 # 005

Stack-Tools for evaluating collagen networks

A frequent task in our institute is to evaluate the poresize statistics of collagen networks from 3D image stacks of the gels. Using a set of executable programs that can be started with command line options, this task can be achieved with the following steps:

Data packing: The raw data consists of many cross-sectional confocal images (typically in the TIF format) of the gel. For easier handling and for (loss-less) data compression, the many separate images are first packed into a PIS (for Packed Intensity Stack) file, using the program SPacker. Note that the free application *ImageMagick* must be installed on the computer in order to read and pack TIF images.

Binarization: The PIS file, which still containes all the gray levels, is then binarized and skeletonized with the latest algorithm of Patrick Krauss, using the program SBinarizer. It produces a PBS (for Packed Binary Stack) file and can handle, both, reflection stacks (with only more or less horizontal fibers) and fluorescence stacks (with fibers running in all 3 directions).

Inspection of results: The PBS is still in a packed format. In order to inspect it, it can be converted into a stack of binary images in the BMP format using the program SUnpacker. Alternatively, the PBS file can be directly viewed using the program SViewer, which allows a 3D rotation of a selected region of interest (ROI) of the PBS.

Statistical analysis: The PBS file can also be fed into the program SAnalyzer, which computes the distribution $P(r_{nod})$ of nearest obstacle distances. The distribution is stored as a dat-file with x-y colums. The program also outputs an info-file with the minimum, maximum and average of r_{nod}. The latter is directly related to the average pore size of the collagen network.

This *ZIP-file* contains all executables, a sample batch file (doit.bat) and sample raw data (stack, 30MB) Please let me know if there are any problems or suggestions for improving the programs.

SPacker:

Example:

```
SPacker -f stack -p img###.tif -s 1 -e 50
-o my.pis -m 2 -x 1.0 -y 1.0 -z 1.0
```

Options:

-f folder of input
-p pattern of image filenames with 1-4 hash wildcards (###)
-s start number of image
-e end number of image
-o (path and) filename of resulting pis-file
-m mode (1=fluorescence, 2=reflection)
-x x-size of voxel in μm
-y y-size of voxel in μm
-z z-size of voxel in μm

SBinarizer:

Example:

```
SBinarizer -i my.pis -o my.pbs
```

Options:

-i (path and) filename of the pis-file to be binarized
-o (path and) filename of the resulting pbs-file

SUnpacker

Example:

```
SUnpacker -i my.pbs -o stack/bin
```

Options:

-i (path and) filename of the pis- or pbs-file to be unpacked
-o (path and) filename of the resulting images without extension

---

SAnalyzer:

Example:

```
SAnalyzer -i my.pbs -o res -m -1.0 -p -1
```

Options:

-i (path and) filename of the pbs-file to be analyzed
-o (path and) filename of the outputs without extension
-m max. r_{nod} (in μm) for histogram (neg. $\rightarrow$ full range)
-p nr. of test points (neg. $\rightarrow$ standard value)

SViewer:

Example:

```
SViewer -f my.pbs -t -1
-x0 -1 -x1 -1 -y0 -1 -y1 -1 -z0 10 -z1 20
```

Options:

-f (path and) filename of the pbs-file to be analyzed
-t threshold (neg. $\rightarrow$ automatic)
-x0 start of x-ROI in voxels (neg. $\rightarrow$ full range)
-x1 end of x-ROI in voxels (neg. $\rightarrow$ full range)
-y0 start of y-ROI in voxels (neg. $\rightarrow$ full range)
-y1 end of y-ROI in voxels (neg. $\rightarrow$ full range)
-z0 start of z-ROI in voxels (neg. $\rightarrow$ full range)
-z1 end of z-ROI in voxels (neg. $\rightarrow$ full range)

CM 2012/10/03