

---

# **vipertools**

***Release 0.0.1***

**Sophia Maedler**

**Jun 01, 2022**

## CONTENTS:

<b>1</b>	<b>parse</b>	<b>1</b>
<b>2</b>	<b>stitch</b>	<b>2</b>
<b>3</b>	<b>Tutorials</b>	<b>4</b>
	<b>Python Module Index</b>	<b>5</b>
	<b>Index</b>	<b>6</b>

## PARSE

Contains functions to parse imaging data into a usable formats for downstream pipelines.

`vipertools.parse.parse_phenix(phenix_dir, flatfield_exported=True, parallel=False)`

Function to automatically rename TIFS exported from Harmony into a format where row and well ID as well as Tile position are indicated in the file name. Example of an exported file name: “Row{#}\_Well{#}\_{channel}\_zstack{#}\_r{#}\_c{#}.tif”

### Parameters

- **phenix\_dir** – Path indicating the exported harmony files to parse.
- **flatfield\_exported** – boolean indicating if the data was exported from harmony with or without flatfield correction.
- **parallel** – boolean value indicating if the data parsing should be performed with parallelization or without (CURRENTLY NOT FUNCTIONAL ONLY USE AS FALSE)

## STITCH

Collection of functions to perform stitching of parsed image Tiffs.

```
vipertools.stitch.generate_stitched(input_dir, slidename, pattern, outdir, overlap,  
                                   stitching_channel='Alexa488', crop={'bottom': 0, 'left': 0, 'right':  
                           0, 'top': 0}, plot_QC=False, filetype=['.tif'])
```

Function to generate a scaled down thumbnail of stitched image. Can be used for example to get a low resolution overview of the scanned region to select areas for exporting high resolution stitched images.

### Parameters

- **input\_dir** – Path to the folder containing exported TIF files named with the following nameing convention: “Row{#}\_Well{#}\_{channel}\_zstack{#}\_r{#}\_c{#}.tif”. These images can be generated for example by running the `vipertools.parse.parse_phenix()` function.
- **pattern** – Regex string to identify the naming pattern of the TIFs that should be stitched together. For example: “Row1\_Well2\_{channel}\_zstack3\_r{row:03}\_c{col:03}.tif”. All values in { } indicate thos which are matched by regex to find all matching tifs.
- **outdir** – path indicating where the stitched images should be written out
- **overlap** – value between 0 and 1 indicating the degree of overlap that was used while recording data at the microscope.
- **name** – string indicating the slidename that is added to the stitched images generated
- **stitching\_channel** – string indicating the channel name on which the stitching should be calculated. the positions for each tile calculated in this channel will be passed to the other channels.
- **crop** – dictionary of the form {‘top’:0, ‘bottom’:0, ‘left’:0, ‘right’:0} indicating how many pixels (based on a generated thumbnail, see `vipertools.stitch.generate_thumbnail`) should be cropped from the final image in each indicated dimension. Leave this set to default if no cropping should be performed.
- **plot\_QC** – boolean value indicating if QC plots should be generated
- **filetype** – list containing any of [“.tif”, “.ome.zarr”, “.ome.tif”] defining to which type of file the stiched results should be written. If more than one element all export types will be generated in the same output directory.

```
vipertools.stitch.generate_thumbnail(input_dir, pattern, outdir, overlap, name,  
                                    stitching_channel='DAPI', export_examples=False)
```

Function to generate a scaled down thumbnail of stitched image. Can be used for example to get a low resolution overview of the scanned region to select areas for exporting high resolution stitched images.

### Parameters

- **input\_dir** – Path to the folder containing exported TIF files named with the following nameing convention: “Row{#}\_Well{#}\_{channel}\_zstack{#}\_r{#}\_c{#}.tif”. These images can be generated for example by running the `vipertools.parse.parse_phenix()` function.

- **pattern** – Regex string to identify the naming pattern of the TIFs that should be stitched together. For example: “Row1\_Well2\_{channel}\_zstack3\_r{row:03}\_c{col:03}.tif”. All values in { } indicate those which are matched by regex to find all matching tifs.
- **outdir** – path indicating where the stitched images should be written out
- **overlap** – value between 0 and 1 indicating the degree of overlap that was used while recording data at the microscope.
- **name** – string indicating the slidename that is added to the stitched images generated
- **export\_examples** – boolean value indicating if individual example tiles should be exported in addition to performing thumbnail generation.

**TUTORIALS**

pages/tutorials

## PYTHON MODULE INDEX

### V

`vipertools.parse`, ??

`vipertools.stitch`, [1](#)

## INDEX

### G

`generate_stitched()` (*in module vipertools.stitch*),  
2  
`generate_thumbnail()` (*in module vipertools.stitch*),  
2

### M

module  
    vipertools.parse, 1  
    vipertools.stitch, 1

### P

`parse_phenix()` (*in module vipertools.parse*), 1

### V

`vipertools.parse`  
    module, 1  
`vipertools.stitch`  
    module, 1