

# locFISH - How to create a deepzoom image to inspect t-SNE visualization results

Here we describe how the t-SNE representation of cells based on their mRNA localization features can be used to create a zoomable high-resolution image (DZI = deep zoom image). In this image, the actual data-points are replaced by a thumbnail showing the detected mRNAs. This permits an interactive and fast inspection of the data. However, it also requires a little bit of work to set this up. Below a detailed workflow is provided.

<b>Overview.....</b>	<b>2</b>
<b>Create high-resolution image of cell classification .....</b>	<b>3</b>
Parameters for the creation of a high-res image.....	3
<b>Create the actual dzi file .....</b>	<b>4</b>
Install libvips .....	4
Use Libvips pyramid builder .....	4
<b>Visualize the dzi image .....</b>	<b>5</b>
Set-up website generator to view dzi images.....	5
Visualize dzi image locally – webserver with Python.....	5
Publish dzi image on a website – option 1: Surge.....	6
Publish dzi image on a website – option 2: GitHub Pages.....	6

## Overview

Creating a deepzoom image (dzi ) consists of three main steps:

1. Create a high-resolution image of the t-SNE visualization. This is done with **Matlab package FISH-sim**.
2. Create a image pyramid. This is essentially a sequence of image-tiles with different resolution settings, which are used for dynamic zooming. This is done with the freely available image processing library **libvips**.
3. In order to visualize the dzi image, a website has to be created. This can be either done locally or on a publically available website. Either options are explained below.

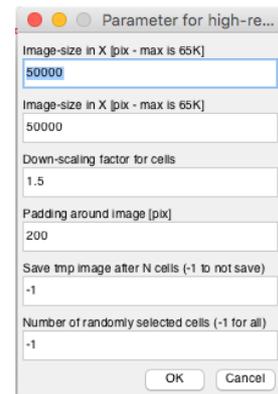
## Create high-resolution image of cell classification

This image is created in Matlab with the user interface `exp_data_look_up`. First, you have to load the csv file with the localization features for the genes you analyzed. Then you have to select the genes that you want to process and perform the tSNE. Once the tSNE is calculated, you can generate the high-resolution image from the menu **Advanced tools > Create high-res image**. This will then generate a high-resolution image based on some default parameters explained next. The resulting image is saved as `tsne_highres.tif` in a subfolder called `dzi_tsne`, relative to the folder containing the csv file with the localization features. During the processing a progressbar is shown indicating how long the generation of the image will approximately take.

### Parameters for the creation of a high-res image

A number of parameters determine how the high-resolution image is created. They can be changed from the menu **Advanced tools > Options for high-res image**. This will open a dialog box as shown below

- **Image-size** determines how large the image will be in X and Y
- **Down-sampling** allows to decrease the size of the cells that will be added to this image.
- **Padding** is a white-space that will be added around the large image to facilitate placement of cells close to the border
- You can save a **temporary image** each a defined number of cells are added. This allows to inspect the result before the final image is created. This image is saved in the same folder as the final image under the name `tsne_highres_tmp.tif`
- You can also choose to not plot all cells but only a **randomly selected sub-sample**. This can be useful when the plot is getting too crowded. However, the creation of the plot is slower since cells are selected randomly and each time the corresponding FISH-quant outline files has to be opened.



## Create the actual dzi file

### Install libvips

Libvips is a image processing library that allows creating the image pyramid used by deepzoom. Libvips can be found together with installation instructions at

<http://jcupitt.github.io/libvips/install.html>

Note, for Mac, installation of homebrew is required: <https://brew.sh/>

### Use Libvips pyramid builder

Libvips provides a function called `dzsave` that generates image pyramids used by the deepzoom format.

1. Open a terminal at the folder containing the high-resolution image created above in Matlab.
2. Type

```
vips dzsave tsne_highres.tif tsne_dzi__demo --tile-size 2000
```

This function call will create a deepzoom image from the high-resolution image named “tsne\_highres.tif”. More specifically, it will create a folder named “tsne\_dzi\_\_demo\_files” and a file called “tsne\_dzi\_\_demo.dzi”. Both are needed! While processing, a temporary file is created with the name of final folder with an added random suffix. Once the processing is finished, the above-mentioned files will appear.

3. For more options see the help of vips by typing `vips dzsave`. In the example code above, we change the default tile-size to reduce the produced data-volume.

## Visualize the dzi image

A very convenient way to visualize the dzi image is by using a **web-browser**. This can be done either locally or on a publically hosted server. In either case, you first have to obtain a minimal website providing dzi viewer. These files can then be used locally or publically.

### Set-up website generator to view dzi images

1. Download the **DZI viewer** from (file with extension *.tar.gz*)  
<https://github.com/davidmclure/osd-dzi-viewer/releases>
2. Unpack the downloaded archive. This archive contains all files needed for the local webserver. You can also rename this folder if you wish, e.g. `locFISH_deepzoom`.
3. **Copy ALL the dzi files** generated by the pyramid builder into this folder. However, you have to respect a certain naming convention
  - a. Create a parental folder, which can contain different DZI image, e.g. name this folder "results"
  - b. Within this folder create another folder with the same name as the actual deepzoom files (the folder and the *.dzi* file), e.g. `tsne_exp` in our example.
  - c. Copy the deepzoom files into this folder.
4. The **final file and folder structure** should look like this



### Visualize dzi image locally – webserver with Python

If you want to inspect the file locally, you can do this with a local web-server powered by Python. You need to download additionally a DZI viewer.

1. In case you don't have **Python** installed, we recommend installing Anaconda.  
<https://www.continuum.io/downloads>
2. Open a new terminal in the folder containing the `index.html` file, and start web server with:

Python 3	<code>python -m http.server</code>
Python 2	<code>python -m SimpleHTTPServer</code>

This should look like this.

```
osd-dzi — python -m SimpleHTTPServer — 85x12
Last login: Tue Jul 11 14:50:06 on ttys004
pilon:osd-dzi fmueller$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

3. Open the webserver in your default browser  
[http://localhost:8000/#results/tsne\\_exp](http://localhost:8000/#results/tsne_exp)

where 8000 is the port listed in the window above, and the folder names are as specified above, e.g for our example `#mRNA1oc/tsne_dzi_demo`

- **DON'T FORGET THE #** before the name of the parental folder!
- **Don't add a "/"** at the end of the web address!

Please also note that you can **have only one webserver open at a time**. To visualize another dzi image, you first have to close the terminal and open a new one.

## Publish dzi image on a website – option 1: Surge

An easy way to make a website is by using surge: <http://surge.sh/help/getting-started-with-surge>

1. You basically have to follow the basic steps outlined in the “getting started” section on the surge website from within the directory containing the “index.html” file. If successful, the output on the command line will look like this



```
dzl_webserver -- -bash -- 121x20
Last login: Wed Jul 12 14:15:51 on ttys003
[redacted]@[redacted]: surge

Surge - surge.sh
  email: [redacted]
  token: [redacted]
  project path: [redacted]/dzi_tsne/dzl_webserver/
  size: 154 files, 13.1 MB
  domain: merciful-robin.surge.sh
  upload: [redacted] 100%, eta: 0.0s
  propagate on CDN: [redacted] 100%
  plan: Free
  users: [redacted]
  IP Address: 45.55.110.124

Success! Project is published and running at merciful-robin.surge.sh
```

2. In order to access the dzi image, you have to type the web-address returned by surge, and add the folder names as specified above for the local html server, e.g.

[http://merciful-robin.surge.sh/#mRNAloc/tsne\\_dzi\\_demo](http://merciful-robin.surge.sh/#mRNAloc/tsne_dzi_demo)

**IMPORTANT:** (1) don't leave out the # before the folder name; (2) no "/" at the end of the web-address!

3. In order to remove a web-site, you have to “tear it down”, see surge website for details.

## Publish dzi image on a website – option 2: GitHub Pages

Another easy way (but slightly more complicated way than Option 1) to host a dzi image is by using GitHub Pages (<https://pages.github.com/>). However, this requires using git, which might be a little confusing at the beginning.

1. Create a GitHub account (<https://github.com>) and install git (<https://git-scm.com>).
2. Create a GitHub repository, e.g. locFISH\_deepzoom.
3. Commit the folder containing all files for the website and the dzi image as specified above. For this, open a terminal at the folder containing index.html file (locFISH\_deepzoom in our example). Most of these commands is also shown after creating the GitHub repository. The one command not being listed is the “git add -A”. The 4<sup>th</sup> line has to be changed to your specific repository name

```
git init
git add -A
git commit -m "first commit"
git remote add origin https://github.com/muellerflorian/locFISH_deepzoom.git
git push -u origin master
```

4. In order for Github-Pages to work properly, you have to create an empty file called “.nojekyll” in the root folder of the GitHub repository. This can be done directly within GitHub by going to the “<>Code” section, and pressing on “Create new file”. In order to really add it to the repository, don't forget to press on the green “Commit new file” button at the bottom. This file disables Jekyll functions on Github-Pages
5. The final repository should look like this

results/tsne_exp	Commit of dzi of experimental data	9 minut
.nojekyll	Create .nojekyll	4 minut
index.html	Commit of dzi of experimental data	9 minut
script.js	Commit of dzi of experimental data	9 minut
style.css	Commit of dzi of experimental data	9 minut

6. Setup GitHub Pages from the Settings panel. Select “master branch” for the source, and press save.

## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

✓ Your site is published at [https://muellerflorian.github.io/deepzoom\\_visualization/](https://muellerflorian.github.io/deepzoom_visualization/)

**Source**  
Your GitHub Pages site is currently being built from the `master` branch. [Learn more.](#)

`master branch`

7. It might take a while until the page is build. You can now access the dzi image via and URL such as [https://muellerflorian.github.io/FISHsim\\_dzi/#mRNAloc/tsne\\_dzi\\_demo](https://muellerflorian.github.io/FISHsim_dzi/#mRNAloc/tsne_dzi_demo)  
[https://muellerflorian.github.io/locFISH\\_deepzoom/#results/tsne\\_exp](https://muellerflorian.github.io/locFISH_deepzoom/#results/tsne_exp)

where the last part “/#results/tsne\_exp” is the same reference to the dzi image as used for the local webserver explained above (and importantly, the same considerations apply!).