program generates a reordered gff3 file sorted by scaffold then start. It takes a gff3 file as input.</td>
</tr>
<tr>
<td>Gff_stat_parser</td>
<td>This program counts entries for known algorithms from a gff3 file and reports totals to stdout. It takes a gff3 file as input.</td>
</tr>
<tr>
<td>LTR_finder_gff_generator</td>
<td>This program generates a gff3 file from an ‘LTR_Finder’ output data file. It takes an ‘LTR_Finder’ output file as input.</td>
</tr>
<tr>
<td>LTRHarvest_gff_corrector</td>
<td>This program generates a gff3 file from entries renamed to the scaffolds in an ‘LTRharvest’ output gff file. It takes a gff3 file as input.</td>
</tr>
<tr>
<td><strong>MassCompile</strong></td>
<td>This bash script enters every directory and compiles all of the code on any POSIX compliant system. It needs no additional input.</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Mauve_clustal_converter</strong></td>
<td>This program pulls the data from a two entry ‘Mauve’ output file and generates a ‘ClustalW’ file. It takes the extensionless ‘Mauve’ output file, the name of the first subject therein, and the name of the second subject therein as inputs.</td>
</tr>
<tr>
<td><strong>Mauve_title_corrector</strong></td>
<td>This program generates a new ‘Mauve’ output file with the titles from a ‘Mauve’ output file changed to match those of the multi-fasta inputs. It takes the ‘Fasta_title_parser’ output used as the first ‘Mauve’ input, the ‘Fasta_title_parser’ output used as the second ‘Mauve’ input, and the extensionless ‘Mauve’ output file as inputs.</td>
</tr>
<tr>
<td><strong>Mauve_title_parser</strong></td>
<td>This program generates a file listing titles matched by ‘Mauve’. It takes an extensionless ‘Mauve’ output file as input.</td>
</tr>
<tr>
<td><strong>POI_gff_generator</strong></td>
<td>This program generates a gff3 file from a POI csv. It takes a csv file as input.</td>
</tr>
<tr>
<td><strong>Pseudochromosome_assembler</strong></td>
<td>This program generates a pseudochromosome fasta file from a tab delimited (-outfmt 6) BLAST output file of positionally renamed anchors aligned to the scaffold set, filling the resulting gaps with 'N'. It takes a fasta file, and a curated tabular BLAST output file as input.</td>
</tr>
<tr>
<td><strong>Pseudochromosome_gff_updater</strong></td>
<td>This program generates a gff3 file of updated locations to match the new pseudochromosome positions. It takes a gff3 file and a</td>
</tr>
<tr>
<td>Program Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pseudochromosomes_scaffold_locations.txt file (from Pseudochromosome_assembler) as inputs.</td>
<td></td>
</tr>
<tr>
<td><strong>Title_comparator</strong></td>
<td>This program compares the titles between two title files and reports any mismatches to stdout. It takes any two ‘Fast*_title_parser’ output files as input.</td>
</tr>
<tr>
<td><strong>Title_uniqueness_parser</strong></td>
<td>This program filters the fasta titles between two title files, generating a title file of only unique titles in either list. It takes any two ‘Fasta_title_parser’ output files as input.</td>
</tr>
<tr>
<td><strong>TRF_gff_generator</strong></td>
<td>This program generates a gff3 file from a ‘TRF’ output data file. It takes a ‘TRF’ output file as input.</td>
</tr>
<tr>
<td><strong>tRNAscan-SE_gff_generator</strong></td>
<td>This program generates a gff3 file from a ‘tRNAscan-SE’ output data file. It takes a ‘tRNAscan-SE’ output file as input.</td>
</tr>
</tbody>
</table>
### Appendix C – Translation Tables

**Table C.1 – IUPAC Nucleotide Code and Meaning**

<table>
<thead>
<tr>
<th>IUPAC Nucleotide Code</th>
<th>Base</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Adenine</td>
</tr>
<tr>
<td>C</td>
<td>Cytosine</td>
</tr>
<tr>
<td>G</td>
<td>Guanine</td>
</tr>
<tr>
<td>T (or U)</td>
<td>Thymine (or Uracil)</td>
</tr>
<tr>
<td>R</td>
<td>Adenine or Guanine</td>
</tr>
<tr>
<td>Y</td>
<td>Cytosine or Thymine</td>
</tr>
<tr>
<td>S</td>
<td>Guanine or Cytosine</td>
</tr>
<tr>
<td>W</td>
<td>Adenine or Thymine</td>
</tr>
<tr>
<td>K</td>
<td>Guanine or Thymine</td>
</tr>
<tr>
<td>M</td>
<td>Adenine or Cytosine</td>
</tr>
<tr>
<td>B</td>
<td>Cytosine, Guanine, or Thymine</td>
</tr>
<tr>
<td>D</td>
<td>Adenine, Guanine, or Thymine</td>
</tr>
<tr>
<td>H</td>
<td>Adenine, Cytosine, or Thymine</td>
</tr>
<tr>
<td>V</td>
<td>Adenine, Cytosine, or Guanine</td>
</tr>
<tr>
<td>N</td>
<td>Any Base</td>
</tr>
<tr>
<td>. or -</td>
<td>Gap of unknown size</td>
</tr>
</tbody>
</table>

(Nomenclature, 1970, 1986)
Figure C.1 – ASCII Table

With various Numeric value equivalents
### Figure C.2 – Canonical Codon Translation Table

<table>
<thead>
<tr>
<th>First base of codon</th>
<th>Second base of codon</th>
<th>Third base of codon</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>UU</td>
<td>U</td>
</tr>
<tr>
<td></td>
<td>UUC</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>UAU</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>UUG</td>
<td>G</td>
</tr>
<tr>
<td>C</td>
<td>CUU</td>
<td>U</td>
</tr>
<tr>
<td></td>
<td>CUC</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>CUU</td>
<td>G</td>
</tr>
<tr>
<td>A</td>
<td>AUC</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>AUG</td>
<td>G</td>
</tr>
<tr>
<td>G</td>
<td>GUU</td>
<td>U</td>
</tr>
<tr>
<td></td>
<td>GUC</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>GUA</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>GUG</td>
<td>G</td>
</tr>
</tbody>
</table>

- **Phenylalanine** (phe): UUU, UUC, UUA, UUG
- **Leucine** (leu): CUU, CUC, CUA, CUG
- **Proline** (pro): CAU, CAC, CAA, CAG
- **Histidine** (his): CAU, CAC, CAA, CAG
- **Alanine** (ala): GCU, GCC, GCA, GCG
- **Valine** (val): GUU, GUC, GUA, GUG
- **Serine** (ser): UUA, UAG
- **Leucine** (leu): CUU, CUC, CUA, CUG
- **Valine** (val): GUU, GUC, GUA, GUG
- **Histidine** (his): CAU, CAC, CAA, CAG
- **Alanine** (ala): GCU, GCC, GCA, GCG
- **Serine** (ser): UUA, UAG
- **Leucine** (leu): CUU, CUC, CUA, CUG
- **Valine** (val): GUU, GUC, GUA, GUG
- **Histidine** (his): CAU, CAC, CAA, CAG
- **Alanine** (ala): GCU, GCC, GCA, GCG
- **Serine** (ser): UUA, UAG
- **Leucine** (leu): CUU, CUC, CUA, CUG
- **Valine** (val): GUU, GUC, GUA, GUG
- **Histidine** (his): CAU, CAC, CAA, CAG
- **Alanine** (ala): GCU, GCC, GCA, GCG
- **Serine** (ser): UUA, UAG
- **Leucine** (leu): CUU, CUC, CUA, CUG
- **Valine** (val): GUU, GUC, GUA, GUG
- **Histidine** (his): CAU, CAC, CAA, CAG
- **Alanine** (ala): GCU, GCC, GCA, GCG
- **Serine** (ser): UUA, UAG
- **Leucine** (leu): CUU, CUC, CUA, CUG
- **Valine** (val): GUU, GUC, GUA, GUG
- **Histidine** (his): CAU, CAC, CAA, CAG
- **Alanine** (ala): GCU, GCC, GCA, GCG
- **Serine** (ser): UUA, UAG
Appendix D – Syntenic Alignments of G-19833 v1 and OAC-Rex Pseudochromosomes v2 via ‘Mauve’

In each, colour coded segments represent syntenic blocks with a line connecting the corresponding blocks. The general order of synteny is maintained from G-19833 v1 (top) but additional unmatching DNA, denoted by white sections within the coloured blocks, is expanding the size of the blocks in OAC-Rex v2 (bottom). The red lines denote a new entry in the supplied fasta files.

Figure D.1 – Chromosome 01

Figure D.2 – Chromosome 02

Figure D.3 – Chromosome 03
Figure D.4 – Chromosome 05

Figure D.5 – Chromosome 06

Figure D.6 – Chromosome 07

Figure D.7 – Chromosome 09
Figure D.8 – Chromosome 10

Figure D.9 – Chromosome 11
Appendix E – Individual Scaffold_600 ‘RDP4’

Generated Graphs

**Figure E.1** – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘3Seq’ via ‘RDP4’

Each line depicts the pairwise alignment of each nucleotide when assuming a different source is the recombinant; green of G-19833, blue for PI440795, and red for OAC-Rex. Starting from 0 each mapped triplet was given a +1 for a match to 1 parent or a -1 for a match to the other parent. Recombination is indicated by an inversion of the score line.

**Figure E.2** – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘Bootscan’ via ‘RDP4’

The lines depict the alignment scores between different combinations of sequences, yellow being G-19833 against PI440795, teal being G-19833 against OAC-Rex, and purple being PI440795 against OAC-Rex. The dashed line represents a minimum cut-off threshold.
Figure E.3 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘BURT’ via ‘RDP4’

Alignment between OAC-Rex and G-19833 is indicated by teal, that of OAC-Rex to PI440795 is also teal, and that of G-19833 to PI440795 is yellow.

Figure E.4 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘Chimaera’ via ‘RDP4’

Breakpoints are represented as a series of ratio values of the match of either parent to an assumed recombinant; green for G-19833, blue for PI440795, and red for OAC-Rex. When plotted across an alignment peaks would represent likely breakpoints, provided they cross a minimum p-value cut-off determined during the testing of each assumed recombinant shown by a dashed line.

Figure E.5 – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Distance Plot via ‘RDP4’
A teal/purple/yellow colour scheme is to identify the sequences being measured to pairwise distance between G-19833 and OAC-Rex, PI440795 and OAC-Rex, and G-19833 and PI440795, respectively.

**Figure E.6** – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘GENECONV’ via ‘RDP4’

Alignments of sequence are shown as squares to depict the length of the alignment along x and the quality of the alignment on the y. Teal denoted G-19833 against OAC-Rex, purple denoted PI440795 against OAC-Rex, and yellow denoted G-19833 against PI440795.

**Figure E.7** – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘MAXChi’ via ‘RDP4’

Potential breakpoints were indicated by a peak similar to ‘Chimaera’ (**Figure E.4**), while using the teal (G-19833 against OAC-Rex), purple (PI440795 against OAC-Rex), and yellow (G-19833 against PI440795) colour scheme. P-values on the graph denote identified peaks.
Recombinations are identified by calculating the p-distance for the alignment of each subject to every sequence on the left and right side of a sliding window. The corresponding left and right alignments are regressed against each other and a regression coefficient is calculated for each nucleotide. Plotting the coefficients generates a line graph showing correlation of each target against all others, where negative spikes indicate breakpoints and the lowest spike indicates the recombinants. Green represents G-19833, blue represents PI440795 and red represents OAC-Rex.

This method aligns every combination of three sequences from those submitted and generates a percent identity between each pair within the triplet. These are then plotted, and recombination events are detected whenever the percent identity of the most alike pair is reduced below that of any other pair. Teal, purple, and yellow denote G-19833 aligned to OAC-Rex, PI440795 align to OAC-Rex, and G-19833 aligned to PI440795, respectively.
**Figure E.10** – OAC-Rex 'AllPaths-LG’ Scaffold_600 Recombination Signal from ‘SiScan’ via ‘RDP4’

The Sister scanning method determines a Z-score likelihood of alignment between the submitted sequences (yellow for G-19833 against PI440795, teal of G-19833 against OAC-Rex, and purple for PI440795 against OAC-Rex) against a randomization based on those sequences. Like the RDP method it will show a recombination by having the plotted line with the highest Z-score drop while another Z-score overtakes it. The dotted lines denote the bounds of multiple testing to establish a z-score cut-off whose mean is shown by the solid black line.

**Figure E.11** – OAC-Rex ‘AllPaths-LG’ Scaffold_600 Recombination Signal from ‘TOPAL’ via ‘RDP4’

This graph identifies breakpoints by comparing the difference of a sum of squares of real sequences and of bootstrap replicates. The light grey lines are those of the replicates and the black line is that of the real sequences. The dashed lines indicate 95% and 99% confidence of expected scores and where the black line raises above these marks is where the recombination breakpoints are predicted to potentially occur.
Figure E.12 – OAC-Rex 'AllPaths-LG' Scaffold_600 Recombination Signal from 'VisRD' via 'RDP4'

The green, yellow, and red zones denote different interaction outcomes, however, no signal was recorded.
Appendix F – Individual Scaffold_570 ‘RDP4’

Generated Graphs

Figure F.1 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘3Seq’ via ‘RDP4’

Each line depicts the pairwise alignment of each nucleotide when assuming a different source is the recombinant; green of G-19833, red for PI440795, and blue for OAC-Rex. Starting from 0 each mapped triplet was given a +1 for a match to 1 parent or a -1 for a match to the other parent. Recombination is indicated by an inversion of the score line.

Figure F.2 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘Bootscan’ via ‘RDP4’

The lines depict the alignment scores between different combinations of sequences, yellow being G-19833 against OAC-Rex, teal being G-19833 against PI440795, and purple being PI440795 against OAC-Rex. The dashed line represents a minimum cut-off threshold.
Figure F.3 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘BURT’ via ‘RDP4’

Alignment between OAC-Rex and G-19833 is indicated by yellow and that of G-19833 to PI440795 is teal

Figure F.4 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘Chimaera’ via ‘RDP4’

Breakpoints are represented as a series of ratio values of the match of either parent to an assumed recombinant; green for G-19833, blue for OAC-Rex, and red for PI440795. When plotted across an alignment peaks would represent likely breakpoints, provided they cross a minimum p-value cut-off determined during the testing of each assumed recombinant shown by a dashed line

Figure F.5 – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Distance Plot via ‘RDP4’
A teal/purple/yellow colour scheme is to identify the sequences being measured to pairwise distance between G-19833 and PI440795, PI440795 and OAC-Rex, and G-19833 and OAC-Rex, respectively.

**Figure F.6** – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘GENECONV’ via ‘RDP4’

Alignments of sequence are shown as squares to depict the length of the alignment along x and the quality of the alignment on the y. Teal denoted G-19833 against PI440795, purple denoted PI440795 against OAC-Rex, and yellow denoted G-19833 against OAC-Rex.

**Figure F.7** – OAC-Rex ‘AllPaths-LG’ Scaffold_570 Recombination Signal from ‘MAXChi’ via ‘RDP4’

Potential breakpoints were indicated by a peak similar to ‘Chimaera’ (**Figure F.4**), while using the teal (G-19833 against PI440795), purple (PI440795 against OAC-Rex), and yellow (G-19833 against OAC-Rex) colour scheme. P-values on the graph denote identified peaks.
Recombinations are identified by calculating the p-distance for the alignment of each subject to every sequence on the left and right side of a sliding window. The corresponding left and right alignments are regressed against each other and a regression coefficient is calculated for each nucleotide. Plotting the coefficients generates a line graph showing correlation of each target against all others, where negative spikes indicate breakpoints and the lowest spike indicates the recombinants. Green represents G-19833, red represents PI440795, and blue represents OAC-Rex.

This method aligns every combination of three sequences from those submitted and generates a percent identity between each pair within the triplet. These are then plotted, and recombination events are detected whenever the percent identity of the most alike pair is reduced below that of any other pair. Teal, purple, and yellow denote G-19833 aligned to PI440795, PI440795 align to OAC-Rex, and G-19833 aligned to OAC-Rex, respectively.
The Sister scanning method determines a Z-score likelihood of alignment between the submitted sequences (yellow for G-19833 against OAC-Rex, teal of G-19833 against PI440795, and purple for PI440795 against OAC-Rex) against a randomization based on those sequences. Like the RDP method it will show a recombination by having the plotted line with the highest Z-score drop while another Z-score overtakes it.

This graph identifies breakpoints by comparing the difference of a sum of squares of real sequences and of bootstrap replicates. The light grey lines are those of the replicates and the black line is that of the real sequences. The dashed lines indicate 95% and 99% confidence of expected scores and where the black line raises above these marks is where the recombination breakpoints are predicted to potentially occur.

The green, yellow, and red zones denote different interaction outcomes, however, no signal was recorded.
NAME: Seth Alexander Munholland

PLACE OF BIRTH: Calgary, Alberta

YEAR OF BIRTH: 1985

EDUCATION: Ernest Manning High School, Calgary, Alberta
2001-2004

University of Windsor, Windsor, Ontario
2009-2013 B.Sc.

University of Windsor, Windsor, Ontario
2013-2019 Ph.D.