Supplementary Information

Contents

Title

Exploring AdaBoost and Random Forests machine learning approaches for infrared pathology on unbalanced data sets

Authors

Jiayi Tang, Alex Henderson and Peter Gardner, University of Manchester, UK

Data and source code availability

The data and MATLAB source code used in this study, and to generate this document, is freely available from the Zenodo data repository here: http://doi.org/10.5281/zenodo.4730312

This document uses *ChiToolbox* to pre-process data and generate figures. ChiToolbox is available here: http://bitbucket.org/AlexHenderson/chitoolbox

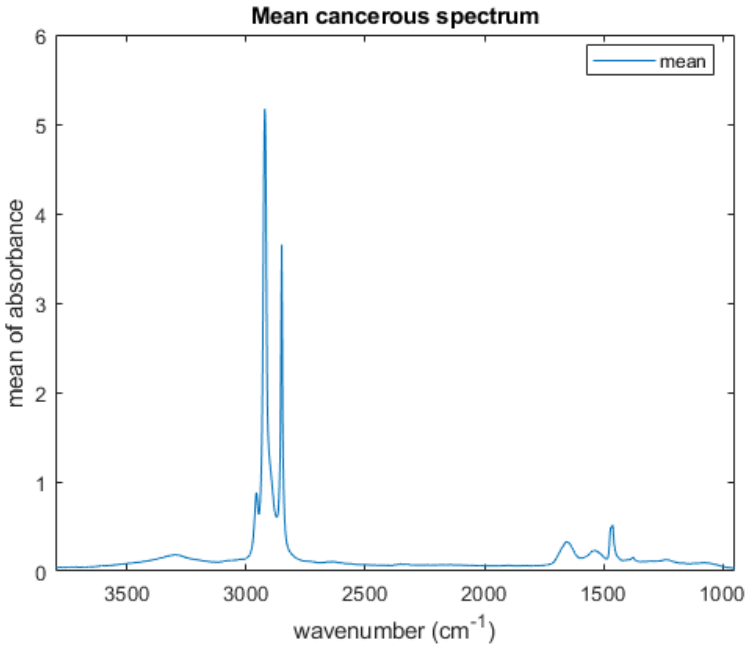
Define plotting options

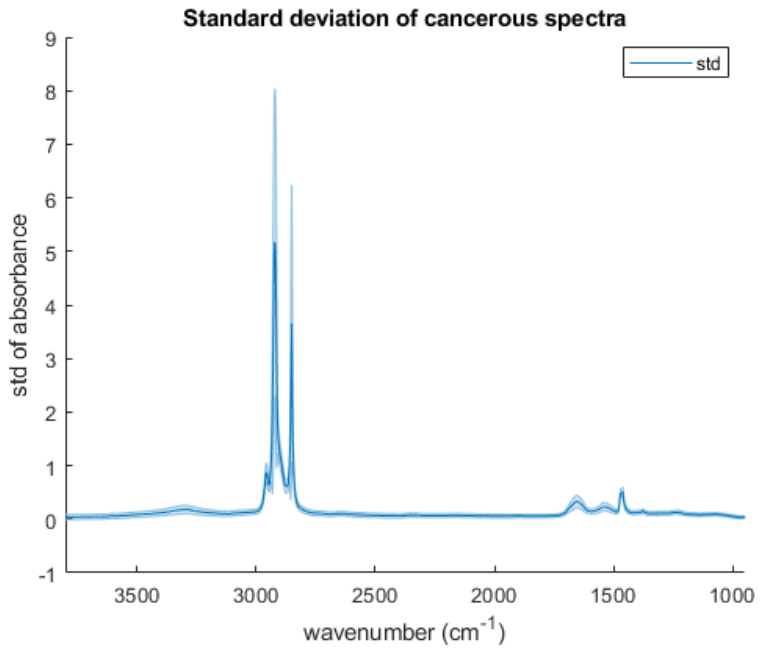
```
showplots = true;
```

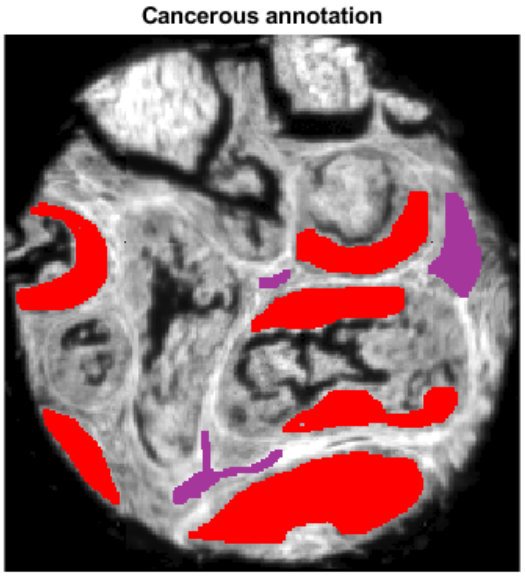
File D8.mat contains FTIR spectra from core id D8 and corresponds to a cancerous tissue core

```
% Read cancerous core spectra into memory
d8mat = load('D8.mat');

% Create a ChiIRSpectralCollection object containing the core's data
d8spectra = ChiIRSpectralCollection(d8mat.wavenumbers,d8mat.spectra);

% Read cancerous core tissue annotation into a ChiImageFile object
d8png = ChiImageFile('D8.png');

% Make space in memory for 3D array of cancerous core data
d8data = zeros(d8png.height * d8png.width, d8spectra.numchannels);

% Insert cancerous spectra into correct locations in the 3D array
d8data(d8mat.indices,:) = d8spectra.data;

% Convert 3D array into a ChiIRImage hyperspectral image object
d8image = ChiIRImage(d8mat.wavenumbers,d8data,d8png.width,d8png.height);


% Define colour of a 'red' pixel
redpixel = [255,0,0]; % RGB

% Identify 'red' pixels in the annotation
d8redmask = d8png.createmask(redpixel);

% Extract 'red' pixel spectra into a ChiIRSpectralCollection object
d8redspectra = d8image.applymask(d8redmask);


% Define colour of a 'purple' pixel
purplepixel = [165,55,156]; % RGB

% Identify 'purple' pixels in the annotation
d8purplemask = d8png.createmask(purplepixel);
```

```
% Extract 'purple' pixel spectra into a ChiIRSpectralCollection object
d8purplespectra = d8image.applymask(d8purplemask);

% Show plots if requested
if showplots
    % Mean spectrum
    d8spectra.plot('mean', 'title','Mean cancerous spectrum')

    % Standard deviation of spectra
    d8spectra.plot('std', 'title','Standard deviation of cancerous spectra')

    % Annotation
    d8png.display('title','Cancerous annotation')
end
```

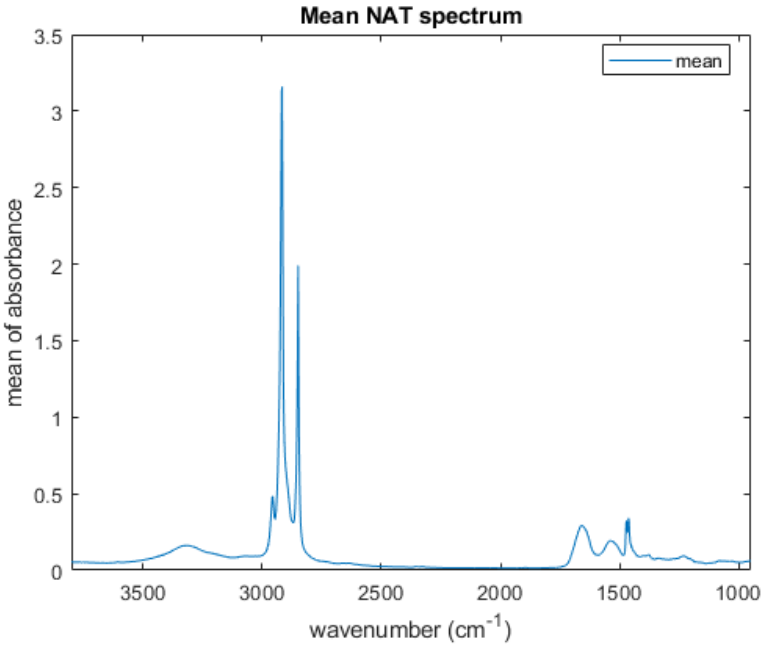


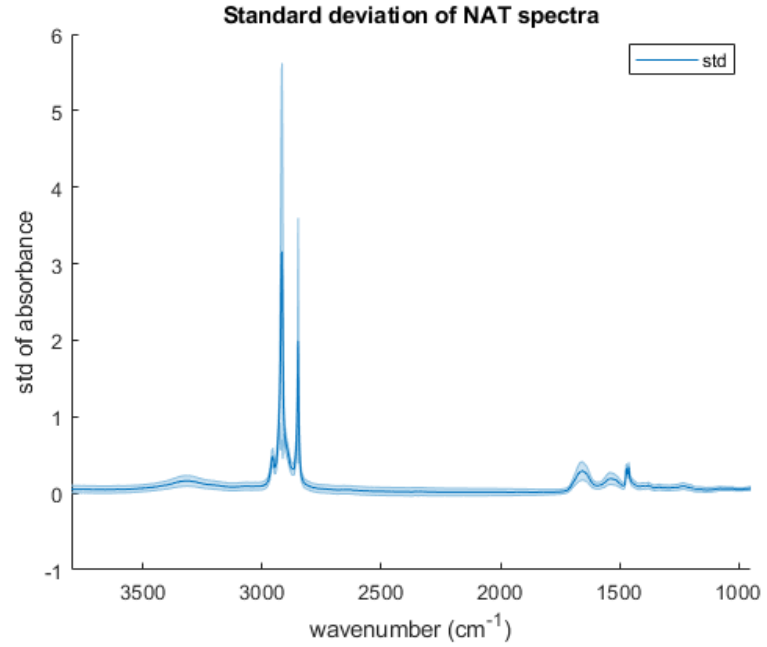


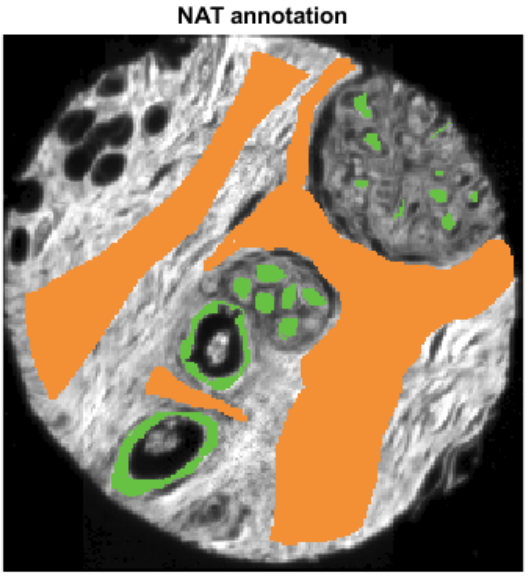


File M4.mat contains FTIR spectra from core id M4 and corresponds to a normal associated tissue core

```
% Read NAT core spectra into memory
m4mat = load('M4.mat');

% Create a ChiIRSpectralCollection object containing the core's data
m4spectra = ChiIRSpectralCollection(m4mat.wavenumbers,m4mat.spectra);

% Read NAT core tissue annotation into a ChiImageFile object
m4png = ChiImageFile('M4.png');

% Make space in memory for 3D array of NAT core data
m4data = zeros(m4png.height * m4png.width, m4spectra.numchannels);

% Insert NAT spectra into correct locations in the 3D array
m4data(m4mat.indices,:) = m4spectra.data;

% Convert 3D array into a ChiIRImage hyperspectral image object
m4image = ChiIRImage(m4mat.wavenumbers,m4data,m4png.width,m4png.height);

% Define colour of a 'green' pixel
greenpixel = [103,193,66]; % RGB

% Identify 'green' pixels in the annotation
m4greenmask = m4png.createmask(greenpixel);

% Extract 'green' pixel spectra into a ChiIRSpectralCollection object
m4greenspectra = m4image.applymask(m4greenmask);

% Define colour of an 'orange' pixel
orangepixel = [243,143,53]; % RGB

% Identify 'orange' pixels in the annotation
m4orangemask = m4png.createmask(orangepixel);

% Extract 'orange' pixel spectra into a ChiIRSpectralCollection object
m4orangespectra = m4image.applymask(m4orangemask);

% Show plots if requested
if showplots
    % Mean spectrum
    m4spectra.plot('mean', 'title','Mean NAT spectrum')

    % Standard deviation of spectra
    m4spectra.plot('std', 'title','Standard deviation of NAT spectra')

    % Annotation
    m4png.display('title','NAT annotation')
end
```

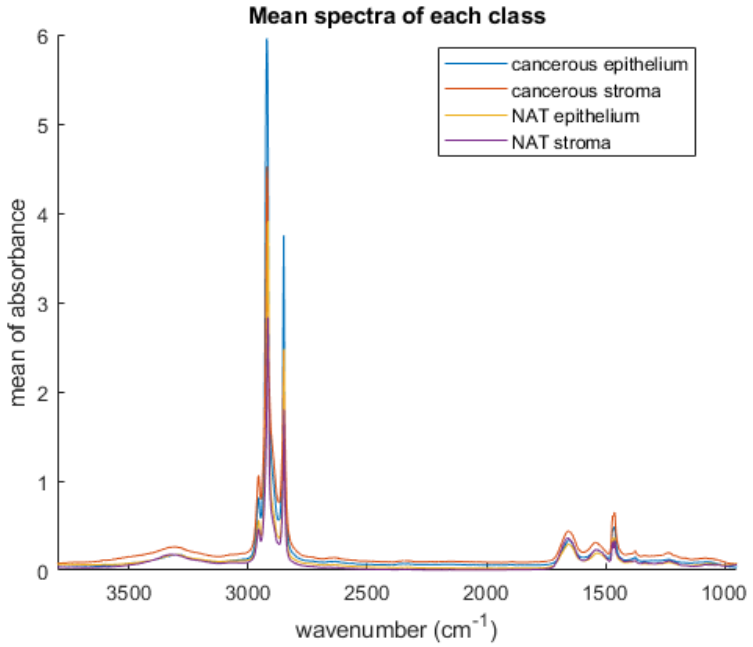

Mean NAT spectrum
mean
mean of absorbance
wavenumber (cm-1)
3500
3000
2500
2000
1500
1000
0
0.5
1
1.5
2
2.5
3
3.5

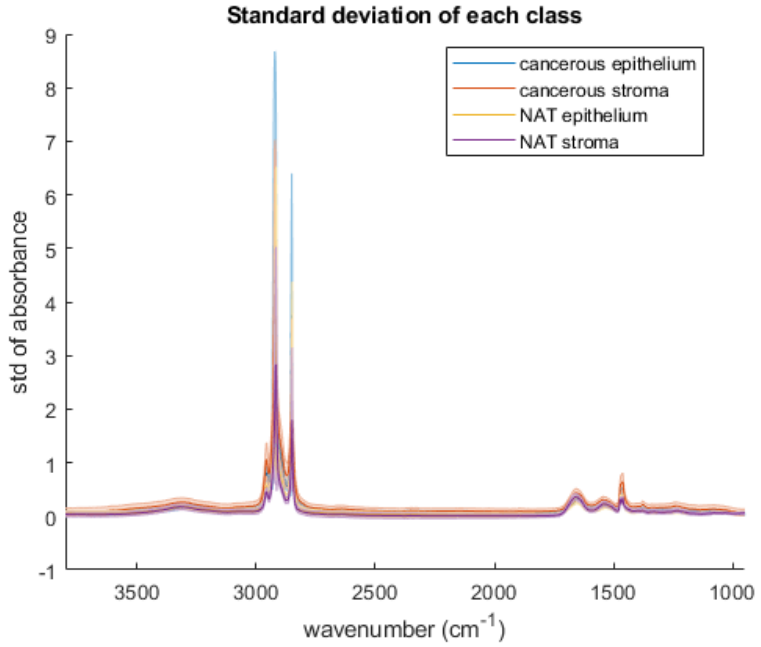

Standard deviation of NAT spectra
std
std of absorbance
wavenumber (cm-1)
3500
3000
2500
2000
1500
1000
-1
0
1
2
3
4
5
6



Pool data

```
% Create a ChiIRSpectralCollection ojbect to hold the pooled data
annotatedpixels = ChiIRSpectralCollection;

% Append each annotated region to this object
annotatedpixels.append(d8redspectra);
annotatedpixels.append(d8purplespectra);
annotatedpixels.append(m4greenspectra);
annotatedpixels.append(m4orangespectra);

% Generate a ChiClassMemberhip object containing labels for each spectrum
labels = ChiClassMembership('annotation', ...
                        'cancerous epithelium',d8redspectra.numspectra, ...
                        'cancerous stroma',d8purplespectra.numspectra, ...
                        'NAT epithelium',m4greenspectra.numspectra, ...
                        'NAT stroma',m4orangespectra.numspectra);

% Attach the labels to the data
annotatedpixels.classmembership = labels;

% Show plots if requested
if showplots
    % Mean spectra of each class
    annotatedpixels.plot('mean', 'byclass', 'title','Mean spectra of each class')

    % Standard deviation of each class
    annotatedpixels.plot('std', 'byclass', 'title','Standard deviation of each class')
end
```

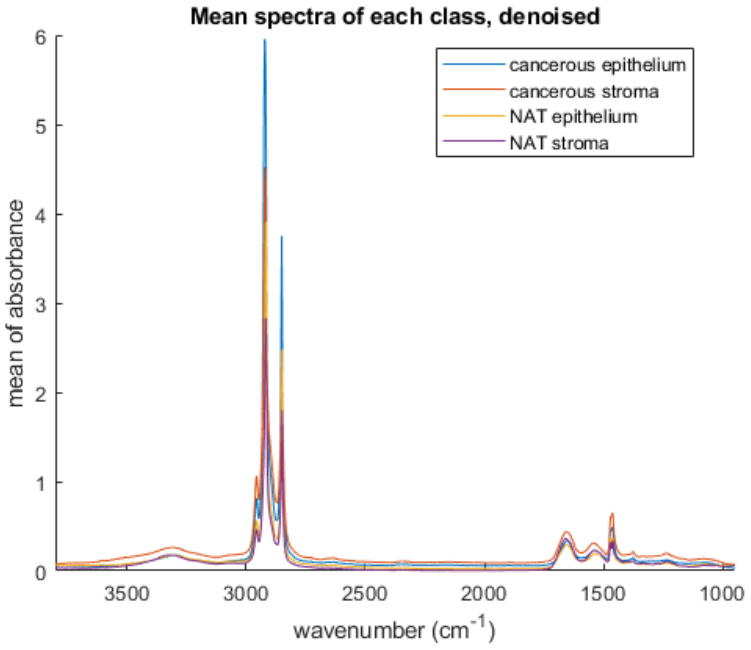


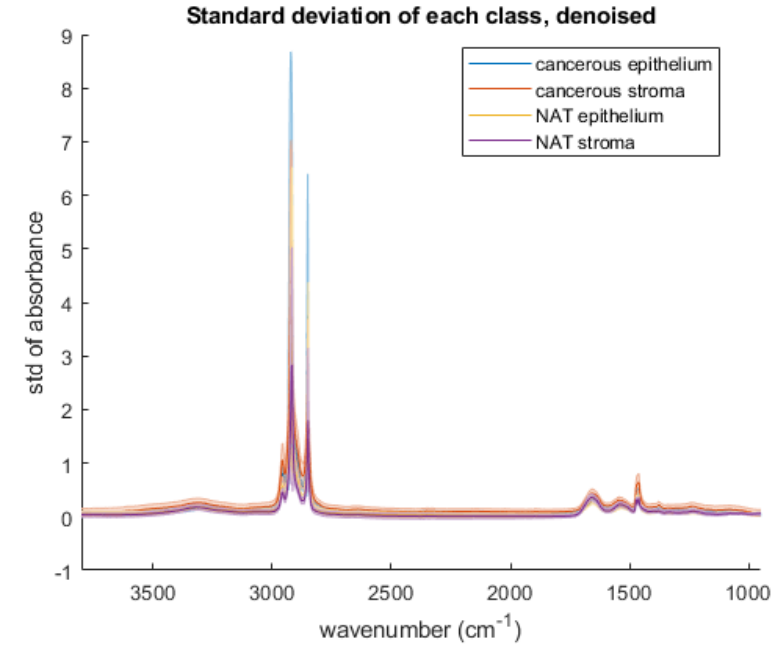


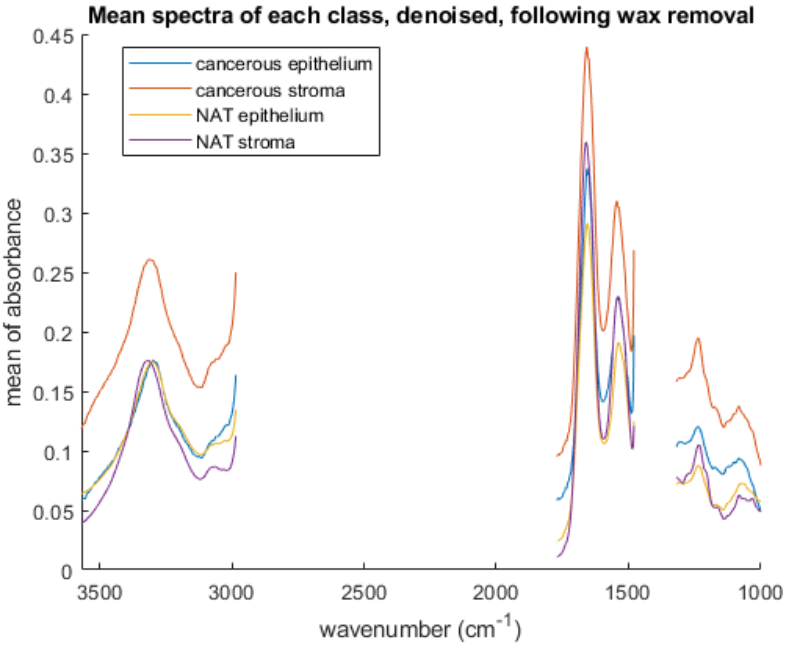
Pre-processing

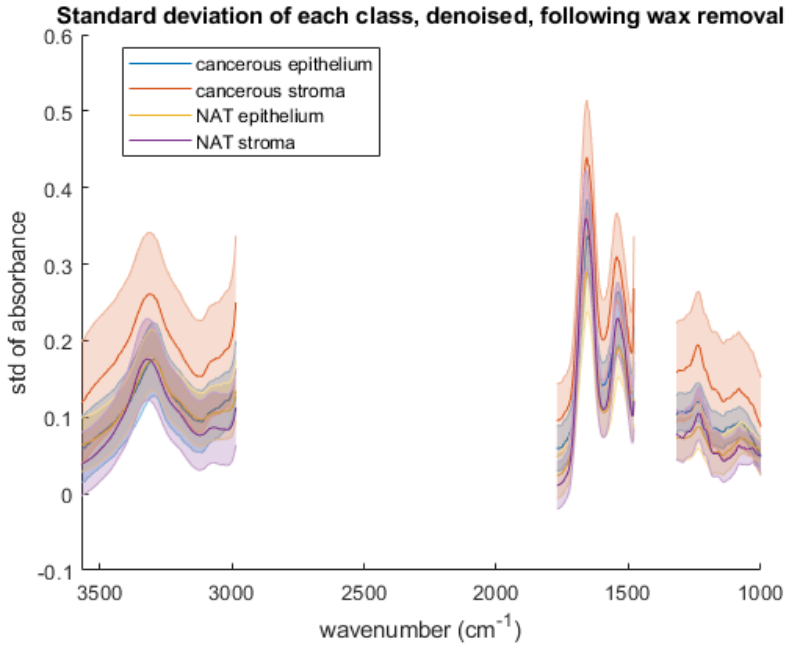
```
% Perform PCA denoising, retaining 80 principal components
annotatedpixels.denoise(80);

% Show plots if requested
if showplots
    % Mean spectra of each class, denoised
    annotatedpixels.plot('mean', 'byclass', 'title','Mean spectra of each class, denoised')

    % Standard deviation of each class, denoised
    annotatedpixels.plot('std', 'byclass', 'title','Standard deviation of each class, denoised')
end

% Remove spectral regions heavily influenced by the paraffin wax embedding
% material
annotatedpixels = annotatedpixels.keeprange(1000,1319, 1481,1769, 2986,3569);

% Show plots if requested
if showplots
    % Mean spectra of each class, denoised, following wax removal
    annotatedpixels.plot('mean', 'byclass', 'title','Mean spectra of each class, denoised, following wax removal')

    % Standard deviation of each class, denoised, following wax removal
    annotatedpixels.plot('std', 'byclass', 'title','Standard deviation of each class, denoised, following wax removal')
```

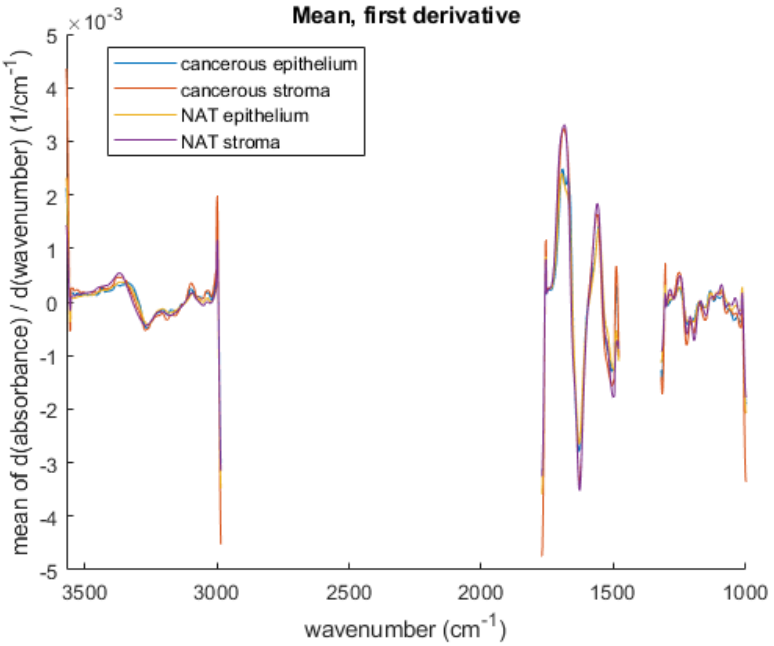
```
end

% Perform fourth-order polynomial Savitzky-Golay first derivative with
% window size of 19
annotatedpixels.firstderiv(19);

% Show plots if requested
if showplots
    % Mean spectra of first derivative of each class, denoised, following
    % wax removal
    annotatedpixels.plot('mean', 'byclass', 'title','Mean, first derivative')

    % Standard deviation of first derivative of  each class, denoised,
    % following wax removal
    annotatedpixels.plot('std', 'byclass', 'title','Standard deviation, first derivative')
end

% Remove regions of the spectra influenced by the ends of the spectral
% regions
annotatedpixels = annotatedpixels.keeprange(1019,1300, 1500,1750, 3005,3550);

% Show plots if requested
if showplots
    % Mean spectra of first derivative of each class, denoised, following
    % wax removal, with end regions removed
    annotatedpixels.plot('mean', 'byclass', 'title','Mean, first derivative, trimmed')

    % Standard deviation of first derivative of  each class, denoised,
    % following wax removal, with end regions removed
    annotatedpixels.plot('std', 'byclass', 'title','Standard deviation, first derivative, trimmed')
end
```

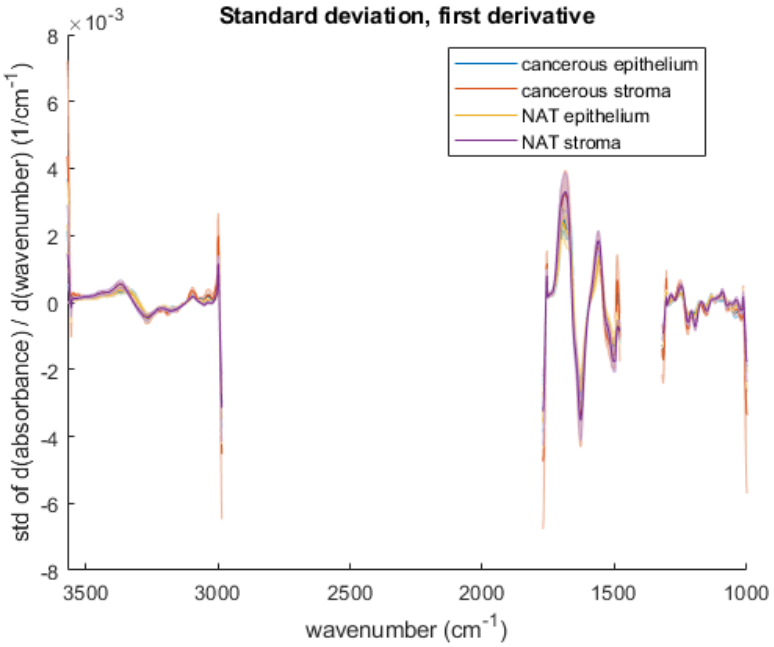


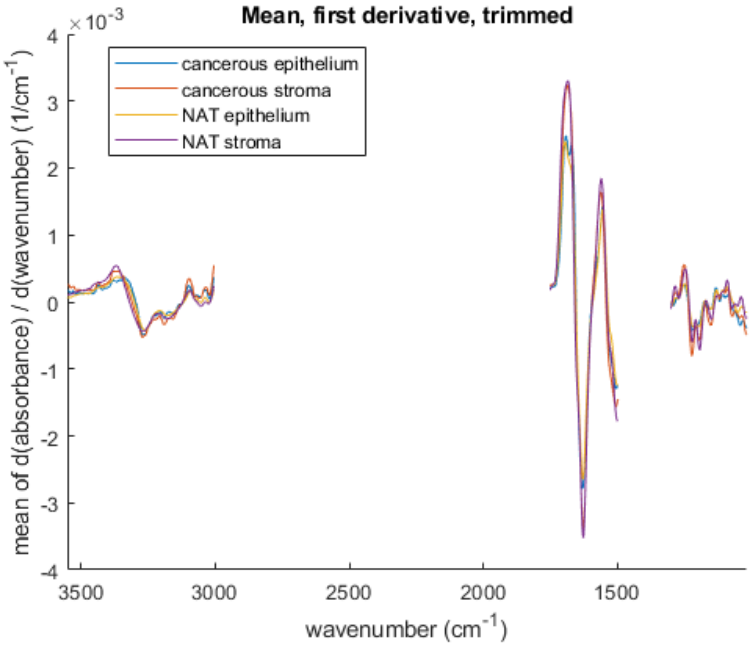

Standard deviation of each class, denoised
cancerous epithelium
cancerous stroma
NAT epithelium
NAT stroma
std of absorbance
wavenumber (cm-1)

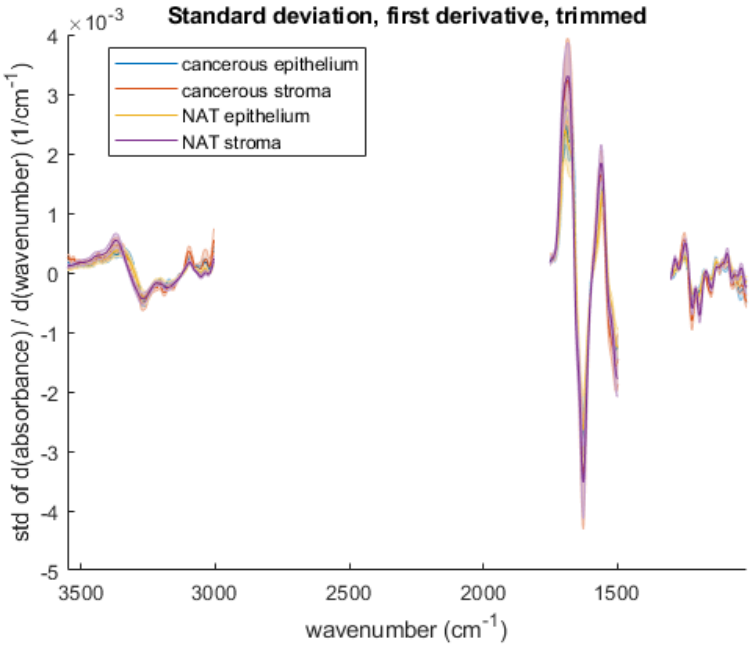

Mean spectra of each class, denoised, following wax removal
cancerous epithelium
cancerous stroma
NAT epithelium
NAT stroma
mean of absorbance
wavenumber (cm-1)


Standard deviation of each class, denoised, following wax removal
cancerous epithelium
cancerous stroma
NAT epithelium
NAT stroma
std of absorbance
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
3500
3000
2500
2000
1500
1000
wavenumber (cm-1)


Mean, first derivative
×10-3
cancerous epithelium
cancerous stroma
NAT epithelium
NAT stroma
mean of d(absorbance) / d(wavenumber) (1/cm-1)
5
4
3
2
1
0
-1
-2
-3
-4
-5
3500
3000
2500
2000
1500
1000
wavenumber (cm-1)


Standard deviation, first derivative
×10^-3
8
6
4
2
0
-2
-4
-6
-8
std of d(absorbance) / d(wavenumber) (1/cm^-1)
cancerous epithelium
cancerous stroma
NAT epithelium
NAT stroma
3500
3000
2500
2000
1500
1000
wavenumber (cm^-1)


Mean, first derivative, trimmed
×10^-3
4
3
2
1
0
-1
-2
-3
-4
mean of d(absorbance) / d(wavenumber) (1/cm^-1)
cancerous epithelium
cancerous stroma
NAT epithelium
NAT stroma
3500
3000
2500
2000
1500
wavenumber (cm^-1)



Published with MATLAB® R2019a