Differential transcript expression in relation to hypoxia distributions

# Overview

Goal is to identify a list of genes which exhibit both a statistically significant and high extent of differential gene expression in regions of hypoxia vs normoxia in spatial transcriptomics (Visium) datasets. Two prerequisites are required: one tissue section processed using the standard Visium workflow, and a serial section stained for a marker of hypoxia (e.g. pimonidazole). The PIMO image will be aligned to the H&E section (generated from the Visium workflow), as per directed by the **Image Alignment.docx** SOP. Once aligned, annotations will be generated based on PIMO intensity to delineate positive vs negative regions. These annotations will be exported as binary masks, and used to select spatial transcriptomics points, using Giotto, Spaniel, or some other spatial transcriptomics analysis platform.

# Instructions

* Perform image alignment as outlined in **Image Alignment.docx** of the PIMO to the H&E. H&E must be the reference image since we can’t move the spatial transcriptomics data underlying it
* Create annotations corresponding to a) PIMO positive b) pimo negative but still viable tissue and c) background/artifact areas
  + One example is in the *PIMO to ST alignment* project, **pos\_gaus4\_thresh0.12** generates pimo positive areas by applying a gaussian blur of 4 and a threshold of deconvolved PIMO of 0.12
  + Manual corrections of annotations needs to be done to clean things up.
* Export annotations as a binary mask using **export annotations.groovy**
  + Currently set to export a grayscale image, such that all pixels with a value of 1 are from `PIMO positive`, 2 is `PIMO negative`, and 3 is `background`.
* Perform DGX of these two areas using Giotto spatial, spaniel, or some other R package

## Giotto

* Follow installation instructions for creating a docker image here: <http://spatialgiotto.rc.fas.harvard.edu/giotto.install.docker.html>
* Setting up the container (tutorial 1.2):
  + **Run container** docker run --name giotto-test -it giotto-1.0.2 /bin/bash
  + **Start R studio** R
  + **Load giotto library** library(Giotto)
  + **Quit R** quit()
  + **Exit container** exit
  + **Delete container** docker rm giotto-test
* 1.3 – real run

**Designate directory to share with giotto:** docker run -it --user dean --name giotto-1.0.2 -p 8000:8000 -p 8787:8787 -v "C:\Users\Mark Zaidi\Documents\Visium:/data" giotto-1.0.2 /bin/bash

Path on Windows OS: C:\Users\Mark Zaidi\Documents\Visium

Path on Linux OS: /data

**Restart R studio in case not started**: sudo rstudio-server restart

Password is dean

Note, disabled “use the WSL 2 based engine”. Enabling may improve performance, disabling is easier to allocate CPU and memory resources.

**First time docker usage**

**Declare some variables in R**

setwd("/data")

my\_working\_dir = "/data"

python\_path = "/usr/bin/python3"

**Starting up docker and R**

docker start giotto-1.0.2

docker attach giotto-1.0.2

sudo rstudio-server restart

*(in browser*): http://localhost:8787

## Giotto – Visium brain example

<http://spatialgiotto.rc.fas.harvard.edu/giotto.visium.brain.html>

For this example, we’ll be using our D1 sample in place of the tutorial data

Set some path variables. Currently, data is mapped to the Visium folder

setwd("/data")

data\_path='GBM\_datasets/D1\_-\_35/outs/'

workdir="GBM\_datasets/D1\_-\_35/outs/Giotto results/"

Make sure that the .gz and only the .gz unextracted files are in the specified input folders. Otherwise, the uncompressed files can throw errors when trying to detect the appropriate input files (i.e. through get10xmatrix)

2. Create Giotto object & process data

When using createGiottoObject to create a giotto object, might be a way to create one for PIMO positive only cells

So far: done one run through of the Visium brain example, substituting D1 sample in place of the tutorial. Next: rerun tutorial with original tutorial data, and sampled B1,C1,A1

Also, review what each of the figures do. And remember, in “Visualize the results of silhouetteRank:”, run each line one at a time, rename the figures to prevent overwrite

And review what each of the figures mean

**You made an R program to automate this, include in SOP**

To do

* make 2 more copies of tissue position list, change “in tissue” column based on whether cell falls within pimo positive, or pimo negative
  + actually don’t do this, find some way to manually assign cluster categories using this
* fix bug with sample C1, maybe post an issue on the github repo
* “inject” deconvolved PIMO stain as a gene. Measure the mean deconvolved pimo for each spot, normalizing intensities across some distribution. Imaging data and rnaSeq are intrinsically different from one another, so how do we get a realistic distribution? Find the distribution of a hypoxic gene, like VEGFA, and “map” the intensities to mimic the VEGFA distribution

## Tips on using giotto

* visualization cheat sheet: <https://rdrr.io/github/RubD/Giotto_site/f/inst/howtos/visualization_options/visualization_options.Rmd>