
KnowPulse Documentation

LA Sanderson, CT Caron, R Tan, Y Shen, KE Bett (University of Sa

May 26, 2021

Use Cases:

- | | | |
|----------|--|----------|
| 1 | Get a list of SNPs/variants between 2+ varieties within a region of interest. | 1 |
| 2 | Developers | 7 |

CHAPTER 1

Get a list of SNPs/variants between 2+ varieties within a region of interest.

Example:

- Region of interest: LcChr4 : 1875000–2075000
- Germplasm of Interest: ILL 8006, CDC Milestone

1.1 Step #1: Go to the Lentil Genotype Matrix Tool

- Go to <https://knowpulse.usask.ca>
- Click on “Genotypes” at the top of the page. This will scroll you down to the genotypes section.
- Click on “Lentil Genotypes” to access the Lentil Genotype Matrix Tool.

The screenshot shows the KnowPulse website interface. At the top, there is a header with the KnowPulse logo and navigation links for 'Sign in' and 'Create an Account'. Below the header, a banner reads 'Explore KnowPulse: a web-resource focused on diversity data for pulse crop improvement.' A navigation bar contains 'New to KnowPulse: Genetic Maps' and three dropdown menus: 'Phenotypes', 'Genotypes', and 'Germplasm'. The 'Genotypes' menu is circled in green, and a green arrow points from it down to the 'Genomic Data' section in the 'Data' category.

Crops

- Chickpea (*Cicer arietinum*)
- Lentil (*Lens culinaris*)
- Dry Bean (*Phaseolus vulgaris*)
- Faba Bean (*Vicia faba*)
- Pea (*Pisum sativum*)

Tools

- BLAST
- JBROWSE
- GENOTYPE MATRIX
- CVT JS
- VCF BULK EXPORT
- EXPERIMENTS
- TRAIT DISTRIBUTION PLOT
- GERMPLASM MATRIX
- USER CHARTS
- POSITION CONVERT
- VCF POSITION SEARCH
- MAP VIEWER

Data

- Phenotypes**
 - Phenotypic Data
 - Phenotypes
 - Trait Distribution Plot
 - Lentil Traits
- Genotype Calls**
 - Genomic Data
 - Sequence Variants
 - Markers
 - Lentil Genotypes
- Germplasm**
 - Germplasm
 - Registered Varieties
 - Genetic Maps
 - MapView Tool

* Current data includes sequence and genotypic data for Chickpea, Common Bean, Field Pea and Lentil.

1.2 Step #2: Find genotypic data for your reference germplasm

- Enter the name of each germplasm you are interested in by typing it in the textfield labelled germplasm. Then check the correct species is selected to the right of the textbox. To add more than one germplasm click the green + button.
- For this example, Enter “ILL 8006” in the top germplasm textfield. You will notice that it autocom-

pleted as you type.

- Then click the green + button and enter “CDC Milestone” in the second textfield which appears.
- Each time you click the green + button or search, the genotypic data for the listed germplasm will be shown.



[Home](#)

Lens Genotypes

- Choose germplasm you are interested in.**

Simply enter the name of one germplasm (e.g. "Eston AGL", "CDC Robin AGL", or "ILL 8007 AGL") of interest below and then click the green plus (+) button. You can enter any number of germplasm you are interested in and each will be added to the matrix as they are entered.

Germplasm

CDC Milestone

Germplasm/Stock Name

Specify the stock (and species of the stock) you want to display the genotypes of.

Polymorphic Variants

Between and

Restrict the variants to those that have different allele calls for the selected germplasm.
- Restrict to the region of the genome. (optional)**

If applicable, we recommend you filter to a given region of the genome to make the genotype set more manageable. For example, to see all of Lentil Chromosome 4 you would enter From "LcChr4" to "LcChr4" leaving the start and end position blank.

Genome Range

From to

The range of the genome you would like to display variants from. If you enter just the start or just the end position then all variants before or after that location, respectively, will be displayed.
- Additional Filter criteria. (optional)**

Download: Requires log in

Total Results?: 95,362
Unique Variants?: 39,436

Sort by Location, Variant Name

| Variant Name | Backbone | Start | End | ILL 8006 | CDC Milestone |
|--------------|----------|--------|--------|----------|---------------|
| LcCo1753p578 | LcChr1 | 5562 | 5562 | | A |
| LcCo1753p537 | LcChr1 | 5603 | 5603 | C | C |
| LcCo1753p60 | LcChr1 | 6081 | 6081 | G | C |
| LcCo4363p155 | LcChr1 | 362015 | 362015 | G | G |
| LcCo1392p800 | LcChr1 | 363602 | 363602 | C | T |
| LcCo1392p854 | LcChr1 | 363757 | 363757 | C | T |

Bioinformatic Tools

- ▼ BLAST
 - Nucleotide Query
 - BLASTn
 - BLASTx
 - Protein Query
 - BLASTp
 - tBLASTn
- ▼ JBrowse
 - Lentil (UofS, v1.2)
 - Chickpea (Kabuli, v1.0)
 - Chickpea (Desi, v2.0)
 - Colletotrichum lentis
 - Common Bean (Phytozome, v2.1)
 - Medicago (IMGAG, v4.0)
 - Pea (Caméor; v1a)
 - Soybean (Phytozome, Williams 82 v4.0)
- User-driven Charts
 - CVITjs Whole-Genome Visualization
 - Position Convert: Lco.8 to Lc1.2
- ▼ VCF Bulk Export
 - VCF Position Search

Genomic Data

- ▼ Sequence Variants
 - Markers
 - Lentil Genotypes

Phenotypic Data

- ▼ Phenotypes
 - Trait Distribution Plot
 - Lentil Traits

Germplasm

- ▼ Germplasm
 - Registered Varieties
- ▼ Genetic Maps
 - MapViewer Tool

1.3 Step #3: Restrict the Sequence Variants to polymorphic between your germplasm

- Underneath germplasm, there is a filter to restrict to polymorphic variants. This filter compares two germplasm and only shows variants with different genotypic calls.
- For our example, we would select ILL 8006 in the first drop down and CDC Milestone in the second drop-down to see only sequence variants with differing genotypes (i.e polymorphic variants) between these two germplasm.
- Click Search to see the results.



Home

Lens Genotypes

- Choose germplasm you are interested in.

Simply enter the name of one germplasm (e.g. "Eston AGL", "CDC Robin AGL", or "ILL 8007 AGL") of interest below and then click the green plus (+) button. You can enter any number of germplasm you are interested in and each will be added to the matrix as they are entered.

Germplasm

ILL 8006 Lens culinaris

CDC Milestone Lens culinaris

Germplasm/Stock Name Lens culinaris

Specify the stock (and species of the stock) you want to display the genotypes of.

Polymorphic Variants

Between ILL 8006 and CDC Milestone

Restrict the variants to those that have different allele calls for the selected germplasm.
- Restrict to the region of the genome. (optional)

If applicable, we recommend you filter to a given region of the genome to make the genotype set more manageable. For example, to see all of Lentil Chromosome 4 you would enter From "LcChr4" to "LcChr4" leaving the start and end position blank.

Genome Range

From Sequence Name Start Position to Sequence Name End Position

The range of the genome you would like to display variants from. If you enter just the start or just the end position then all variants before or after that location, respectively, will be displayed.
- Additional Filter criteria. (optional)

Download: Requires log in

Total Results?: 24,704
Unique Variants?: 8,486

Sort by Location, Variant Name

| Variant Name | Backbone | Start | End | ILL 8006 | CDC Milestone |
|---------------|----------|--------|--------|----------|---------------|
| LcC01753p60 | LcChr1 | 6081 | 6081 | G | C |
| LcC01392p800 | LcChr1 | 363602 | 363602 | C | T |
| LcC01392p854 | LcChr1 | 363757 | 363757 | C | T |
| LcC01392p920 | LcChr1 | 363823 | 363823 | C | T |
| LcC01392p932 | LcChr1 | 363835 | 363835 | T | C |
| LcC01392p935 | LcChr1 | 363838 | 363838 | G | T |
| LcC01392p1135 | LcChr1 | 364038 | 364038 | C | T |

Bioinformatic Tools

- BLAST
 - Nucleotide Query
 - BLASTn
 - BLASTx
 - Protein Query
 - BLASTp
 - tBLASTn
- JBrowse
 - Lentil (UoIS, v1.2)
 - Chickpea (Kabuli, v1.0)
 - Chickpea (Desi, v2.0)
 - Colletotrichum lentis
 - Common Bean (Phytozome, v2.1)
 - Medicago (IMGAG, v4.0)
 - Pea (Caméor; v1a)
 - Soybean (Phytozome, Williams 82 v4.0)
- User-driven Charts
- CVIT's Whole-Genome Visualization
- Position Convert: Lc0.8 to Lc1.2
- VCF Bulk Export
 - VCF Position Search

Genomic Data

- Sequence Variants
 - Markers
 - Lentil Genotypes

Phenotypic Data

- Phenotypes
 - Trait Distribution Plot
 - Lentil Traits

Germplasm

- Germplasm
 - Registered Varieties
- Genetic Maps
 - MapViewer Tool

1.4 Step #4: Restrict to you trait-implicated Region of the Genome.

- The second section of the filter criteria available for the genotype matrix allows you to enter the region of the genome you are interested in. Once you click search, the genotype matrix will only show sequence variants found in this region.
- In our example, the region of interest is LcChr4:1875000-2075000. To enter this we place LcChr4 for the Sequence Name, 1875000 for the start position and 2075000 for the end position.



Home

Lens Genotypes

1

Choose germplasm you are interested in.

Simply enter the name of one germplasm (e.g. "Eston AGL", "CDC Robin AGL", or "ILL 8007 AGL") of interest below and then click the green plus (+) button. You can enter any number of germplasm you are interested in and each will be added to the matrix as they are entered.

Germplasm

ILL 8006

Lens culinaris

×

CDC Milestone

Lens culinaris

×

Germplasm/Stock Name

Lens culinaris

+

Specify the stock (and species of the stock) you want to display the genotypes of.

Polymorphic Variants

Between

ILL 8006

and

CDC Milestone

Restrict the variants to those that have different allele calls for the selected germplasm.

2

Restrict to the region of the genome. (optional)

If applicable, we recommend you filter to a given region of the genome to make the genotype set more manageable. For example, to see all of Lentil Chromosome 4 you would enter From "LcChr4" to "LcChr4" leaving the start and end position blank.

Genome Range

From

LcChr4

to

LcChr4

The range of the genome you would like to display variants from. If you enter just the start or just the end position then all variants before or after that location, respectively, will be displayed.

3

Additional Filter criteria. (optional)

Search

Download: Requires log in

Total Results?: 15

Unique Variants?: 10

Sort by Location, Variant Name

| Variant Name | Backbone | Start | End | ILL 8006 | CDC Milestone |
|--------------|----------|---------|---------|----------|---------------|
| LcC17900p664 | LcChr4 | 1914565 | 1914565 | T | G |
| LcC17900p83 | LcChr4 | 1914646 | 1914646 | T | C |
| LcC17900p577 | LcChr4 | 1914652 | 1914652 | T | C |
| LcC17900p520 | LcChr4 | 1914709 | 1914709 | A | G |
| LcC05929p69 | LcChr4 | 1961110 | 1961110 | AA | G |
| LcC23917p331 | LcChr4 | 1985063 | 1985063 | T | C |
| LcC00132p450 | LcChr4 | 2043711 | 2043711 | T | A |
| LcC00132p312 | LcChr4 | 2043849 | 2043849 | T | C |
| LcC00132p295 | LcChr4 | 2043866 | 2043866 | G | A |
| LcC00132p206 | LcChr4 | 2045002 | 2045002 | C | T |

« first 100

« previous

LcChr4: 1914564-2045002

next »

last 100 »

Bioinformatic Tools

▼ BLAST

▼ Nucleotide Query

• BLASTn

• BLASTx

▼ Protein Query

• BLASTp

• tBLASTn

▼ JBrowse

• Lentil (UofS, v1.2)

• Chickpea (Kabuli, v1.0)

• Chickpea (Desi, v2.0)

• Colletotrichum lentis

• Common Bean (Phytozome, v2.1)

• Medicago (IMGAG, v4.0)

• Pea (Caméor; v1a)

• Soybean (Phytozome, Williams 82 v4.0)

► User-driven Charts

► CVITjs Whole-Genome Visualization

• Position Convert: Lc0.8 to Lc1.2

▼ VCF Bulk Export

• VCF Position Search

Genomic Data

▼ Sequence Variants

• Markers

• Lentil Genotypes

Phenotypic Data

▼ Phenotypes

• Trait Distribution Plot

• Lentil Traits

Germplasm

▼ Germplasm

• Registered Varieties

▼ Genetic Maps

• MapViewer Tool

Our Research

• Home

• Bioinformatics Resource

• Student Information

• Job Postings

1.5 Step #5: (Optionally): Restrict to specific variants.

- Say you are interested in a specific set of variants and would like to see that subset.
- You can enter the specific variant names by expanding the Additional Filter criteria section then clicking Search.



[Home](#)

Lens Genotypes

1 Choose germplasm you are interested in.

Simply enter the name of one germplasm (e.g. "Eston AGL", "CDC Robin AGL", or "ILL 8007 AGL") of interest below and then click the green plus (+) button. You can enter any number of germplasm you are interested in and each will be added to the matrix as they are entered.

Germplasm

| | | | |
|----------------------|-----------------------|----------------|----------------------------------|
| ILL 8006 | <input type="radio"/> | Lens culinaris | <input type="button" value="x"/> |
| CDC Milestone | <input type="radio"/> | Lens culinaris | <input type="button" value="x"/> |
| Germplasm/Stock Name | <input type="radio"/> | Lens culinaris | <input type="button" value="+"/> |

Specify the stock (and species of the stock) you want to display the genotypes of.

Polymorphic Variants

Between and

Restrict the variants to those that have different allele calls for the selected germplasm.

2 Restrict to the region of the genome. (optional)

If applicable, we recommend you filter to a given region of the genome to make the genotype set more manageable. For example, to see all of Lentil Chromosome 4 you would enter From "LcChr4" to "LcChr4" leaving the start and end position blank.

Genome Range

From to

The range of the genome you would like to display variants from. If you enter just the start or just the end position then all variants before or after that location, respectively, will be displayed.

3 Additional Filter criteria. (optional)

We recommend you fill out as many of the following optional filters as possible to narrow the genotype set to those you are most interested in.

Variant Name(s)

LcC00132p450
LcC00132p312
LcC00132p295
LcC00132p206

A list of variant names you wish to see genotypes for with one variant per line.

Project Name

The name of the project you want to restrict genotypes to.

Variant Type

- Choose One to Filter -

The types of variants you would like to see genotypes for (e.g. indels only).

Marker Type

- Choose One to Filter -

The types of markers you would like to see genotypes for (e.g. exome capture).

Download: [Requires log in](#)

Total Results?: 7
Unique Variants?: 4

Sort by Location, [Variant Name](#)

| Variant Name | Backbone | Start | End | ILL 8006 | CDC Milestone |
|--------------|----------|---------|---------|----------|---------------|
| LcC00132p206 | LcChr4 | 2045002 | 2045002 | C | T |
| LcC00132p295 | LcChr4 | 2043866 | 2043866 | G | A |
| LcC00132p312 | LcChr4 | 2043849 | 2043849 | T | C |
| LcC00132p450 | LcChr4 | 2043711 | 2043711 | T | A |

« first 100 « previous LcChr4: 2045001-2043711 next » last 100 »

Bioinformatic Tools

- ▼ BLAST
 - Nucleotide Query
 - BLASTn
 - BLASTx
- ▼ Protein Query
 - BLASTp
 - tBLASTn
- ▼ JBrowse
 - Lentil (UofS, v1.2)
 - Chickpea (Kabuli, v1.0)
 - Chickpea (Desi, v2.0)
 - Colletotrichum lentis
 - Common Bean (Phytozome, v2.1)
 - Medicago (IMGAG, v4.0)
 - Pea (Cam60r; v1a)
 - Soybean (Phytozome, Williams 82 v4.0)
- User-driven Charts
- CVITjs Whole-Genome Visualization
- Position Convert: Lc0.8 to Lc1.2
- ▼ VCF Bulk Export
 - VCF Position Search

Genomic Data

- ▼ Sequence Variants
 - Markers
 - Lentil Genotypes

Phenotypic Data

- ▼ Phenotypes
 - Trait Distribution Plot
 - Lentil Traits

Germplasm

- ▼ Germplasm
 - Registered Varieties
- ▼ Genetic Maps
 - MapViewer Tool

Our Research

- Home
- Bioinformatics Resource
- Student Information
- Job Postings
- Field Lab
- Projects
- Publications
- Workshops
- ▼ Research
 - Breeding & Genetics
 - Pathology
 - Physiology
 - Tissue Culture
- ▼ Crops
 - Chickpea
 - Dry Bean
 - Faba Bean
 - Lentil
 - Pea
 - All Species of Interest

CHAPTER 2

Developers

Warning: This guide is meant for the KnowPulse Development Team.

This guide is meant for the KnowPulse team to share information and document procedures. That said, it may be helpful for anyone developing a Tripal site -we are welcoming to all developers and welcome suggestions and fixes to our documentation.

2.1 Developer Workflow

This document describes the workflow that the KnowPulse team follows for development, which depends on the use of GitHub features and the [ZenHub Agile Project Management Extension for GitHub](<https://www.zenhub.com/>).

All Tasks, Bugs, Features, etc. should be GitHub Issues

- Create an issue on github: **Be Descriptive!**
- Label the issue (bug? enhancement?)
- Using Boards, indicate if this is a long term goal (icebox) or if you intend to do it right away (Up Next)
- If you know who will be doing work on this issue then assign it to them.
- Keep issues to reasonable amounts of work. A good rule of thumb is that if you think the issue would take more than a week of development time to complete, then the scope of work is not reasonable for a single issue ;-).
- Also keep in mind, the smaller the issue the easier it is to review and the faster it will be released.

2.1.1 Workflow

1. Choose an issue to work on. Bugs should take priority. Look first at issues assigned to you.
2. Once you've chosen a task, assign it to yourself and move it to "In Progress" using Boards.

3. Create a local branch and push it to github `_before_` making a commit. Use the format *[issue number]-[short description]* when you name the branch.
 - Create local branch: `git checkout -b [issue number]-[short description]`
 - Push it to github: `git push origin [issue number]-[short description]`
 - Any changes you will make in completing this issue should now be done within this branch.
4. Work on the issue.
 - Commit often to your local branch using descriptive commit messages.
 - Push to github at least twice a week!
 - Also, try to keep a record of design decisions and approach to solving the issue by commenting on the issue itself. The more context that you provide, the less you will have to explain to your co-workers during review ;-)
5. Once work is complete, signal a Review - Create a pull request between your branch and the master branch
 1. On the GitHub repository page, click on the “Pull Requests” tab and click the “New Pull Request” button.
 2. Leave `base:master`; change `compare:master` to point to your branch created in step 3.
 3. The page immediately shows the changes between your branch and master. Create the pull request by clicking the “Create Pull Request” button.
 4. Link the pull request to the issue
 5. Describe the original issue, how you fixed it and how to test your code: what pages to look at, etc.
 - Request all team members review the pull request
 - In Boards, move the issue to the “Review” pane
6. Review
 - Give your team members a day or two to look over your code
 - Answer any questions they have
 - Wait to implement suggestions until the review process is completed by everyone
 - Reviewers should indicate whether they approve your changes or not.
7. All reviewers approved your changes?
 - If no, then go back to step 4.
 - Make sure to address all concerns.
 - If you are uncertain their approach is the best one, then **defend yours**. Ask for clarification on any suggestions if necessary.
 - Keep in mind, you do not need to create a new pull request after addressing concerns. The pull request will automatically update as you make changes to the branch created in step 3. That is why you are doing this work in a issue-specific branch.
 - If yes, then move onto step 8
8. Close the issue
 - Done by lead Reviewer (Once all issues/suggestions are addressed)
 - Lead Reviewer is (1) Lacey or (2) the person who did the review if Lacey is gone.
 - Pull request is merged into the master branch

1. Click “Conversation” tab at the top of the pull request
 2. Scroll down, click “Merge pull request” assuming there are no conflicts with the base branch. If there are conflicts then those should be fixed by the person who submitted the pull request.
 3. After you will see “Pull request successfully merged and closed” with a “Delete Branch” button. Click “Delete Branch”.
- Issue is closed
 1. The issue should be referenced in the original description of the pull request -go to it.
 2. Click “Close Issue”.
 - New issue is created to update module on KnowPulse
 1. Go to <https://github.com/UofS-Pulse-Binfo/KnowPulse>
 2. Create an issue specifying that module XXX has received either new features or bug fixes and needs to be updated. Reference the issue linked to the pull request by entering [organization]/[repo]#[issue number] (e.g. UofS-Pulse-Binfo/tripal_daemon#3)
 3. Assign this issue to whomever is in charge of updating KnowPulse (e.g. Lacey).

2.1.2 Review

- When reviewing, look at the code changes directly _and_ test through the interface.
 1. To see all the changes, click on “Files Changed” at the top of the pull request.
 2. To start a code review, hover in front of the first line you would like to comment on then click the blue + icon that shows up. This opens a comment box where you can enter your suggestion/critique. Then click the green “Start review” button.
 3. Continue adding comments throughout the changes in this manner.
- If you don’t actually have code review comments to make, you can submit your review by:
 1. Click on “Files Changed” at the top of the pull request.
 2. Click on “Review Changes” button at the top right of the page.
 3. In the resulting popup, enter your comments and select whether you approve or request changes.
- If you simply need clarification, just comment on the pull request with your questions and hold off reviewing.
- For interface functionality or design critique/suggestions, enter a comment (preferably with screenshot) in the “Conversation” tab of the pull request.
- You may need to checkout the branch on a development site to adequately test the code.
 1. *git fetch origin* will fetch all branches from that repository
 2. *git checkout -b [branch] origin/[branch]* will pull the code from the branch which you want to test
 3. Once you’ve finished testing, you can revert to the master branch with *git checkout master*

2.2 KnowPulse Content Types

The following breaks down our content types based on the primary Chado table storing their data.

- chado.organism table (full table)

- Species (OBI:organism)
- chado.project table (seperated using projectprop)
 - Research Project (semantic science ontology; study; SIO:001066)
 - Genome Project (local:genome)
- chado.publication (full table)
 - Publication (TPUB; Publication)
- chado.feature table (seperated using feature.type_id)
 - Variant (Sequence Ontology; sequence_variant)
 - Genetic Marker (Sequence Ontology; genetic_marker)
 - QTL (Sequence Ontology; QTL; SO:0000771)
- chado.stock (seperated using stock.type_id)
 - Germplasm Accession (Crop ontology; accession; CO_010:0000044)
 - Breeding Cross (local:F1)
 - Registered Varieties (Sequence Ontology; cultivar; CO_010:0000029)
 - Recombinant Inbred Line (local:Recombinant Inbred Line)
 - Biological Sample (biological sample; sep:00195)
- chado.cvterm (specific cv_id)
 - Chickpea Trait (CO_338:ROOT)
 - Common Bean Trait (CO_335:ROOT)
 - Lentil Trait (CO_339:ROOT)
- chado.featuremap (full table)
 - Genetic Map (data onotology; Map)
- chado.stock_collection (full table)
 - Stock Collections (don't know what term to use yet)

2.2.1 Species (Organism)

The organism content type will include each of our crops and wild relatives. Stored in the Chado Organism table. Design-wise we are thinking of these as launch pads which both summarize the data per crop and link to pre-filtered searches.

Genome Pane

- @todo Current Assembly: This field should provide information on the most recent genome assembly or assemblies for this species.
- @todo Genome Browser: This field should summarize the data in the current JBrowse and have an image linking to the JBrowse instance for the current assembly
- @todo Whole-Genome Diagram: This field should show an embedded CViTjs diagram summarizing the current assembly's annotation and markers.

- @todo Past Assemblies: This field should list the past assemblies with links to archived information.

Genetic Markers Pane

- Types: This field summarizes the types of markers available for this organism.
- Quick Search: This field provides a simple keyword search with limited advanced search collapsed. When the user clicks search, they are redirected to the marker search page with their results pre-filtered.

Genetic Maps & QTL Pane

- @todo Tiled display of available maps with each tile containing map metadata such as population, date generated, number of markers, QTL list
- @todo QTL Quick Search: simple keyword search for QTLs.

Germplasm Pane

- Types: This field summarizes the types of germplasm available for this organism. Should this be subject to permissions?
- Quick Search: This field provides a simple keyword search. No need for advanced search or breakdown by type.
- @todo Recent UofS Varieties: a list of the most recent varieties.
- @todo Variety Spotlight: Highlight a single variety
- @todo RIL Populations: This could be a cool search that lets them select things like interspecific, species, phenotype of interest. Essentially allow people to search for a RIL without knowing it's number
- @todo Germplasm Diversity: Show a map with each of our germplasm indicated.
- @todo Wild Relatives: summarize the gene pools and/or phylogeny

Phenotypes

- @todo Quick Search: This field provides a simple keyword search. No need for advanced search? It would be helpful this field searched both names and linked cvterms.
- @todo Phenotype Projects: Highlight our biggest and/or most recent phenotyping projects. Should show meta-data for each such as population phenotyped, purpose, siteyears w/ data, funders logos if applicable. Links to project pages.

2.2.2 Research Project

Currently just a regular old project page. Stored in the Chado Project table. Need a way to break out? What about datasets which are currently projects with data? These pages should include all metadata about the project and summarize all the data generated by it.

Warning: Not yet available as a Tripal3 content type. Currently in progress.

Overview Pane

- Project meta-data (stored as project properties)
 - Years
 - Collaborators
 - Funders
 - Locations
 - Measurements/Traits being collected
 - Genotyping being done?
 - Germplasm collection
- Abstract
- Methods

Genomic Data

- Barchart or infographic summarizing the types and amount of genomic data (sequence, genes, etc.) generated as part of this project.
- Assembly information if generated as part of this project?

Genotypic Data

- Barchart or infographic summarizing the types and amount of genotypic data (markers, variants, germplasm assayed, genotype calls, etc.) for this project.
- Genotypes Search: Redirects to the genotype matrix pre-filtered for the current project. Provides form elements to further restrict the matrix.

Phenotype Data

- Barchart or infographic summarizing the siteyears, traits and germplasm phenotyped for this project.

Germplasm

- List the germplasm associated with this project with download. It might be a long list so make it searchable. Also, we should support stock collections which would then be listed at the top. For example, AGILE is using the Lentil Diversity Panel so it would be nice to say that at the top so users know it's not a random list ;-).
- We also need a way to add germplasm to a project through the form so keep in mind this one will need a form element.

2.2.3 Publication

Stored in the Chado Publication table. These are really basic pages meant to provide attribution to data in KnowPulse. As such **All panes will be open by default.**

Warning: Not yet available as a Tripal3 content type. Currently in progress.

Overview

- citation
 - this should be really obvious. Currently highlighted in green.
- meta-data (stored as project properties)
 - date published
 - journal
 - authors
- abstract
- obvious link to full text

Data

- associated data?
- maybe in a bubble/circle chart?

2.2.4 Genome Project

The plan is to store the genome assembly as collection of analysis records (one for each program ran to create the assembly) grouped together by this project.

Warning: Not yet available as a Tripal3 content type.

Overview

Open by Default

- Publication (prominent) with link
- Meta-data about the assembly
 - Scaffolds: size + number
 - Contigs: size + number
 - Scaffold N50/L50
 - Contig N50/L50
- Meta-data about the annotation

Assembly Details

- How was the assembly generated?
 - paragraph description
 - a list of all the analysis' records with versions, etc.
- How was it annotated?
 - paragraph description
 - a list of all the analysis' records with versions, etc.

Contacts

- List of contacts for who will answer questions

Whole Genome Browsing

Open by Default

- Genome Browser
 - summarize the data in the current JBrowse
 - have an image linking to the JBrowse instance for this assembly
- Whole-Genome Diagram
 - show an embedded CViTjs diagram summarizing this assembly's annotation and markers.

BLAST Search

- BLAST quick search where database is this assembly

Bulk Data

- Files for download associated with this genome

Restrictions on dataset usage

Open by Default

- any agreements, etc.

2.2.5 Variant

This page indicates a point of known variation in the genome. This single point may be assayed by many genetic markers. Stored in the Chado Feature table broken out by type_id.

Overview

- **Meta-data**
 - Name
 - Type
 - Species
 - Gene name and location in gene (if within gene)

Variant Locations

- variant locations by assembly

Genetic Markers

- List all the genetic markers assaying this variant

Genotypic Data

- pie chart summarizing the known alleles
- Genotype look-up field: mini genotype matrix for a single variant. Germplasm are the rows.
- Genotypes Matrix Link: Redirects to the genotype matrix pre-filtered for the current variant.

Sequence

- Variant Marked-up Sequence (FASTA format) -state assembly? for all locations

Genome View

- JBrowse field located on this marker

2.2.6 Genetic Marker

The genetic marker page is currently a customized feature page. Stored in the Chado Feature table broken out by type_id.

Overview

- **Meta-data**
 - Name
 - Marker technology
 - Species
 - @todo Gene name and location in gene (if within gene)
 - @todo any additional meta-data helpful for running the marker (e.g. primers)

Marker Locations

- marker locations by assembly (@todo not yet by assembly)

Genotypic Data

- pie chart summarizing the known alleles
- @todo Genotype look-up field: mini genotype matrix for a single variant. Germplasm are the rows.
- @todo Genotypes Matrix Link: Redirects to the genotype matrix pre-filtered for the current marker.

Sequence

- Variant Marked-up Sequence (FASTA format) (@todo -state assembly?) for all locations

Genome View

- @todo JBrowse field located on this marker

Genetic Maps

- @todo List all the genetic maps containing this marker

QTL

- @todo List of all QTL this marker falls within?

2.2.7 Germplasm Accession

Stored in the Chado Stock table broken out by type_id.

| |
|--|
| Warning: Not yet available as a Tripal3 content type. |
|--|

Overview

- Basic information such as name, accession, what-not. This will remain a default Tripal pane
- It would be great to finally have the breeder short-hand for the pedigree! This could be a separate field from the pedigree diagram so I can put it in the overview.

Parental Pedigree

- Shows the parentage pedigree diagram with all but the first 3 levels collapsed.
- All diagrams should now include a figure legend.

Genotypes Pane

- Types: This field summarizes the number of markers per type with calls for the current germplasm.
- Genotype Look-up field: a mini-genotype matrix filtered to this germplasm only. User enters a marker or region to display.
- Genotype Matrix Link: Redirects to the genotype matrix pre-filtered for the current germplasm.

Breeding Markers

- List the calls for a set of CORE breeding markers. Make the set configurable at the field level.

Phenotypes Pane

- List the value for a set of CORE phenotypes (e.g. seed coat, cotyledon color). Make the set configurable at the field level. Should it be entereable on the form? How do we handle multiple experiments, siteyears for a single trait/variety combo.

Relationships

- list any non-parental/progeny relationships.

2.2.8 Cultivated Variety

Stored in the Chado Stock table broken out by type_id. See <https://www.crop-diversity.org/mgis/accession/01BEL084245>

Warning: Not yet available as a Tripal3 content type.

Overview

- Basic information such as name, accession, what-not. This will remain a default Tripal pane
- It would be great to finally have the breeder short-hand for the pedigree! This could be a separate field from the pedigree diagram so I can put it in the overview.
- specific variety information
- link to registration pub
- abstract from pub

Parental Pedigree

- Shows the parentage pedigree diagram with all but the first 3 levels collapsed.
- All diagrams should now include a figure legend.

Genotypes Pane

- Types: This field summarizes the number of markers per type with calls for the current germplasm.
- Genotype Look-up field: a mini-genotype matrix filtered to this germplasm only. User enters a marker or region to display.
- Genotype Matrix Link: Redirects to the genotype matrix pre-filtered for the current germplasm.

Breeding Markers

- List the calls for a set of CORE breeding markers. Make the set configurable at the field level.

Phenotypes Pane

- List the value for a set of CORE phenotypes (e.g. seed coat, cotyledon color). Make the set configurable at the field level. Should it be entereable on the form? How do we handle multiple experiments, siteyears for a single trait/variety combo.
- Phenotype pane should highlight phenotypes specifically of interest for this cultivar.

Relationships

- list any non-parental/progeny relationships.

2.2.9 Breeding Cross

Stored in the Chado Stock table broken out by type_id.

Warning: Not yet available as a Tripal3 content type.

Overview

- Basic information such as name, accession, what-not. This will remain a default Tripal pane
- Overview will include information about the crossing block
- It would be great to finally have the breeder short-hand for the pedigree! This could be a separate field from the pedigree diagram so I can put it in the overview.

Crossing Block

- year, season,
- number of crosses made in the same crossing block

Parental Pedigree

- Shows the parentage pedigree diagram with all but the first 3 levels collapsed.
- All diagrams should now include a figure legend.

Progeny

- list of progeny the current cross was a parent for (includes crossing block info).

Genotypes Pane

- Types: This field summarizes the number of markers per type with calls for the current germplasm.
- Genotype Look-up field: a mini-genotype matrix filtered to this germplasm only. User enters a marker or region to display.
- Genotype Matrix Link: Redirects to the genotype matrix pre-filtered for the current germplasm.

Breeding Markers

- List the calls for a set of CORE breeding markers. Make the set configurable at the field level.

Phenotypes Pane

- List the value for a set of CORE phenotypes (e.g. seed coat, cotyledon color). Make the set configurable at the field level. Should it be entereable on the form? How do we handle multiple experiments, siteyears for a single trait/variety combo.

Relationships

- list any non-parental/progeny relationships.

2.2.10 Recombinant Imbred Line

Stored in the Chado Stock table broken out by type_id.

Warning: Not yet available as a Tripal3 content type.

Overview

- Basic information such as name, accession, what-not. This will remain a default Tripal pane
- RIL Specific information
 - the original cross
 - parents,
 - latest generation,
 - status
 - number of individuals.
- It would be great to finally have the breeder short-hand for the pedigree! This could be a separate field from the pedigree diagram so I can put it in the overview.
-

RIL Development Pane

- Reiterate the type of cross (Interspecific, intraspecific)
- Indicates the parents of the RIL including their species.
- Development timeline including which years & how many individuals
- If the RIL is complete, indicate that and the final number of individuals.

Parental Pedigree

- Shows the parentage pedigree diagram with all but the first 3 levels collapsed.
- All diagrams should now include a figure legend.

Genotypes Pane

- Types: This field summarizes the number of markers per type with calls for the current germplasm.
- Genotype Look-up field: a mini-genotype matrix filtered to parents of this RIL. User enters a marker or region to display.
- Genotype Matrix Link: Redirects to the genotype matrix pre-filtered for the parents of this RIL.
- Links to the VCF Filter form for the current RIL to facilitate downloading of the data.

Breeding Markers

- List the calls for a set of CORE breeding markers. Make the set configurable at the field level.

Phenotypes Pane

- List the value for a set of CORE phenotypes (e.g. seed coat, cotyledon color). Make the set configurable at the field level. Should it be enterable on the form? How do we handle multiple experiments, siteyears for a single trait/variety combo.
- Phenotype pane should highlight phenotypes specifically of interest for this RIL.

Relationships

- list any non-parental/progeny relationships.

2.2.11 Stock Collection

Warning: Not yet available as a Tripal3 content type.

Overview Pane

- paragraph describing the population
- development/source information

Genomic Data

- Barchart or infographic summarizing the types and amount of genomic data (sequence, genes, etc.) collected for this collection.

Genotypic Data

- Barchart or infographic summarizing the types and amount of genotypic data (markers, variants, germplasm assayed, genotype calls, etc.) for this collection.
- Genotypes Search: Redirects to the genotype matrix. Provides form elements to choose germplasm from this collection.

Phenotype Data

- Barchart or infographic summarizing the siteyears, traits and germplasm phenotyped for this collection.

Germplasm

- List the germplasm associated with this collection with download. It might be a long list so make it searchable.

2.3 KnowPulse Field Types

The following sections described common field types.

2.3.1 Quick Search (Should be in Core?)

- This field type allows a user to search for additional content quickly and easily.
- It consists of a single keyword search box and optional, configurable advanced search form.
- Configuration: - Search URL this field redirects to - what key to use in the query string for entered keywords - Drupal Form API definition of advanced search elements where the keys of the elements are used in the query string - Description and/or help text to display to the user - Quick Search Title

2.3.2 Summary Field (Should be in Core?)

- This field displays a graphical summary based on a materialized view.
- Configuration: - Select the Materialized View to get the summary from - Select which mview column maps to the legend (e.g. type name) - Optional filter by field: select field and mview column mapping - Select chart type: Pie, bar, etc. - Figure Title - Figure Description - Chart type specific configuration

2.3.3 Embedded JBrowse by organism

- Embeds a JBrowse instance
- Provided by the JBrowse integration module
- Contains linkout to full JBrowse page for more intensive browsing

- Configuration: - Select Organism Field for the current content type - Show/Hide Menubar, search, track selector
- For each organism: select jbrowse, tracks, and location to show

2.3.4 Whole-Genome View by organism

- Embeds a CViTjs diagram
- Provided by the CViTjs integration module
- Contains linkout to full diagram
- Configuration: - Select Organism Field for the current content type - Show/Hide Controls, track selector - For each organism: select diagram and tracks to show

2.4 Task-based Tutorials

2.4.1 Update KnowPulse

The following documentation outlines the steps and testing that should be done when updating KnowPulse, whether it be simply updating a custom module or updating Drupal itself. Updating between major versions (e.g. Drupal 7 => 8 or Tripal 2 => 3) is beyond the scope of this document.

The following are the basic steps to be taken. There is more information below if needed.

1. Create a fresh clone (Dev Fresh) of the production site
2. Upgrade the clone.
 - In the directory for the site, execute *drush up*. This will tell you what modules need to be updated and prompt you to do so. Remember, if you don't have permission on the directory you will need to use *sudo*.
3. Test the upgrade by checking the pages listed under "To be Checked" below.
4. If there are no errors (check the current page, the recent log messages and the apache error log) then upgrade the production site using the same steps.
 - Recent Log Messages: <http://knowpulse.usask.ca/portal/admin/reports/dblog>
 - Apache Error Log: `/var/log/apache2/error.log` (requires *sudo*).
5. Check the pages listed under "To be Checked" on the production site as well.

Pages To Be Checked

This list should be ever expanding. Feel free to document the specifics under each page.

- Home Page: <http://knowpulse.usask.ca/>
- BLAST: <http://knowpulse.usask.ca/portal/blast/nucleotide/nucleotide>
 - You should actually execute an example blast to ensure that the functionality works.
- JBrowse: <http://knowpulse.usask.ca/portal/jbrowse/Lentil>
 - Ensure you have your cache killer turned on!
 - Make sure to turn on/off tracks, navigate side to side and search.
- Views Search: <http://knowpulse.usask.ca/portal/search/variants>

- Check multiple combinations of filters (e.g. genus:Lens, type:SNP, name:p390).
- If you get no results and see a green message as follows, you likely need to re-apply the Tripal Views patch.

Warning: > Exception in KP Variant Search[kp_variant_search]: SQLSTATE[42P01]: Undefined table: 7 ERROR: relation "feature" does not exist > LINE 5: feature feature

- To apply views patch, navigate to views module directory and execute *git apply ../tripal/tripal_views/views-sql-compliant-three-tier-naming-1971160-30.patch*
- Feature Page: <http://knowpulse.usask.ca/portal/feature/lens-culinaris/SNP/LcC00002p390>
 - Check the genotypes & sequence panes especially.
- Raw Phenotypes
 - Chart: <http://knowpulse.usask.ca/portal/phenotypes/raw>
 - Backup: <http://knowpulse.usask.ca/portal/phenotypes/raw/backup>
- Germplasm: <http://knowpulse.usask.ca/portal/stock/Lens/culinaris/Variety/KP%3AGERM58>
 - Check that you can collapse nodes & follow links on the pedigree
- Research Category: <http://knowpulse.usask.ca/portal/node/21>
 - Check that there are projects listed & pager works.

Create a Clone of KP

Note: Replace *** with the name of the database user in all the commands below.

1. Backup the production site to your home directory

- `pg_dump kp3_production --user *** --host thunder > ~/kp3_production.2017Aug18.sql`

2. Remove the existing Clone. This should be safe since no one is supposed to develop on the clone. That said, it wouldn't hurt

- `sudo rm -R dev/fresh`
- `psql --user *** --host thunder --command "DROP DATABASE kp3_fresh"`

3. Re-create the clone database and restore the production site backup to it.

- `psql --user *** --host thunder --command "CREATE DATABASE kp3_fresh WITH OWNER ***"`
- `psql kp3_fresh --user *** --host thunder < ~/kp3_production.2017Aug18.sql`

4. Copy the production site files to dev/fresh.

- `sudo cp -R ../portal fresh`
- `sudo chown -R lacey-local:www-dev fresh/`

5. Change the database in the dev/fresh/sites/default/settings.php from kp3_production to kp3_fresh.

- You need to use sudo to change this file.

6. Change the write permission of dev/fresh/sites/default/files to be writable by all.

- `sudo chmod 777 files/`

2.4.2 Add new Genome Build to KP

Overview

The following list is a general overview. More details will be added below as needed.

1. Create a blast database and register it with the Tripal Blast UI module.
2. Create a JBrowse instance and register it with the Tripal JBrowse module.
3. Setup bulk download, if desired.

Blast

1. Prefix the record names in your FASTA file using [this GIST](#). The naming convention is The first letter of the Genus and two letters from the species followed by the version and an underscore. For example, *Lens culinaris* 1.0 would have `Lcul.0_` added to the beginning of each record name.

```
> perl prefix_fasta.pl [original fasta] [new fasta] [prefix]
```

2. Create the BLAST indices.

```
> formatdb -p F -t Genus_species_version_genome -i Genus_species_version_genome.fasta -n Genus_species_version_genome -o T
```

```
> makeblastdb -dbtype 'nucl' -input_type 'fasta' -title Genus_species_version_genome -in Genus_species_version_genome.fasta -parse_seqids -hash_index
```

3. Create BLAST database on KnowPulse. Go to Admin Toolbar > Content > Add Content > Blast Database. Convention for naming is Genus species: Genome version.

Note: File Location: Both the prefixed FASTA and the associated indices should be located in the `blast_dbs` folder with all the other KnowPulse blast databases. If you are uncertain where that is, simply check the page for an existing blast database to see it's path.

Note: Restrict Access: You can restrict the users which can blast against a specific blast database by editing the page created in step #3 above. Near the bottom there is a section for `Access` in the horizontal tabs. Simple select the group you would like to give access (usually `University of Saskatchewan`) and all users not in that group will not see that blast database as an option.

Warning: More information on Link-outs to come.

JBrowse

1. Setup a new JBrowse following the instructions here: <https://jbrowse.org/install/>. The folder containing the JBrowse files should be named `[random 5 letters]-[genome prefix]`.

- Use the prefixed fasta file created for the BLAST as the backbone.
 - Make sure to update any annotation so the backbone/chromosome feature names match.
2. Add KnowPulse themeing to the new JBrowse by copying the CSS files from an existing JBrowse. **This needs to be improved. Ideally all KnowPulse specific changes would be in a single CSS file added to the index.**
 3. Edit `jbrowse.conf` to change the Share link to point to the KnowPulse path by adding the following to the [General] section.

```
## Set the share link to respect being within an iFrame.
shareURL=function(browser){ return 'http://knowpulse.usask.ca/portal/[Instance Path]/?
↪loc='+browser.view.visibleRegionLocString()+'&tracks='+ (browser.view.
↪visibleTrackNames().join(',')); }
```

Note: Notice the [Instance Path] in the URL above; this should be changed to the path selected in step 4. Use existing `jbrowse.conf` as an example for other configuration changes.

4. Register the JBrowse with KnowPulse by creating a JBrowse Instance. Go to Admin Toolbar > Content > Add Content > JBrowse Instance and fill in the requested information. Also, change the URL path settings to provide an alias for the JBrowse page. Conventions are as follows:
 - Cultivated crops, current version: [Common Name]
 - Cultivated crops, archived version: [Common Name]/[version]
 - Wild Species, current version: [Genus species]
 - Wild Species, archived version: [Genus species]/[version]

2.5 Tool Customization

The following documents discuss how we customize third-party tools for better integration with KnowPulse in an effort to provide a consistent user experience.

2.5.1 JBrowse

We have changed the colours of our JBrowse to make it match KnowPulse for more seamless integration. The following shows our customizations:

```
/* Change the color of the upper trapezoid */
#GenomeBrowser.jbrowse div.locationTrap {
  border-bottom-color: #17313E
}
#GenomeBrowser.jbrowse div.locationThumb {
  border: 2px solid #17313E;
}

/* Change the color/Background of the top menu bar */
#GenomeBrowser.jbrowse .menuBar {
  background: url('http://knowpulse.usask.ca/portal/sites/all/themes/kptheme/images/
↪mainbarbg.jpg') repeat-x;
  text-align: right;
}
```

(continues on next page)

(continued from previous page)

```
/** Customize HTML Features */
#GenomeBrowser.jbrowse .feature2,
  #GenomeBrowser.jbrowse .plus-feature2,
  #GenomeBrowser.jbrowse .minus-feature2 {
    height: 7px;
    background-color: #62d335;
  }
#GenomeBrowser.jbrowse .match_part,
  #GenomeBrowser.jbrowse .plus-match_part,
  #GenomeBrowser.jbrowse .minus-match_part {
    height: 10px;
    background-color: #00FF00;
  }
```

2.6 ReadtheDocs Guide

Documentation for our generic modules should be written in **Restructured Text**, compiled with **Sphinx**, and built/hosted with **ReadTheDocs**. This directory, when compiled, is hosted at [https://\[yourmodule\].readthedocs.io/en/latest/](https://[yourmodule].readthedocs.io/en/latest/)

For minor changes, you can simply [Edit the file using the Github editor](#), which will allow you to make a Pull Request. Once approved, your changes will be reflected in the documentation automatically!

2.6.1 Install Sphinx

For minor changes, you don't need to build the documentation! If you want to see how your changes will look on the built site, however, you will need Sphinx installed. Sphinx packages are published on the Python Package Index. The preferred tool for installing packages from PyPI is pip. This tool is provided with all modern versions of Python.

On Linux or MacOS, you should open your terminal and run the following command.

```
$ pip install -U sphinx
```

After installation, type `sphinx-build --version` on the command prompt. If everything worked fine, you will see the version number for the Sphinx package you just installed.

For more information, please see the Sphinx setup guide: <http://www.sphinx-doc.org/en/master/usage/quickstart.html>

2.6.2 Building your changes

For more extensive edits, or when contributing new guides, you should build the documentation locally. From the docs root (e.g. `[your module]/docs/`), execute `make html`. The built site will be in `docs/_build/html/index.html`.

2.6.3 Guidelines

Please follow these guidelines when updating our docs. Let us know if you have any questions or something isn't clear.

Please place images in the same folder as the guide text file, following the convention `[file_name].[n].[optional description].[extension]`. For example, `configuring_page_display.3.rearrange.png` or `configuring_page_display.1.png` are both located in `docs/user_guide/` and are part of the `configuring_page_display.rst` guide.

We currently use the following syntax:

```
Title of File (using title case)
=====

Introduction text.

.. note::

    This would show up as a boxed note to the reader. It's good for ensuring important
    ↪points/hints are seen but should be used sparingly. It's also a good way to note
    ↪that "This guide is under development."
```

Section Title

```
-----

We use double backticks to indicate ``inline-code`` including file names, function
↪and method names, paths, etc.

Longer code-blocks should begin with the ``.. code-block:: [type]`` directive and
↪should be indented at least one
level. There should also be a blank line before and after it as shown below.

.. code-block:: bash

    if ($needs_documentation) {
        use $these_guidelines;
        $contribute_docs = $appreciated;
    }
```

Section 1.1 Title

```
^^^^^^^^^^^^^^^^^^^^

The use of appropriate sections makes reading documentation and later specific
↪details easier. Sub sections such
as this one will be hidden unless the main section is already selected.

The following toctree specifies that there are 3 files with additional content for
↪the current page.
The order is as it will appear in the sidebar and the link titles will be the "Title
↪of File" for each file.

.. toctree::
   :maxdepth: 2
   :caption: Contents:

   dir/file1
   dir/file2
   dir/file3
```

For more information on syntax, see [this RestructuredText Primer](#).

2.6.4 Starting Docs for a new Module

1. Install sphinx using pip and navigate yourself into git directory.
2. Create a docs folder there and go inside. Run `sphinx-quickstart` and set to your needs. Simply accept the defaults except for:

```
The project name will occur in several places in the built documentation.
> Project name: Genotypes Loader
> Author name(s): Carolyn T Caron et al., University of Saskatchewan, Pulse_
↪Bioinformatics
> Project release []:
```

3. Create one .rst document per planned section and populate it according to the guidelines above. If a given section is long enough to be broken up into sub-sections then create a folder with the same name as [your section].rst and place all .rst files for that section within it.
4. For each .rst file, ensure it is added to the `toctree` for that section. The base sections should be added to the `index.rst`
5. Edit the `index.rst`: change the title to `[My Module]: Available Documentation` and remove the `Indices and Tables` section.
6. Ensure your `conf.py` matches the one provided here. by comparing each section. Do not change your project information and pay special attention to `PHP Highlighting` and `Options for HTML output`.
7. Copy `_static` from this repo to your docs folder. This contains the `UofS-Pulse-Binfo` theme alterations.
8. Build your documentation using `make html` and check `_build/html/index.html` to ensure your documentation looks as it should.
9. Add `_build/` to your `.gitignore` file and commit your changes pushing them out to github.
10. Go to <https://readthedocs.org/dashboard/> and add this module as a project.

Note: If Theme error: no theme named 'sphinx_rtd_theme' found (missing theme.conf?) is encountered in step 8. This error indicates sphinx theme package `sphinx-rtd-theme` is not available in your system and it can be installed by executing following command on Linux or MacOS.

```
sudo pip install sphinx-rtd-theme
```
