
limelight

Michael Riffle

Mar 18, 2023

CONTENTS

1	Citing	3
2	Getting Help, Providing Feedback, or Reporting Problems	5
3	Introduction	7
4	Tutorials for Running Pipeline	9
4.1	Using Limelight	9
4.2	Tutorials	16
4.3	Administration	28

Limelight is a web application for analyzing, visualizing, and sharing and bottom-up proteomics results.

CITING

If you use Limelight in your work, please cite:

Discovery and Visualization of Uncharacterized Drug-Protein Adducts Using Mass Spectrometry. Riffle M, Hoopmann MR, Jaschob D, Zhong G, Moritz RL, MacCoss MJ, Davis TN, Isoherranen N, Zelter A. *Anal Chem.* 2022 Mar 1;94(8):3501-3509. doi: 10.1021/acs.analchem.1c04101

GETTING HELP, PROVIDING FEEDBACK, OR REPORTING PROBLEMS

If after reading this documentation you have questions about Limelight, ideas for new features, or want to report any problems, please contact us!

- Try using the [Limelight Issue Page](#) at GitHub.
- You can [join our Slack](#) and chat with us.
- You can email us at limelightms@uw.edu

INTRODUCTION

Limelight is designed to provide you with the full-stack of proteomics results, regardless of which processing pipeline you used to search your data. Full-stack means that you have access to the global views of your data (such as statistically comparing conditions), to viewing lists of proteins and peptides, to individual PSMs and spectra—all showing the native scores from whichever pipeline you used. Additionally, all native scores from your pipeline are available to you for filtering—even when contrasting multiple searches that each used different pipelines.

TUTORIALS FOR RUNNING PIPELINE

We have developed tutorials to accompany Riffle M, Hoopmann M, et al. (2022) that describe the process for running the Magnum, Percolator, Limelight pipeline; and how to then perform the statistical test for finding mods of interest in Limelight. Please see our [Tutorials Page](#)

4.1 Using Limelight

Here are guides and tutorials related to using Limelight.

4.1.1 How to Upload Data to Limelight

Follow these steps to upload your data to Limelight.

Convert to Limelight XML

Before search results can be uploaded to Limelight, they must be converted to Limelight XML. Please see our [How to Convert Your Data to Limelight XML](#) guide for more information.

Upload via Limelight Web Site

The simplest option for uploading data is to upload through the Limelight web interface. Just follow these steps:

1. Log into Limelight
2. Navigate to an existing project or create a new project.
3. Scroll down to the Upload Data section and click the arrow on the left to expand the section.

▼ Upload Data (Pending 0)

Import Limelight XML File

Refresh

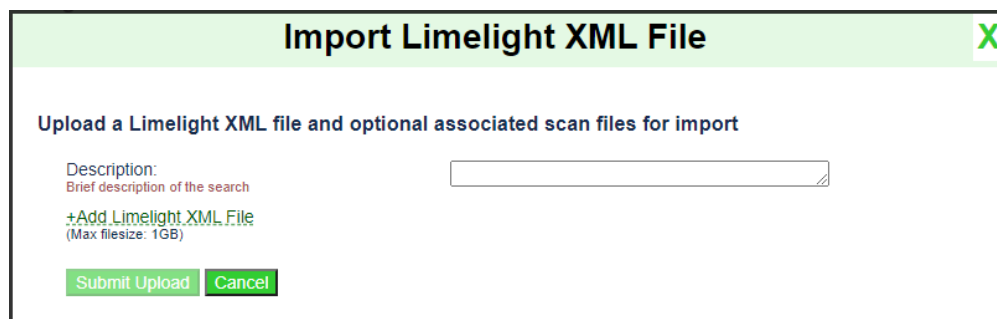
Command Line Import Info

Data must be converted to limelight XML before upload.

▼ Pending

No uploads pending

4. Click the Import Limelight XML File button. You should see the following dialog appear:



5. Enter a description of this run. E.g., Treated.
6. Click on the +Add Limelight XML File link and select your Limelight XML file on your computer.
7. (Optional) Click on the +Add Scan File link and select the corresponding mzXML (or mzXML) file on your computer. This is required if you would like to view spectra associated with peptide identifications.
8. Click the Submit Upload button to submit your data to Limelight.

After several minutes, refresh the page and your search should appear under the Explore Search Results section of the project page. Click on the Peptides, Proteins, or Modifications links to view the data.

Upload via Command Line

You may also upload to Limelight directly from the command line of your computer or server. This is useful for automating the upload of a large number of files, or if the server has a faster connection to Limelight. Follow these steps to upload from the command line.

Obtain the submission Java program

If you haven't already, download the submission program. This is the program you will run on the command line to send data to Limelight.

1. Log into Limelight
2. Navigate to an existing project or create a new project.
3. Scroll down to the Upload Data section and click the arrow on the left to expand the section.
4. Click the Command Line Import Info button.

▼ Upload Data (Pending 0)

[Import Limelight XML File](#)
[Refresh](#)
[Command Line Import Info](#)

Data must be converted to limelight XML before upload.

▼ Pending

No uploads pending



5. In the window that opens, click the `Submit Import Program` link to download the import program. Save it to a location on your computer that you will remember.

Command Line Import Information



Download the Submit Import Program

[Submit Import Program](#)



Documentation for using the Submit Import Program

[Documentation](#)

Parameters for use with the Submit Import Program

`--limelight-web-app-url=https://limelight.yeastrc.org/limelight`

`--project-id=68`

No value set for: `--user-submit-import-key`

[Make Key](#)

Alternatively, you can download it using `wget` in your terminal. Right-click on the `Submit Import Program`, copy the link address to your clipboard and type the following into your terminal:

```
wget <paste link address>

# for example:
wget https://limelight.yeastrc.org/limelight/static/limelightSubmitImport/
↪ limelightSubmitImport.jar
```

Run the submission Java program

Follow these directions to run the command line submission program.

1. Ensure Java is installed on the system where you will be running the submission program. If it is not installed, see the [official download site for Java](#).
2. Following the first 4 steps above to open the `Command Line Import Information` window. If you have not created a user submit import key, click the `Make Key` button as depicted below. This submit import key is unique to your account and should not be shared. Anyone with this key can upload data to your projects.

Command Line Import Information



Download the Submit Import Program

[Submit Import Program](#)

Documentation for using the Submit Import Program

[Documentation](#)

Parameters for use with the Submit Import Program

--limelight-web-app-url=https://limelight.yeastrc.org/limelight

--project-id=68

No value set for: --user-submit-import-key

[Make Key](#)



3. Reveal your submit import key by clicking on the Show Key button:

Command Line Import Information



Download the Submit Import Program

[Submit Import Program](#)

Documentation for using the Submit Import Program

[Documentation](#)

Parameters for use with the Submit Import Program

--limelight-web-app-url=https://limelight.yeastrc.org/limelight

--project-id=68

--user-submit-import-key=

[Show Key](#)



[Change Key](#)

[Remove Key](#)

Do not share user-submit-import-key, it is unique to your account.

4. Note the following highlighted information that appears:

Command Line Import Information



Download the Submit Import Program

[Submit Import Program](#)

Documentation for using the Submit Import Program

[Documentation](#)

Parameters for use with the Submit Import Program

```
--limelight-web-app-url=https://limelight.yeastrc.org/limelight
--project-id=68
--user-submit-import-key=c4fed1f053e11b331b9816a194a128813fb18cbadf
```

[Change Key](#)

[Remove Key](#)

Do not share user-submit-import-key, it is unique to your account.

This information is needed to upload data to this project. The first line is the URL of this Limelight installation URL, the second line is the project id number for this project, and the third line is an unguessable key that allows data to be uploaded as you. You can save this information for future use, without logging into Limelight.

Be ready to copy and paste this information into the command for running the submission program (below).

5. Run the command line submission program. The command to run the importer is the following. You will need to replace the locations of the limelightSubmitImport.jar, your Limelight XML file, and your mzML file(s) to your actual locations. You will also need to replace lines 2, 3, and 4 with what you copied from step 2 above.

```
# the command line program if you DO have scan file(s)
java -jar /location/to/limelightSubmitImport.jar \
  --limelight-web-app-url=https://limelight.yeastrc.org/limelight \
  --project-id=7 \
  --user-submit-import-key=12dee46123ceb09c61af8d34a9151c512657cf7f39 \
  --limelight-xml-file=/location/to/limelight.xml \
  --scan-file=/location/to/your.mzML

# --scan-file can be repeated multiple times if you have multiple mzML or
↪mzXML files in the search results.
```

If you do not have scan files to upload, the command would be the following:

```
# the command line program if you DO NOT have scan file(s)
java -jar /location/to/limelightSubmitImport.jar \
  --limelight-web-app-url=https://limelight.yeastrc.org/limelight \
  --project-id=7 \
  --user-submit-import-key=12dee46123ceb09c61af8d34a9151c512657cf7f39 \
  --limelight-xml-file=/location/to/limelight.xml \
  --no-scan-files
```

Note: If you are using a Windows command line, the lines should end with ^ instead of \. For example:

```
java -jar c:\path\to\limelightSubmitImport.jar ^
  --limelight-web-app-url=https://limelight.yeastrc.org/limelight ^
  --project-id=7 ^
  --user-submit-import-key=12dee46123ceb09c61af8d34a9151c512657cf7f39 ^
  --limelight-xml-file=c:\path\to\limelight.xml ^
  --no-scan-files
```

4.1.2 How to Convert Your Data to Limelight XML

Limelight is designed to support data generated by any mass spec data processing pipeline. To accomplish this, we have designed a file format called Limelight XML that implements an XML schema that stores the data that Limelight needs in the most generalized way possible. Data that are uploaded to Limelight need to be formatted as Limelight XML. And, importantly, any data represented as Limelight XML can be uploaded to Limelight—regardless of the software pipeline that generated the data.

What are Converters

Converters are programs that convert the native output of software pipelines to Limelight XML that can be imported into Limelight.

This design moves the job of reading the output of different software pipelines to smaller individual programs we call converters. Each of these programs can be maintained individually, separate from Limelight—ensuring they stay current. And adding support for a new search pipeline becomes the relatively simple matter of writing a converter for the output of that program.

How to Get Converters

We have written a number of converters for popular mass spec processing pipelines. Follow the links below to the GitHub repository for the individual converter for information on how to run that converter.

Table 1: Table of Current Converters

Comet + Percolator	https://github.com/yeastrc/limelight-import-comet-percolator
Comet + Trans Proteomic Pipeline	https://github.com/yeastrc/limelight-import-comet-tpp
Comet-PTM	https://github.com/yeastrc/limelight-import-cometptm
Crux	https://github.com/yeastrc/limelight-import-crux-comet-percolator
Magnum + Percolator	https://github.com/yeastrc/limelight-import-magnum-percolator
Magnum + Trans Proteomic Pipeline	https://github.com/yeastrc/limelight-import-magnum-tpp
MetaMorpheus	https://github.com/yeastrc/limelight-import-metamorpheus
modA	https://github.com/yeastrc/limelight-import-moda
MSFragger	https://github.com/yeastrc/limelight-import-msfragger-tsv
MSFragger + Trans Proteomic Pipeline	https://github.com/yeastrc/limelight-import-msfragger-tpp
Open pFind	https://github.com/yeastrc/limelight-import-open-pfind
Philosopher (MSFragger or Comet)	https://github.com/yeastrc/limelight-import-philosopher-tsv
ProLuCID (processed with DTASelect, using IP2)	https://github.com/yeastrc/limelight-import-prolucid-dtaselect
Tag Graph	https://github.com/yeastrc/limelight-import-taggraph

How to Run Converters

Each of the above converters has a link to the specific converter's GitHub repository with instructions for running that specific converter. Each converter has slightly different options related to the specific pipeline being converted.

In general, the process of converting your data to Limelight XML would work like the following:

- Run your pipeline and generate your results.
- If not already installed, download the converter program from one of the above repositories.
- Follow the directions for running the converter. Generally each converter will require that you specify the following by passing it command line parameters when you run the program:
 - The data file(s) generated by the search pipeline
 - The configuration file used to run the search
 - The FASTA file containing the sequences searched
- Now the resulting Limelight XML may be uploaded to Limelight.

Getting Help or Requesting a New Converter

If you have any questions, bug reports, or requests for new converters, please contact us!

- Try using the [Limelight Issue Page](#) at GitHub.
- You can [join our Slack](#) and chat with us.
- You can email us at limelightms@uw.edu

4.2 Tutorials

Here are some tutorials related to using limelight:

4.2.1 Installing Docker



Many of these tutorials will assume Docker is installed on your system. Docker allows you to run software on your computer in containers that include the tested environment for running that software—regardless of your own operating system or other installed programs. If Docker is not already installed on your system, see below for instructions to install Docker on most major operating systems.

For more information about Docker, see [the official Docker website](#)

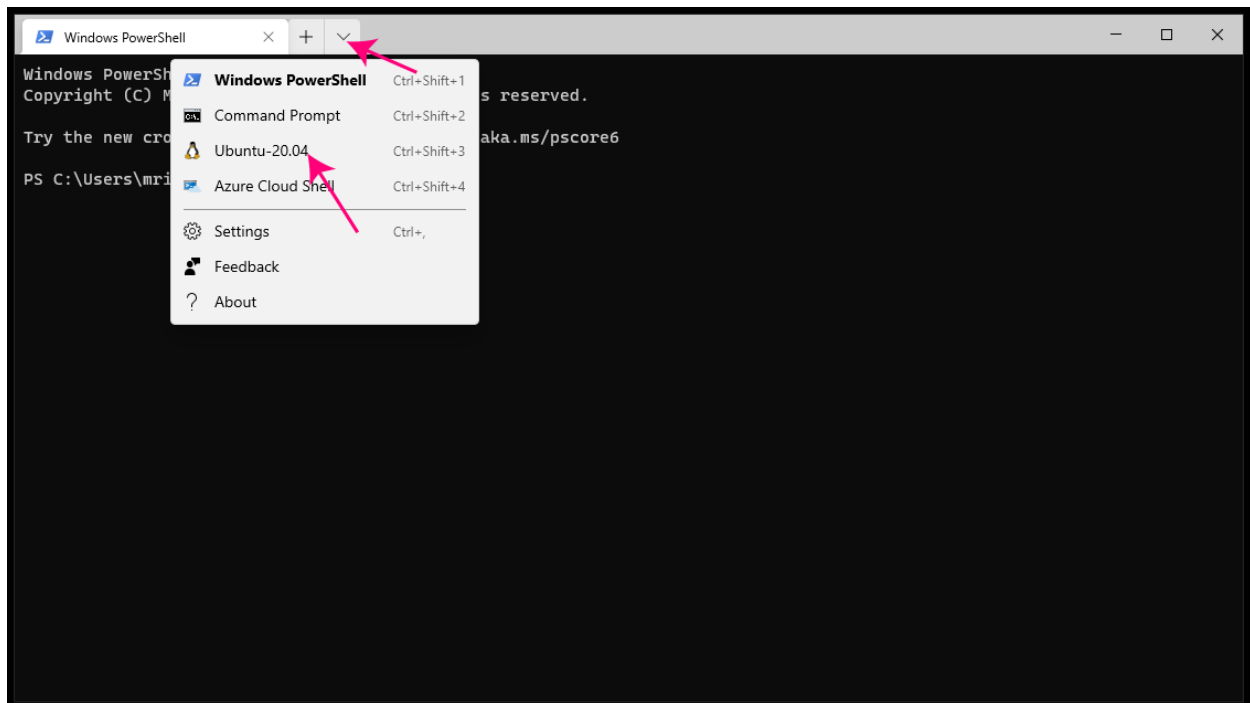
Microsoft Windows

This tutorial will install WSL2 (Microsoft’s Windows Subsystem for Linux) and Docker Desktop, which will allow you to run any Docker containers natively in Linux on Windows.

1. Install WSL2 by [following these directions from Microsoft](#). If WSL2 is already installed, skip this step.
2. Install Docker Desktop and link it to WSL2 by [following these directions from Microsoft](#). If Docker Desktop is already installed, follow the link and follow the directions to ensure it is enabled in your Linux distribution.

Windows Terminal

If you are using Windows 10, we recommend installing **Windows Terminal** at this time. Windows 11 has Windows Terminal installed by default. [Follow these directions from Microsoft to install Windows Terminal](#). Once installed you can easily access the command line of your new Ubuntu LTS installation by launching Windows Terminal and clicking the menu icon as depicted below. Note *Ubuntu 20.04* may be a different version number on your system.



If you do not install Windows Terminal, you can access your new Ubuntu LTS installation by opening a command prompt (Press the Win + R keys on your keyboard, then type cmd, and press Enter on your keyboard or click/tap OK). Enter bash into the command prompt and hit enter.

Apple macOS

Follow the official directions to install Docker on macOS. Once installed and running, Docker may be accessed by opening a new Terminal and typing docker.

Linux

Below are instructions and links to instructions for installing Docker on the most popular Linux distributions.

Ubuntu 20.04

Copy and paste each of the following commands, one by one, into the Linux terminal:

```
sudo apt update -y
sudo apt install apt-transport-https ca-certificates curl software-properties-
common -y
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/
ubuntu focal stable"
sudo apt update -y
apt-cache policy docker-ce
sudo apt install docker-ce -y
```

CentOS

Follow the official Docker instructions for CentOS.

Debian

Follow the official Docker instructions for Debian.

Fedora

Follow the official Docker instructions for Fedora.

Other Linux Distributions

Please see the official Docker install guides for more information about installing Docker on other Linux distributions.

4.2.2 Magnum Protein Adduct Pipeline

This tutorial covers how to run the pipeline for protein adduct discovery and visualization using **Magnum**, **Percolator**, and uploading to **Limelight** as discussed in:

Discovery and Visualization of Uncharacterized Drug-Protein Adducts Using Mass Spectrometry. Riffle M, Hoopmann MR, Jaschob D, Zhong G, Moritz RL, MacCoss MJ, Davis TN, Isoherranen N, Zelter A. Anal Chem. 2022 Mar 1;94(8):3501-3509. doi: 10.1021/acs.analchem.1c04101

This tutorial assumes you have Docker installed on your system. Please see our [Docker Installation Tutorial](#) to get Docker installed. Because Docker is being used, you **do not need to install Magnum, Percolator, or the Limelight XML converter**.

We have also made example data input files available, and the steps below include downloading those.

Note: Although this tutorial makes use of Docker, Docker is not required to run these programs. Please see the web sites for the individual programs for more details on other ways to run the software.

Set Up Your Project Folder

1. Open up a terminal. If you are on Windows, follow the directions on our [Docker Installation Tutorial](#) to open a Linux terminal.
2. Create a project folder

```
mkdir ~/my-project  
cd ~/my-project
```

3. Copy the example input data into the project folder

```
# download data from Google Drive into your project directory
wget --load-cookies /tmp/cookies.txt "https://docs.google.com/uc?export=download&
↪confirm=$(wget --quiet --save-cookies /tmp/cookies.txt --keep-session-cookies --
↪no-check-certificate 'https://docs.google.com/uc?export=download&
↪id=13IeYZu2Jb71VjBlw2BxFM0-3yFJV33Jl' -O- | sed -rn 's/.*confirm=([0-9A-Za-z_]+).
↪*/\1\n/p')&id=13IeYZu2Jb71VjBlw2BxFM0-3yFJV33Jl" -O treated.tgz && rm -rf /tmp/
↪cookies.txt
tar -xvzf treated.tgz
```

Note: If you are using macOS, you may not have `wget` installed on your system so the above step may fail. You can download `treated.tgz` using a web browser using [this link](#), and then save it to your project directory and run `tar -xvzf treated.tgz`. You can alternatively install `wget` using Homebrew by typing the following:

```
# install Brew (if it is not installed)
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/
↪master/install)"

# install wget
brew install wget
```

If the above was successful, you can delete `treated.tgz`

```
rm treated.tgz
```

You should now have three files in your project directory:

1. `SearchDatabase.fasta` - A FASTA-formatted text file containing protein sequences to be searched.
2. `treated-Magnum.conf` - A Magnum configuration file, pre-configured for this tutorial. For more information about configuring Magnum, [visit the Magnum configuration website](#).
3. `treated.mzML` - The raw data from the mass spectrometer converted to mzML format.

Run Magnum

1. Ensure you are in your project directory (where you copied the files above).

```
cd ~/my-project
```

2. Run the Magnum search.

```
sudo docker run --rm -it --user $(id -u):$(id -g) -v `pwd`:`pwd` -w `pwd` mriffle/
↪magnum:alpha5 ./treated-Magnum.conf
```

You should see Magnum output its progress as it searches the file.

Visit the [official Magnum website](#) for more information about Magnum.

Run Percolator

In the above example, Magnum would have generated a file named `treated.perc.txt`. This is the input file for the Percolator post processing software. To run percolator enter:

```
sudo docker run --rm -it --user $(id -u):$(id -g) -v `pwd`:`pwd` -w `pwd` \
↳mriffle/percolator:3.05 percolator -X percout.xml treated.perc.txt
```

This will generate a file named `percout.xml` that contains the percolator results in XML format.

Visit the [official Percolator website](#) for more information about Percolator.

Convert results to Limelight XML

To import data into Limelight, it must be converted to a Limelight XML file. After running the above steps you should have the following files (plus others) in your project directory:

1. `treated-Magnum.conf` - configuration file used by Magnum
2. `treated.pep.xml` - Magnum search results in pepxml format. This will have the prefix of your mzml file name and the suffix of `pep.xml`.
3. `percout.xml` - The results from running percolator.
4. `SearchDatabase.fasta` - The FASTA file you used to perform the Magnum search.

To generate the Limelight XML file, enter the following:

```
sudo docker run --rm -it --user $(id -u):$(id -g) -v `pwd`:`pwd` -w `pwd` \
↳mriffle/magnum-percolator-to-limelight:v4.1.1 -c ./treated-Magnum.conf -p ./
↳percout.xml -f ./SearchDatabase.fasta -m ./treated.pep.xml -o treated.
↳limelight.xml
```

Here is that same command in multi-line format with comments. Note the command below won't run as-is, it is only meant to show you which parameters are being used in the event that you need to change the values.

```
sudo docker run --rm -it --user $(id -u):$(id -g) -v `pwd`:`pwd` -w `pwd` \
↳mriffle/magnum-percolator-to-limelight\
  -c ./treated-Magnum.conf      # the Magnum configuration file
  -p ./percout.xml             # the output from percolator
  -f ./SearchDatabase.fasta    # the FASTA file used in the search
  -m ./treated.pep.xml         # the Magnum results
  -o treated.limelight.xml     # the limelight XML file that will be created
```

You should now have a `treated.limelight.xml` file that will be used to import the results into Limelight.

Visit the [converter GitHub repository](#) for more details about this converter.

Upload to Limelight

Important: This section assumes you have access to a running Limelight installation. You may use the installation at <https://use.limelight-ms.org/> or you may run your own. To run your own follow our [Limelight Installation Tutorial](#).

To view the results in Limelight, use the Limelight web interface to upload the Limelight XML and (optionally) the mzml file.

Steps to upload your data to Limelight:

1. Log into Limelight
2. Navigate to an existing project or create a new project.
3. Scroll down to the Upload Data section and click the arrow on the left to expand the section.

▼ Upload Data (Pending 0)

Import Limelight XML File

Refresh

Command Line Import Info

Data must be converted to limelight XML before upload.

▼ Pending

No uploads pending

4. Click the Import Limelight XML File button. You should see the following dialog appear:

5. Enter a description of this run. E.g., Treated.
6. Click on the +Add Limelight XML File link and select your Limelight XML file on your computer.

Note: Windows Users: Access to files on WSL 2 (Linux) is slightly more complicated than accessing normal files. You have two options:

1. You can copy your Limelight XML file (and, optionally, mzML file) to a Windows directory, such as C:\data_directory. To copy treated.limelight.xml to C:\data_directory\ you would enter the following into your terminal.

```
mkdir /mnt/c/data_directory
cp ~/my-project/treated.limelight.xml /mnt/c/data_directory/treated.
```

(continues on next page)

(continued from previous page)

```
↪ limelight.xml

# optional: if you wish to upload and view spectra
cp ~/my-project/treated.mzML /mnt/c/data_directory/treated.mzML
```

Note: /mnt/c/data_directory corresponds to C:\data_directory. To select your data to upload below, you would navigate to C:\data_directory.

2. You can access the files on your WSL 2 (Linux) installation directly. When you click +Add Limelight XML File a file browser will appear. At the top, click in the address bar. Erase what is there and put in: \\wsl\$\Ubuntu-20.04\home and hit the Enter key. You will see your Linux home directory listed. Double click your home directory, then double click *my-project*. You should see your files from this tutorial listed here.

7. (Optional) Click on the +Add Scan File link and select your mzml file on your computer. This is required if you would like to view spectra associated with peptide identifications.

8. Click the Submit Upload button to submit your data to Limelight.

After several minutes, refresh the page and your search should appear under the Explore Data section of the project page. Click on the Peptides, Proteins, or Modifications links to view the data.

Optional: Analyze and upload the untreated sample

The above steps search and upload the results for the treated sample. If you would also like to search and upload the untreated sample to compare in Limelight follow these directions.

Get the untreated data

1. Ensure you are in the project folder

```
cd ~/my-project
```

2. Copy the example input data into the project folder

```
# download data from Google Drive into your project directory
wget --load-cookies /tmp/cookies.txt "https://docs.google.com/uc?export=download&
↪confirm=$(wget --quiet --save-cookies /tmp/cookies.txt --keep-session-cookies --
↪no-check-certificate 'https://docs.google.com/uc?export=download&
↪id=1AzGMBh9kCByX2K5esBS3RDF-7ZTxSiK' -O- | sed -rn 's/.*confirm=([0-9A-Za-z_]+).
↪*/\1\n/p')&id=1AzGMBh9kCByX2K5esBS3RDF-7ZTxSiK" -O untreated.tgz && rm -rf /tmp/
↪cookies.txt
tar -xvzf untreated.tgz
```

Note: If you are using macOS, you may not have wget installed on your system so the above step may fail. You can download treated.tgz using a web browser using [this link](#), and then save it to your project directory and run tar -xvzf untreated.tgz.

If the above was successful, you can delete untreated.tgz

```
rm untreated.tgz
```

You should now have three files in your project directory:

1. `SearchDatabase.fasta` - A FASTA-formatted text file containing protein sequences to be searched.
2. `untreated-Magnum.conf` - A Magnum configuration file, pre-configured for this tutorial. For more information about configuring Magnum, visit the [Magnum configuration website](#).
3. `untreated.mzML` - The raw data from the mass spectrometer converted to mzML format.

Run Magnum on Untreated Data

```
sudo docker run --rm -it --user $(id -u):$(id -g) -v `pwd`:`pwd` -w `pwd` \
↳mriffle/magnum:alpha5 ./untreated-Magnum.conf
```

Run Percolator on Untreated Data

```
sudo docker run --rm -it --user $(id -u):$(id -g) -v `pwd`:`pwd` -w `pwd` \
↳mriffle/percolator:3.05 percolator -X percout.xml untreated.perc.txt
```

Convert Untreated Results to Limelight XML

```
sudo docker run --rm -it --user $(id -u):$(id -g) -v `pwd`:`pwd` -w `pwd` \
↳mriffle/magnum-percolator-to-limelight:v4.1.1 -c ./untreated-Magnum.conf -p .
↳/percout.xml -f ./SearchDatabase.fasta -m ./untreated.pep.xml -o untreated.
↳limelight.xml
```

You should now have a `untreated.limelight.xml` file that will be used to import the results into Limelight.

Upload Untreated Data to Limelight

Follow the instructions above to [Upload to Limelight](#).

4.2.3 Find Statistically Significant Modifications with Limelight

This tutorial covers how to use Limelight to compare the results of two searches (treated and untreated) to find the modifications with the most statistically significant differences (in terms of their spectral counts) between the two conditions. This is the method employed in the following publication:

Discovery and Visualization of Uncharacterized Drug-Protein Adducts Using Mass Spectrometry. Riffle M, Hoopmann MR, Jaschob D, Zhong G, Moritz RL, MacCoss MJ, Davis TN, Isoherranen N, Zelter A. *Anal Chem*. 2022 Mar 1;94(8):3501-3509. doi: 10.1021/acs.analchem.1c04101

This tutorial will assume the user has completed our [Magnum Pipeline Tutorial](#) to generate a `treated.limelight.xml` file, and has completed the optional step to produce the `untreated.limelight.xml` file.

Upload Data to Limelight

Although uploading data is covered in our *Magnum Pipeline Tutorial*, we will repeat those steps here. If you have already uploaded your `treated.limelight.xml` and `untreated.limelight.xml` files to Limelight, skip this step.

Note: If you are using Windows, this step is greatly simplified by first copying your Limelight XML files to a Windows filesystem drive, such as C:\. To copy `treated.limelight.xml` and `untreated.limelight.xml` to C:\data_directory\ you would enter the following into your terminal.

```
cp ~/my-project/treated.limelight.xml /mnt/c/data_directory/treated.limelight.
↪xml
cp ~/my-project/untreated.limelight.xml /mnt/c/data_directory/untreated.
↪limelight.xml

# Optional: You must upload your mzML files to view spectra. Copy to C:\ for
↪upload later
cp ~/my-project/treated.mzML /mnt/c/data_directory/treated.mzML
cp ~/my-project/untreated.mzML /mnt/c/data_directory/untreated.mzML
```

/mnt/c/ corresponds to your C:\ drive. /mnt/d/ corresponds to your D:\ drive, and so on.

Steps to upload your data to Limelight:

1. Log into Limelight
2. Navigate to an existing project or create a new project.
3. Scroll down to the Upload Data section and click the arrow on the left to expand the section.

▼ Upload Data (Pending 0)

Import Limelight XML File

Refresh

Command Line Import Info

Data must be converted to limelight XML before upload.

▼ Pending

No uploads pending

4. Click the Import Limelight XML File button. You should see the following dialog appear:

Import Limelight XML File

Upload a Limelight XML file and optional associated scan files for import

Description:

Brief description of the search

+Add Limelight XML File

(Max filesize: 1GB)

Submit Upload

Cancel

5. Enter a description of this run. E.g., Treated.

- Click on the **+Add Limelight XML File** link and select your `treated.limelight.xml` on your computer.
- (Optional) Click on the **+Add Scan File** link and select your `treated.mzML` file on your computer. This is required if you would like to view spectra associated with peptide identifications.
- Click the **Submit Upload** button to submit your data to Limelight.
- Repeat steps 4-6 for your `untreated.limelight.xml` and, optionally, your `untreated.mzML` file.

After several minutes, refresh the page and your searches should appear under the **Explore Data** section of the project page.

Navigate to the Mod View Page

Once your data are imported, the searches will appear under the **Explore Data** section:

▼ Explore Data

Expand All Collapse All Copy Searches Move Searches Organize Searches Filter Overrides

☐ Untreated (2346) [Peptides] [Proteins] [Modifications]

☐ Treated (2345) [Peptides] [Proteins] [Modifications]

Compare Peptide View Compare Protein View Compare Mod View

You will see your two searches listed here. To find the modifications with significantly different spectral counts between the searches, click the check box next to each search and click the **Compare Mod View** button:

▼ Explore Data

Expand All Collapse All Copy Searches Move Searches Organize Searches Filter Overrides

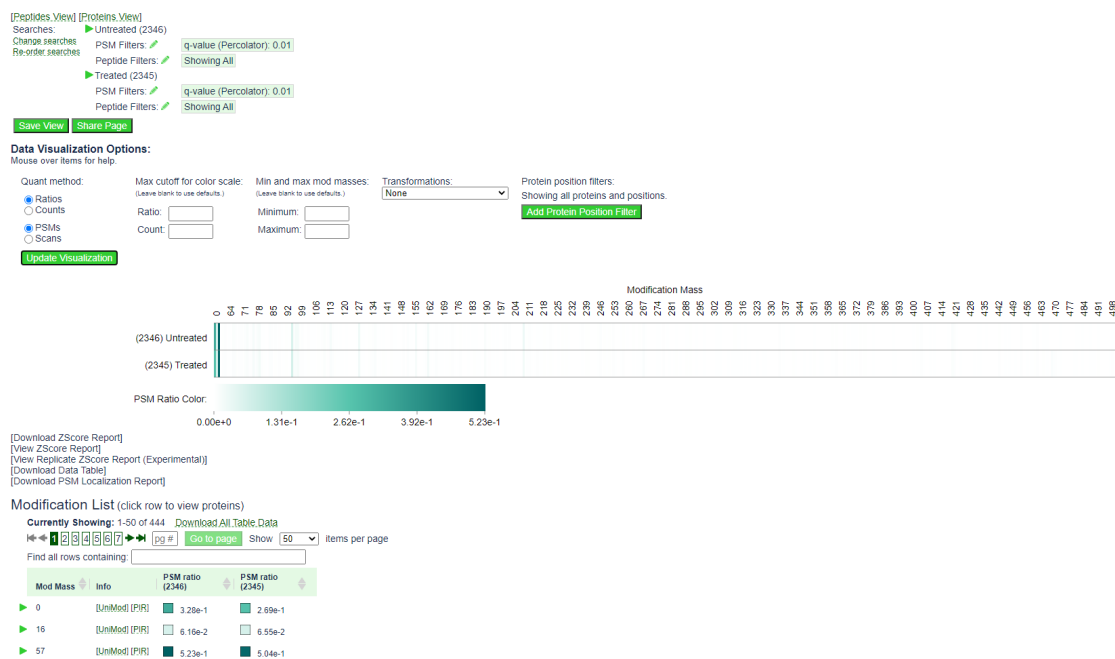
☒ Untreated (2346) [Peptides] [Proteins] [Modifications]

☒ Treated (2345) [Peptides] [Proteins] [Modifications]

Compare Peptide View Compare Protein View **Compare Mod View**

Change Visualization Options

By default the mod page will show the following:



The heat map shows the spectral count for all PSMs with a given mod mass in each of the two searches. In this example, the spectral counts for 0, 16, and 57 are drowning out the signal for the open modification masses, which will have a modification mass of 60 or higher.

To restrict the mod masses to those with a mass of 60 or higher, change the minimum mod mass to 60 and click the Update Visualization button, as depicted below:

Data Visualization Options:
Mouse over items for help.

Quant method:
☒ Ratios
☐ Counts
☒ PSMs
☐ Scans

Max cutoff for color scale:
(Leave blank to use defaults.)
 Ratio:
 Count:

Min and max mod masses:
(Leave blank to use defaults.)
 Minimum:
 Maximum:

Transformations:

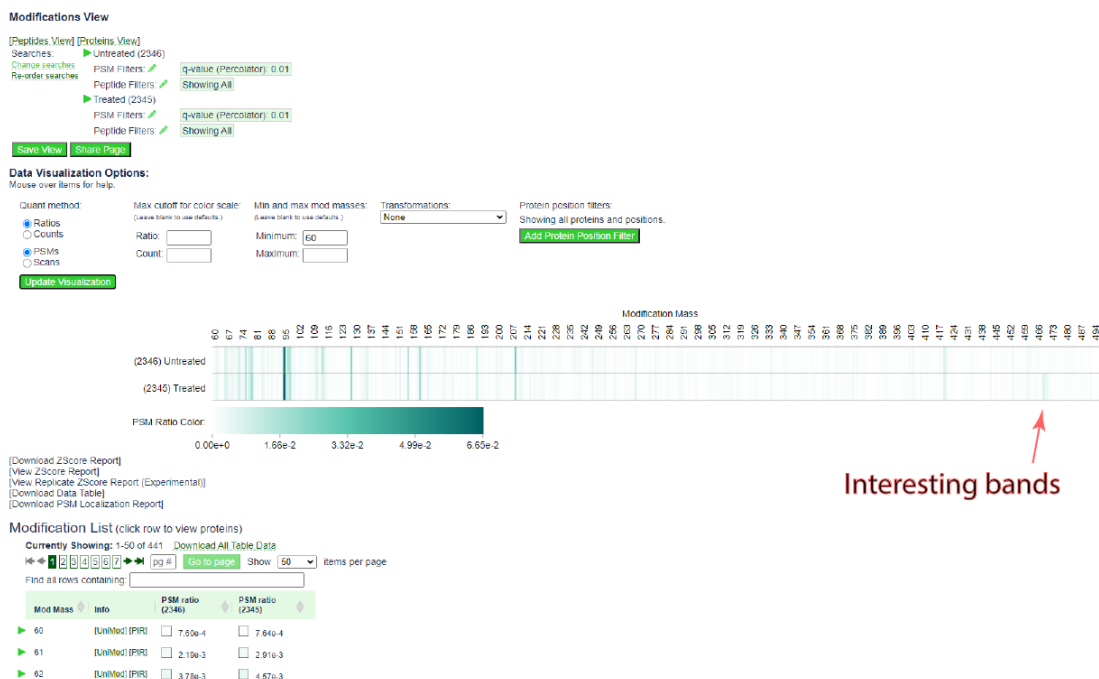
Protein position filters:
 Showing all proteins and positions.
[Add Protein Position Filter](#)

[Update Visualization](#)

1

2

The page should update to the following:



The heat map is more informative. Note the bands that appear near 470 in the treated sample, but not the untreated sample.

Run the Report

To run a statistical analysis comparing the spectral counts for mod masses in the two searches, click the View ZScore Report link below the data visualization:

0.00e+0

[Download ZScore Report]

[View ZScore Report] ←

[View Replicate ZScore Report (Experimental)]

[Download Data Table]

[Download PSM Localization Report]

Modification List (click row to view proteins)

Currently Showing: 1-50 of 441 [Download All Table Data](#)

1 2 3 4 5 6 7 page # 100 page

This will compare the ratio of PSMs that have a given mod mass to all PSMs in each search using a test for proportions and produce the following report:

Significant Mods Table							
search1	search2	mod mass	PSMcount 1	PSMcount 2	z-score	p-value	filtered z-score
2346	2345	469	13	396	-19.272017209888148	0	-18.4477810
2346	2345	470	32	237	-12.755931209018906	0	-12.0815401
2346	2345	471	31	186	-10.749269018885856	0	-10.1421113
2346	2345	114	752	428	8.979623213268997	0	10.4451033
2346	2345	209	1463	1003	8.636585917271546	0	10.7846033
2346	2345	216	45	155	-7.993086139719835	5.875300246316328e-13	-7.40840316
2346	2345	97	854	549	7.645457516420007	9.204637052562248e-12	9.24362316
2346	2345	79	686	962	-7.44952370875627	4.1322945065758176e-11	-5.76634339
2346	2345	80	59	157	-6.8904525361177615	2.4526440878247513e-9	-6.28206303
2346	2345	347	4	44	-5.876695308604806	0.0000018457505424063214	-5.59007140
2346	2345	490	7	50	-5.807985678079681	0.000002788395513486641	-5.49551831
2346	2345	238	161	73	5.535682916598987	0.000013671793464342663	6.17785457
2346	2345	95	3229	3568	-5.491659356271488	0.000017559509601361434	-1.97449971
2346	2345	147	12	56	-5.458606209146715	0.00002116300652221792	-5.11708301
2346	2345	268	259	387	-5.42429523686115	0.00002565898749973261	-4.36933786
2346	2345	356	26	80	-5.398639486028379	0.0000296123347063304	-4.97197856
2346	2345	91	30	85	-5.28888439301042	0.00005427150552061999	-4.84439065
2346	2345	327	22	70	-5.147271725870931	0.00011655733052629014	-4.74973415
2346	2345	141	14	56	-5.144457957003533	0.00011831779829485356	-4.79781011
2346	2345	431	9	46	-5.099314180371545	0.00015033087179794347	-4.79213750

This report is ordered by the magnitude of the Z-score. Note that the modification masses 469, 470, and 471 have the most significant Z-scores. A negative Z-score in this case denotes enrichment in the treated sample.

4.3 Administration

Here are guides and tutorials related to running your own Limelight installation.

4.3.1 Installing Limelight

Follow these steps to set up your own installation of Limelight on your own computer. These instructions include running Limelight with default settings and require only minimal configuration by the user. If you already have access to Limelight, you do not need to do this to use Limelight.

This tutorial assumes you have Docker installed on your system. Please see our [Docker Installation Tutorial](#) to get Docker installed.

Important: System Requirements: Limelight will consume a large amount of RAM, particularly when uploading data. You should have at least 6 gigabytes of RAM available on your system.

1. Open a Terminal

On Linux and MacOS, open a normal terminal. On Windows, if you followed our instructions for installing Docker, follow the directions on our [Docker Installation Tutorial](#) to open a Linux terminal.

2. Install Docker Compose

Docker Compose is an official add-on to Docker that greatly simplifies running applications that have multiple parts. Limelight has several parts, including a database, multiple web applications, and running programs. Docker Compose allows you to run a single command to launch and correctly stitch all of those components together into a working system. This all happens inside of Docker and does not install the software elsewhere on your computer.

If you are on MacOS, you will likely already have Docker Compose installed. If you are on Linux (including Windows users who installed Docker according to our instructions), test if Docker Compose is installed by typing `docker-compose`. If the command is not found, please install Docker Compose by typing the following:

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.1/
↪docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

If you run into any trouble installing Docker Compose, please see [Docker's official Docker Compose installation instructions](#).

3. Download Limelight Install Files

First set up a Limelight project directory:

```
mkdir ~/limelight
cd ~/limelight
```

Now, download the Limelight Docker Compose files:


```
# Download Limelight Docker Compose files
curl -L "https://github.com/yeastrc/limelight-core/releases/latest/download/
↪docker-compose-files.tgz" -o docker-compose-files.tgz

# Expand the archive
tar -xvzf docker-compose-files.tgz
```

Note: If you prefer to download a ZIP file or if you prefer to download the file another way, the latest release can be found on GitHub at <https://github.com/yeastrc/limelight-core/releases/latest>

4. Configure Limelight

Copy the sample configuration file into place:

```
cp docker/env-sample ./env
```

The `.env` file holds all of the necessary configuration for Limelight. It is recommended (but not required) that you change the first two lines of the file, which contain passwords to be used for the MySQL database.

The `.env` file should look something like this:

```
# .env file for supplying settings to initializing Limelight using docker-
↪compose

# Change these passwords.
MYSQL_ROOT_PASSWORD=change_this_password
MYSQL_PASSWORD=change_this_password

# Can change the mysql user Limelight uses, but not necessary
MYSQL_USER=limelight_db_user

# This manages the memory usage of components of Limelight
IMPORTER_JAVA_OPTIONS=-Xmx3g -Xms500m
WEBAPP_JAVA_OPTIONS=-Xms2024m -Xmx2024m

# This manages optimization settings for MySQL
MYSQL_OPTIONS=--max-connections=500
```

These can be changed using your favorite text editor. On Linux (including Docker on Windows), we'll assume that is `nano`. To edit the file, type:

```
nano .env
```

Change the passwords and type `Control-o`, `<ENTER>`, and `Control-x` to save and exit.

Important: By default, Docker manages where data are stored on your disk. If you would like to customize where Limelight stores data, please follow our [Customize Data Locations](#) tutorial. This should be done before continuing on to Step 6 below. Once that is complete, proceed to Step 6.

5. Starting and Stopping Limelight

At this point, starting and stopping Limelight should be straight forward.

To start Limelight:

```
sudo docker-compose up --detach
```

To stop Limelight:

```
sudo docker-compose down
```

Note: The first time you start Limelight, all of the components will download and the database will initialize. This may take a few minutes, depending on your download speed. Subsequent startups of Limelight will not require these steps and will be faster.

Note: These commands must be typed while you are in the project code directory. If you followed these instructions, you can ensure you are in this directory by typing:

```
cd ~/limelight
```

6. Connect to Your Limelight Installation

Point your web browser to to access Limelight running on your own computer!

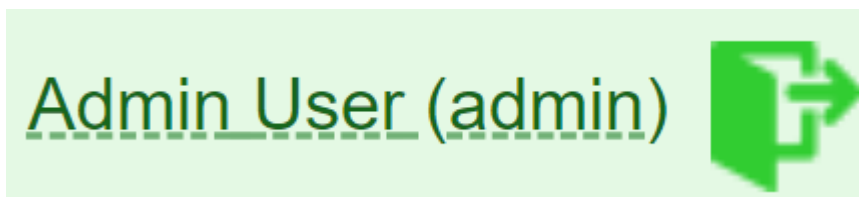
Note: If this is the first time bringing up Limelight, it may take a minute for the database to initialize. If you see message saying there was a problem with your request, try again in about a minute.

Login with Default User

By default, you can log in using `admin` as the username and `changeme` as the password.

Change Default User Information

To change the default log in information click on `Admin User (admin)` in the top right of the page:



Change the name, username, and password in the form to your liking.

Start Using Limelight

That's it, you are ready to use Limelight!

7. Optional - Set up SMTP For Emails

Some functions of Limelight require sending email to users. Examples of this include inviting new users to projects, resetting forgotten passwords, and notifications that data uploads have been completed. Although it's not required that you set up SMTP, the above features will not be enabled unless you do. If you would like to enable these features, please see our [Limelight SMTP Setup Guide](#).

If you do not set up SMTP, you must use the administrative interface to add new users to Limelight. See our [guide for managing users](#).

4.3.2 Update Limelight

New versions of Limelight with new features and bug fixes are regularly released. It's important to keep your Limelight installation up-to-date. Follow these instructions to update your installation of Limelight.

Limelight releases use semantic versioning and are given version numbers such as 2.1.6. The 2 is considered the major version number. Limelight versioning is carefully controlled, so that all updates to a new release that use the same major version (e.g., 2.1.6 to 2.3.12) do not require any database updates. This kind of update is considered a **minor version update**. Updates that move to a new major version (e.g. 2.1.6 to 3.5.2) may require database updates. This kind of update is considered a **major version update**.

Minor version updates are relatively simple to perform. Major version updates are more complex—requiring extra steps such as backing up and updating the database.

Determine Latest Limelight Version

Visit the [official Limelight Release Page](#) to view the latest releases of Limelight. The most current release is listed at the top of the list of releases.

Determine Your Limelight Version

Follow these directions to determine your major version number.

Important: This guide assumes you have completed all the steps in the [Installing Limelight](#) tutorial.

1. Open a Terminal

On Linux and MacOS, open a normal terminal. On Windows, if you followed our instructions for installing Docker, follow the directions on our [Docker Installation Tutorial](#) to open a Linux terminal.

2. Get Version Number From `docker-compose.yml`

Type the following into your terminal. The first line takes you into the directory where you downloaded Limelight. The second command retrieves the version numbers.

```
cd ~/limelight
grep 'mriffle/limelight-webapp' docker-compose.yml
```

The output should be similar to:

```
image: mriffle/limelight-webapp:2
```

The **2** at the end of that line is your **major version number**.

Perform Appropriate Update

If the most recent release of Limelight has the same major version number as your local installation, perform a minor version update.

If the most recent release of Limelight has a newer major version number than your local installation, you may perform a minor or a major version update. A minor version update will ensure you are using the latest release of Limelight that has the same major version number as your installation (e.g., the latest **2.x.x** version). A major version update will be more complicated to perform, but will ensure you have access to all the latest features and bug fixes in Limelight.

Follow the instructions below to perform a minor or major version update.

Minor Version Update Instructions

Follow these steps to update Limelight to the latest release of your current major version. E.g., update your **2.x.x** installation of Limelight to the latest release of **2.x.x**.

This guide assumes you have completed all the steps in the *Installing Limelight* tutorial.

1. Open a Terminal

On Linux and MacOS, open a normal terminal. On Windows, if you followed our instructions for installing Docker, follow the directions on our *Docker Installation Tutorial* to open a Linux terminal.

2. Shutdown Limelight

Go into your Limelight project directory:

```
cd ~/limelight
```

If you have not customized the data locations for Limelight, type the following to start Limelight:

```
sudo docker-compose down
```

If you have customized the data locations for Limelight by following our *Customize Data Locations* tutorial, type the following:

```
sudo docker-compose -f docker-compose-custom-data.yml down
```

3. Update Limelight

Use the following command to automatically download the latest version of all of the Limelight application components:

```
sudo docker-compose pull
```

4. (Optional) Remove Orphaned Docker Images

If new Limelight components are pulled down using the command above, the replaced Docker images will be “orphaned”. To remove the old Docker images and free up disk space, you can use the following command:

```
sudo docker system prune
```

You will see the following output:

```
WARNING! This will remove:
- all stopped containers
- all networks not used by at least one container
- all dangling images
- all dangling build cache

Are you sure you want to continue? [y/N]
```

Enter y and hit enter. The orphaned images will be removed.

5. Restart Limelight

Use the following command to bring Limelight back up using the new images:

If you have not customized the data locations for Limelight, type the following to start Limelight:

```
sudo docker-compose up --detach
```

If you have customized the data locations for Limelight by following our [Customize Data Locations](#) tutorial, type the following:

```
sudo docker-compose -f docker-compose-custom-data.yml up --detach
```

Major Version Update Instructions

Follow these steps to update your Limelight installation to a newer major version number. E.g., update your 2.x.x installation of Limelight to the latest release of 3.x.x.

This guide assumes you have completed all the steps in the [Installing Limelight](#) tutorial.

Important: Never update to a new major version that is more than one ahead of your current version! If you are on version 2.x.x (major version 2), you can only go to 3.x.x (major version 3). After 3.x.x is running, you may then update to version 4.x.x, and so on.

To update multiple steps (e.g., from major version 2 to 5), follow all of these steps to upgrade from major version 2 to 3, then repeat steps 8 and 9 once to update from major version 3 to 4, then repeat steps 8 and 9 again to update from major version 4 to 5.

You can see your current Limelight major version by following the instructions on our [Update Limelight](#) instructions. This must be done before following these steps.

1. Open a Terminal

On Linux and MacOS, open a normal terminal. On Windows, if you followed our instructions for installing Docker, follow the directions on our [Docker Installation Tutorial](#) to open a Linux terminal.

2. Backup the Limelight Database

It is critical that you backup your database before updating it. Please follow the instructions for backing up MySQL Data on the [Back Up and Restore Limelight Data](#) instructions page.

3. Shutdown Limelight

Go into your Limelight project directory:

```
cd ~/limelight
```

If you have not customized the data locations for Limelight, type the following to start Limelight:

```
sudo docker-compose down
```

If you have customized the data locations for Limelight by following our [Customize Data Locations](#) tutorial, type the following:

```
sudo docker-compose -f docker-compose-custom-data.yml down
```

4. Download the Latest Installation Package

Important: This process will overwrite your `docker-compose.yml` and `docker-compose-custom-data.yml` files. Most likely you have not changed either file, but if you have, back it up by typing:

```
# back up whichever file you have customized:
cp docker-compose.yml docker-compose-customized.yml
cp docker-compose-custom-data.yml docker-compose-custom-data-customized.yml
```

Download the latest Limelight Docker Compose files:

```
# Download Limelight Docker Compose files
curl -L "https://github.com/yeastrc/limelight-core/releases/latest/download/
↪ docker-compose-files.tgz" -o docker-compose-files.tgz
```

(continues on next page)

(continued from previous page)

```
# Expand the archive
tar -xvzf docker-compose-files.tgz
```

Note: If you prefer to download a ZIP file or if you prefer to download the file another way, the latest release can be found on GitHub at <https://github.com/yeastrc/limelight-core/releases/latest>

If you have customized your `docker-compose.yml` or `docker-compose-custom-data.yml` files, copy your customizations from the backup you made above into the new `docker-compose.yml` or `docker-compose-custom-data.yml` file.

5. Update Other Limelight Components

Use the following command to download the latest version of all of the Limelight application components:

```
sudo docker-compose pull
```

6. Add Any New .env File Entries

This is only necessary if you have customized your data storage locations, as described on our [Customize Data Locations](#) tutorial. If you have not customized the data storage locations for Limelight, you can skip this step.

You will need to configure any new data locations needed by the new release of Limelight. All required data locations are listed at our [Customize Data Locations](#) tutorial under step 2. Follow the directions for step 2 to add any missing configuration values to your `.env` file.

7. Bring up Limelight

If you have not customized the data locations for Limelight, type the following to start Limelight:

```
sudo docker-compose up --detach
```

If you have customized the data locations for Limelight by following our [Customize Data Locations](#) tutorial, type the following:

```
sudo docker-compose -f docker-compose-custom-data.yml up --detach
```

8. Perform Necessary Database Updates

Determine If Database Update Is Necessary

Type the following, but replace 3 in the first line with the major version number you are upgrading to.

```
export LIMELIGHT_NEW_VERSION="3"
ls database_scripts/version_upgrades/$LIMELIGHT_NEW_VERSION/version_upgrade.sql
```

Important: It is critical that you set the `LIMELIGHT_NEW_VERSION` to the correct version number. If you are upgrading from major version 2, you must set this number to 3. Never set this number to be more than 1 higher than your current version—doing so may corrupt your database.

If the response is something like:

```
ls: cannot access 'database_scripts/version_upgrades/3/version_upgrade.sql': No such
↪file or directory
```

Then you do *not* need to perform a database upgrade and you can skip this step and step 8.

If you see something like:

```
database_scripts/version_upgrades/3/version_upgrade.sql
```

Then you *do* need to perform a version upgrade.

Perform The Database Update

Run the following command to update the database. This assumes you ran the `export LIMELIGHT_NEW_VERSION=` command given above.

Note: `MYSQL_ROOT_PASSWORD` should be replaced with the actual root password for your MySQL installation. If you followed our [Installing Limelight](#) tutorial, this will be in your `.env` file as described on step 4 of the tutorial.

```
cat database_scripts/version_upgrades/$LIMELIGHT_NEW_VERSION/version_upgrade.sql | sudo
↪docker exec -i limelight-mysql sh -c 'exec mysql -u root -p"$MYSQL_ROOT_PASSWORD"
↪limelight'
```

9. Restart Limelight

If you did not perform a database update in step 7, you can skip this step.

If you have not customized the data locations for Limelight, type the following to start Limelight:

```
sudo docker-compose down
sudo docker-compose up --detach
```

If you have customized the data locations for Limelight by following our [Customize Data Locations](#) tutorial, type the following:

```
sudo docker-compose -f docker-compose-custom-data.yml down
sudo docker-compose -f docker-compose-custom-data.yml up --detach
```


10. (Optional) Remove Orphaned Docker Images

If new Limelight components are pulled down using the command above, the replaced Docker images will be “orphaned”. To remove the old Docker images and free up disk space, you can use the following command:

```
sudo docker system prune
```

You will see the following output:

```
WARNING! This will remove:
- all stopped containers
- all networks not used by at least one container
- all dangling images
- all dangling build cache

Are you sure you want to continue? [y/N]
```

Enter y and hit enter. The orphaned images will be removed.

4.3.3 Limelight SMTP Setup Guide

This guide assumes you have completed all the steps in the *Installing Limelight* tutorial.

Setting up SMTP allows Limelight to send emails to users. This enables you to invite researchers to projects using their email, users to reset forgotten passwords and to receive notifications when their file uploads are complete. Setting up SMTP is not required to run Limelight, but these features will not be available.

1. Acquire SMTP Relay Server Information

This is, potentially the most complex part of enabling SMTP in Limelight. Acquiring SMTP relay server information means finding the **host name**, **port**, **username**, and **password** to use to send email through an email server. Below we present some options on how to locate or set up SMTP relay server information.

Check With Your Organization

The best place to begin this process is checking with your organization’s IT department or your internet service provider to see if this service is available to you. If it is, it would potentially look something like:

Table 2: Hypothetical SMTP Relay Server Information

Host Name	smtp.organization.com
Port	587
Username	my_username
Password	my_password

SendGrid

SendGrid is a free service that allows applications like Limelight to send emails through their servers. The free tier allows you to send up to 100 emails per day. Here are the rough steps for setting up email sending capabilities:

1. Go to <https://sendgrid.com/>.
2. Create a free account
3. Establish a single sender, which is an email address from which you will be allowed to send emails through SendGrid. SendGrid will require that you verify you can receive emails at this address. **This does not give SendGrid access to your email.** It only allows you to send email **from this address** using SendGrid's servers. See [SendGrid's Guide for Single Sender Verification](#).
4. Click on the Email API navigation option on the left and choose Integration Guide. Click on the SMTP Relay option that appears on the page.
5. Follow the directions to acquire your SMTP relay server information.

Your SendGrid SMTP relay server information will look something like:

Table 3: SendGrid SMTP Relay Server Information

Host Name	smtp.sendgrid.net
Port	587
Username	apikey
Password	YOUR_API_KEY

Google SMTP Relay

Google allows your Google Workspace account to use their servers as a SMTP relay. To enable this, follow [Google's Guide for setting up SMTP relay](#). In the Authentication section, you want to enable Require SMTP Authentication.

Your Google SMTP relay server information will look something like:

Table 4: Google SMTP Relay Server Information

Host Name	smtp-relay.gmail.com
Port	587
Username	Google username
Password	Google password

Other Options

There are other service on the internet that provide SMTP relay server information. Any of them should work, so long as you have a **host name**, **port**, **username**, and **password**.

2. Update Your .env File

The .env file you set up during the *Installing Limelight* tutorial should contain the following lines (among others):

```
# Settings for setting up sending of emails by Limelight
SMTP_HOST=smtp.example.com
SMTP_PORT=587
SMTP_USERNAME=smtp_username
SMTP_PASSWORD=smtp_password
```

Open this file using your favorite text editor. On Linux (including Docker on Windows), we'll assume that is nano. To edit the file, type:

```
# ensure you are in correct directory. if you followed tutorial type:
cd ~/limelight

# edit the file
nano .env
```

Update these lines to reflect the SMTP relay server information from part 1. If you used SendGrid for your SMTP relay server, your information would be something close to:

```
SMTP_HOST=smtp.sendgrid.net
SMTP_PORT=587
SMTP_USERNAME=apikey
SMTP_PASSWORD=your API KEY goes here
```

Type Control-o, <ENTER>, and Control-x to save and exit nano.

3. Update Email Address for Sender in Limelight

1. Log into Limelight and click the ADMIN link in the top right. You must be logged in as an administrator user, such as the initial user created when you followed the *Installing Limelight* tutorial.
2. Click the Manage Configuration link.
3. Edit the field for From Address for emails sent. This is the email address from which emails sent by Limelight will appear to come. You may be restricted by what email address you can use here by the SMTP server you are using. For example, if you set up SMTP relay service with SendGrid, this email must match the verified sender you set up.
4. Click the Save button to save the changes.

4. Restart Limelight

Limelight must be restarted to use the new configuration settings in the .env file. Type the following into your terminal to restart Limelight:

```
# ensure you are in correct directory. if you followed tutorial type:
cd ~/limelight

# shutdown Limelight
sudo docker-compose down
```

(continues on next page)

(continued from previous page)

```
# startup Limelight
sudo docker-compose up --detach
```

5. Investigating Problems

If after following this guide, emails are not being sent, you can view the logs of the SMTP server by typing the following into a terminal:

```
sudo docker logs limelight-smtp
```

Carefully read this log and look for error messages, such as an authentication failure or other reasons the message may have been rejected.

4.3.4 Customize Data Locations

Note: This tutorial assumes you have completed our *Installing Limelight* tutorial through step 4.

By default, our installation tutorial will allow Docker to manage where Limelight stores its data. This includes things like where MySQL stores its data files, where uploaded scans are stored, and working directories for processing uploaded data. On Linux (including Windows running Ubuntu), these data will mostly likely be kept under `/var/lib/docker/`.

It is recommended that you let Docker manage the data directories if you can. However, if you would like to customize where the data are stored for Limelight, follow the steps below.

1. Create data directories

You will need to create the following directories for Limelight to store its data.

1. MySQL data directory. This is the directory used to store the database.
2. Spectr upload directory. This the directory used for spectra processing.
3. Spectr storage directory. This the directory used to store spectra.
4. Limelight upload directory. This is the directory where uploads are temporarily stored.
5. Blib conversion work directory. This is the directory where .blib conversions are processed.
6. Blib destination directory. This is the directory where final .blib files will be stored.

Note: If you are using WSL2 on Windows, specifying a Windows filesystem drive (e.g., `/mnt/d/`) for your data directories is not supported.

For example, if you would like store store all data in the `/data/limelight-data` directory, you would type the following:

```
# make a parent directory for limelight data
sudo mkdir -p /data/limelight-data
```

(continues on next page)

(continued from previous page)

```
# make the four directories for storing data
sudo mkdir /data/limelight-data/mysql
sudo mkdir /data/limelight-data/spectr-upload
sudo mkdir /data/limelight-data/spectr-storage
sudo mkdir /data/limelight-data/limelight-upload
sudo mkdir /data/limelight-data/blib-workdir
sudo mkdir /data/limelight-data/blib-download
```

2. Update .env with data storage locations

The `.env` configuration file will need to be updated to include the locations of the data directories. Open this file using your favorite text editor. On Linux (including Docker on Windows), we'll assume that is `nano`. To edit the file, type:

```
# ensure you are in correct directory. if you followed tutorial type:
cd ~/limelight

# edit the file
nano .env
```

Add the following lines to the end of the file. Substitute the actual directories with directories you chose above. This example uses the example directory names:

```
MYSQL_DATA_DIRECTORY=/data/limelight-data/mysql
SPECTR_UPLOAD_DIRECTORY=/data/limelight-data/spectr-upload
SPECTR_STORAGE_DIRECTORY=/data/limelight-data/spectr-storage
LIMELIGHT_UPLOAD_DIRECTORY=/data/limelight-data/limelight-upload
BLIB_WORK_DIR=/data/limelight-data/blib-workdir
BLIB_DESTINATION_DIR=/data/limelight-data/blib-download
```

Type `Control-o`, `<ENTER>`, and `Control-x` to save and exit `nano`.

3. Starting and Stopping Limelight

Important: The commands below are different than the commands for starting and stopping Limelight on our *Installing Limelight* tutorial! You must always use these commands if you have customized the data locations.

At this point, starting and stopping Limelight should be straight forward.

To start Limelight:

```
sudo docker-compose -f docker-compose-custom-data.yml up --detach
```

To stop Limelight:

```
sudo docker-compose -f docker-compose-custom-data.yml down
```

Note: The first time you start Limelight, all of the components will download and the database will initialize. This may take a few minutes, depending on your download speed. Subsequent startups of Limelight will not require these steps and will be faster.

Note: These commands must be typed while you are in the project code directory. If you followed these instructions, you can ensure you are in this directory by typing:

```
cd ~/limelight
```

4. Proceed with installation

You should now proceed to step 6 in our tutorial for installing limelight. However, recall that your command for stopping and starting is different than that listed in the tutorial. (See above.)

4.3.5 Manage Users (Add, Disable, Permissions)

This guide will walk you through how to add, remove, and manage the permissions of users of your Limelight installation.

Add Users

Users can be added to Limelight in two ways: project users inviting new users to projects and through the administration interface.

Invite Researchers to Projects

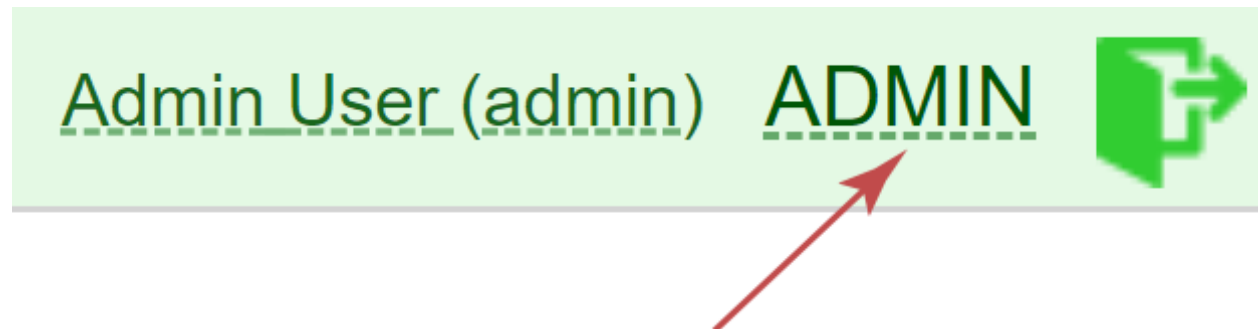
The primary way users are added to Limelight is for users of individual projects to invite new users to projects by sending an invitation to their email. This is covered in our user guide for adding researchers to a project. (Link forthcoming).

Note: This method only works if SMTP is configured for Limelight. Please see our [Limelight SMTP Setup Guide](#) for more information about setting that up.

Administration Interface

An alternative to inviting researchers to projects is to use the administration interface to create new users. If SMTP is not set up, this is the only option for adding new users to Limelight. To add users to Limelight, follow these steps.

1. Log into Limelight and click the ADMIN link in the top-right.



2. Click the Manage Users link.

Webapp Administration Main Page

[Manage Configuration](#)

[Manage Users](#)

Advanced Administration

[Manage Cached Data](#)

3. Click the Create User link.

Manage Users

[Webapp Administration Main Page](#)

Manage Users



[Create User](#)



Admin User



Administrator

3. Fill out the form and click Create Account to create a new user.

Important: The first field is the type of user to create. Choose User to create a normal user that has normal user

permissions. Choose **Administrator** to give complete access to all of Limelight, including all settings and adding and deleting projects and users.

Manage Users

[Webapp Administration Main Page](#)

Manage Users

Fill out the form below to create an account.

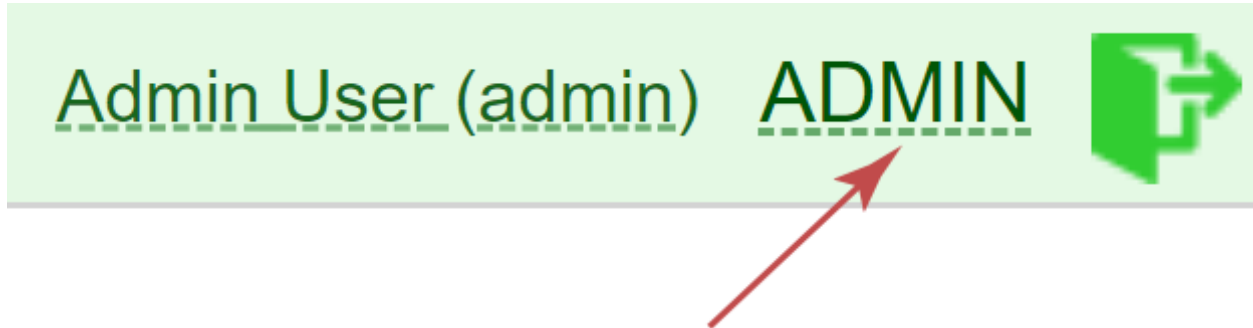
Administrator ▾	
First name	
Last name	
Organization	
Email address	
Username	
Password	
Confirm Password	
Create Account	Cancel

The created user will now be able to log into your installation of Limelight. They can create their own projects or be added to existing projects by the users of existing projects.

Disable Users

Users may be disabled using the administration interface. Disabled users will no longer be able to log into your installation of Limelight. To delete users, follow these steps:

1. Log into Limelight and click the ADMIN link in the top-right.



2. Click the Manage Users link.

Webapp Administration Main Page

Manage Configuration
Manage Users

Advanced Administration

Manage Cached Data

3. Click the circle with an X next to the user you wish to disable.

 Admin User  Administrator

 Test User  User

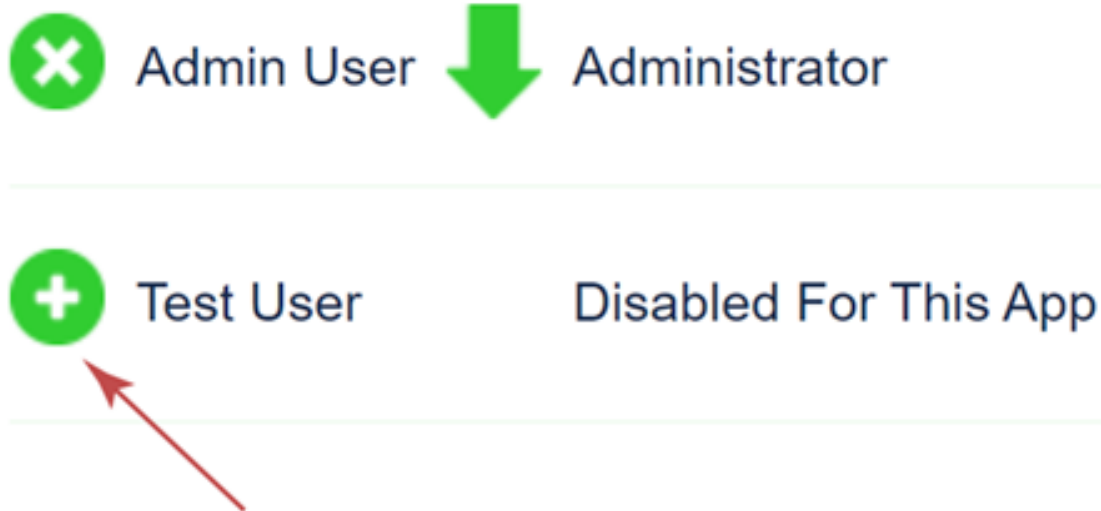
The user will now appear as disabled.

 Admin User  Administrator

 Test User Disabled For This App

Re-enable Users

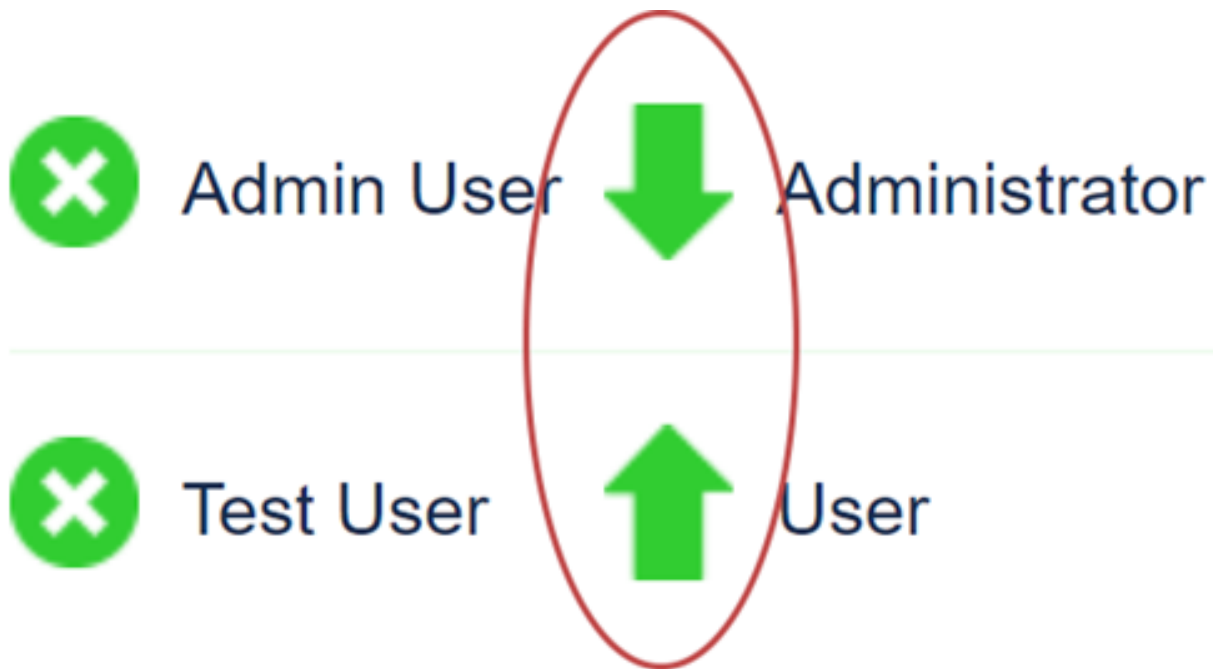
To re-enable this user, click the circle with a + next to the user you wish to re-enable.



Promote and Demote Users

Important: Administrators have complete access to all of Limelight, including all settings and adding and deleting projects and users. Proceed with caution.

You may promote normal users to Administrators or Administrators to normal users. To promote a user to Administrator, click the up arrow next to the user. To demote an Administrator to a normal user, click the down arrow.



4.3.6 Back Up and Restore Limelight Data

If you are running your own Limelight installation, this guide will provide direction on how to back up and restore your data. This guide will assume you have followed our [Installing Limelight](#) tutorial and are running Limelight using Docker and Docker Compose. Even if this is not the case, this guide should provide sufficient detail for you to back up and restore your data.

Back Up Data

Most data in Limelight are stored in MySQL, a relational database management engine. Spectra that are uploaded are stored as processed files on disk. Follow the sections below to back up each of these sets of data.

Note: Backing up will work best if no data imports are currently running.

Back Up MySQL Data

Important: The Limelight application must be running to back up MySQL in this way. [Review how to start and stop Limelight](#).

To back up a current snapshot of your data, use the `mysqldump` command. This can be done via Docker by typing the following into your Linux terminal (including Windows users running Docker on WSL2 as per our [Installing Docker](#) tutorial).

```
sudo docker exec limelight-mysql sh -c 'exec mysqldump --opt --max_allowed_packet=200M --
↪single-transaction -u root -p"MYSQL_ROOT_PASSWORD" limelight' | pigz --stdout > ~/
↪limelight-backup.sql.gz
```

If you are not running MySQL in Docker, the command would be:

```
mysqldump --opt --max_allowed_packet=200M --single-transaction -u root -p"MYSQL_ROOT_
↪PASSWORD" limelight | pigz --stdout > ~/limelight-backup.sql.gz
```

Note: `MYSQL_ROOT_PASSWORD` should be replaced with the actual root password for your MySQL installation. If you followed our [Installing Limelight](#) tutorial, this will be in your `.env` file as described on Step 4.

Note: If the `pigz` command is not found, `gzip` can be substituted into the command.

This will create a file in your home directory named `limelight-backup.sql.gz` that contains a compressed backup of the database.

To output this file with the name `limelight-backup-20210501.sql.gz` and to a different location, perhaps `D:\limelight-backups\` on a Windows machine, ensure that directory exists then run:

```
sudo docker exec limelight-mysql sh -c 'exec mysqldump --opt --max_allowed_
↪packet=200M --single-transaction -u root -p"MYSQL_ROOT_PASSWORD" limelight'
↪| pigz --stdout > /mnt/d/limelight-backups/limelight-backup-20210501.sql.gz
```

For extensive documentation of `mysqldump` see [the official MySQL website](#)

Back Up Spectra Data

The spectra data are files on disk, and so can be easily backed up using standard archiving and compression programs, like tar, gzip, or zip. To do this the files must first be located.

Docker-Managed Data Location

If Docker is managing where files are store (the default behavior in our tutorial), type the following into your Linux terminal (including Windows users running Docker on WSL2 as per our [Installing Docker](#) tutorial).

```
sudo docker inspect limelight_spectr_storage
```

The output should be similar to:

```
[
  {
    "CreatedAt": "2021-12-09T17:14:15-08:00",
    "Driver": "local",
    "Labels": {
      "com.docker.compose.project": "limelight",
      "com.docker.compose.version": "1.29.1",
      "com.docker.compose.volume": "spectr_storage"
    },
    "Mountpoint": "/var/lib/docker/volumes/limelight_spectr_storage/_data",
    "Name": "limelight_spectr_storage",
    "Options": null,
    "Scope": "local"
  }
]
```

The directory is the one labeled with Mountpoint. In this case, the data directory is: `/var/lib/docker/volumes/limelight_spectr_storage/_data`.

Customized Data Location

If you have customized the data storage locations, as per our [Customize Data Locations](#) tutorial, this will be the directory you have assigned to the `SPECTR_STORAGE_DIRECTORY` in your `.env` file. For example, if the line in your `.env` file reads:

```
SPECTR_STORAGE_DIRECTORY=/data/limelight-data/spectr-storage
```

Your directory is `/data/limelight-data/spectr-storage`.

Back Up the Data

Using the directory you found above, type the following into a Linux terminal to back up the spectra data:

```
# replace DIRECTORY_PATH with the directory you found above
sudo tar -C DIRECTORY_PATH -czf /path/to/limelight-spectra-backup.tgz .

# for example:
sudo tar -C /var/lib/docker/volumes/limelight_spectr_storage/_data -czf ~/limelight-
↪spectra-backup.tgz .
```

This will create a file named `limelight-spectra-backup.tgz` in your home directory that contains the spectra data. The `~/limelight-spectra-backup.tgz` part of the command may be replaced with any directory or filename you prefer to use for your backup. For example:

```
# for example:
sudo tar -C /var/lib/docker/volumes/limelight_spectr_storage/_data -czf /mnt/d/limelight-
↪backups/limelight-spectra-20210501.tgz .
```

Would save the backup in the `limelight-backups` directory of the `D:\` drive with the filename `limelight-spectra-20210501.tgz`.

Note: If you are on a different operating system or want to back up the data in a different way, any method for backing up the above directory should suffice.

Restore Data

Follow the guides below to restore your data from a backup.

Restore MySQL Data

Important: Restoring data will replace your existing data! You will lose any data created after the time of the backup file you are restoring.

Important: The Limelight application must be running to restore MySQL data in this way. [Review how to start and stop Limelight](#).

To restore a Limelight MySQL backup, type the following into your Linux terminal (including Windows users running Docker on WSL2 as per our Installing Docker tutorial).

```
zcat /path/to/limelight-backup.sql.gz | sudo docker exec -i limelight-mysql sh -c 'exec
↪mysql -u root -p"MYSQL_ROOT_PASSWORD" limelight'
```

If you are not using Docker to run MySQL, the command would be:

```
zcat /path/to/limelight-backup.sql.gz | mysql -u root -p"MYSQL_ROOT_PASSWORD" limelight
```

For example, if the backup file is named `limelight-backup-20210501.sql.gz` and is in `D:\limelight-backups\` on a Windows machine, then run:

```
zcat /mnt/d/limelight-backups/limelight-backup-20210501.sql.gz | sudo docker exec -i \
↳ limelight-mysql sh -c 'exec mysql -u root -p"MYSQL_ROOT_PASSWORD" limelight'
```

Note: `MYSQL_ROOT_PASSWORD` should be replaced with the actual root password for your MySQL installation. If you followed our [Installing Limelight](#) tutorial, this will be in your `.env` file as described on Step 4.

This will replace your Limelight MySQL database with the data in the backup file.

You should now stop and start the Limelight app. [Review how to start and stop Limelight.](#)

Restore Spectra Data

Important: Shut down the Limelight app before restoring spectra data! [Review how to start and stop Limelight.](#)

To restore spectra data, determine the directory in which the spectra data should be found by [following the directions above](#).

Type the following into a Linux terminal to restore a spectra data backup:

```
sudo tar -xzf /path/to/backup.tgz -C /path/to/spectra/data
```

`/path/to/backup.tgz` is the location of your backup file made with the instructions above. `/path/to/spectra/data` is the location where the spectra data should go, determined by following the instructions above.

For example, to restore a backup file named `limelight-spectra-20210501.tgz` found in `D:\limelight-backups` to the directory `/var/lib/docker/volumes/limelight_spectr_storage/_data` type the following:

```
sudo tar -xzf /mnt/d/limelight-backups/limelight-spectra-20210501.tgz -C /var/lib/docker/
↳ volumes/limelight_spectr_storage/_data
```

You can now start Limelight. [Review how to start and stop Limelight.](#)

Getting Help

If you have any questions, bug reports, or feature requests, please contact us!

- Try using the [Limelight Issue Page](#) at GitHub.
- You can [join our Slack](#) and chat with us.
- You can email us at limelightms@uw.edu