1 Investigate Datasets
   • Get Dataset Information
   • View Data

2 Pre-processing
   • Reproject the data to UTM
   • Subset the image to a region of interest
   • Create a composite image
   • Apply Lee filter
   • Convert to dB

3 Summary
Getting started

Open a terminal window and change to the directory the data is stored by typing the following:

```
cd ~/RSGISLibCourse/Data
```
To get information on the image type the following command:

```
gdalinfo -norat N06W053_JERS1_96_HH.kea
```
This should print the following:

Driver: KEA/KEA Image Format (.kea)
Files: N06W053_JERS1_96_HH.kea
Size is 4500, 4500
Coordinate System is:
GEOGCS["GCS_WGS_1984",
   DATUM["WGS_1984",
      SPHEROID["WGS_84",6378137,298.257223563]],
      PRIMEM["Greenwich",0],
      UNIT["Degree",0.017453292519943295]]
Origin = (-53.000000000000000,6.000000000000000)
Pixel Size = (0.000222222222222,-0.000222222222222)

Try for the PALSAR data.
Open PALSAR data from 2007 in TuiView:

tuiview N06W053_PALSAR_07_HH_utm.kea
TuiView

1) Manage layers
2) Linked window
3) Pan and Zoom

Query Tool
Local Stretch
Flicker between layers

Pete Bunting and Daniel Clewley
The Remote Sensing and GIS Software Library
The following pre-processing steps need to be performed:

1. Reproject data to UTM
2. Subset the image to a region of interest.
3. Create a composite image.
4. Apply a Lee filter to reduce speckle.
5. Calibrate data to dB.

These steps provide:

- A Lee-filtered stack with HH-data, in dB, for three years for visualisation and input to the segmentation.
- A subset image for each year, used to attribute the segments.
Reproject the JERS-1 scene to UTM by typing the following command:

```bash
gdalwarp -t_srs EPSG:32622 -of KEA -tr 25 25 \N06W053_JERS1_96_HH.kea N06W053_JERS1_96_HH_utm.kea
```

- Run `gdalinfo` again to confirm the projection is the same as the PALSAR data.
- Type `gdalwarp` to view other available options.
- Go to the `gdalwarp` page (http://www.gdal.org/gdalwarp.html) for more information on the options.
Image subset

Open the image and vector in TuiView
We will use the `subset` command within the `imageutils` module of RSGISLib to subset image to bounding box of Shapefile.

```python
# Import required RSGISLib modules
import rsgislib
from rsgislib import imageutils

# Create variables for input and output datasets
inputImage = 'N06W053_JERS1_96_HH_utm.kea'
inputVector = 'bounding_box_utm.shp'
outputImage = 'N06W053_JERS1_96_HH_utm_sub.kea'

# Subset image
imageutils subset(inputImage, inputVector, outputImage, 'KEA', rsgislib.TYPE_32UINT)
```
Subset image (multiple)

As there are a number of images to subset, it’s easier to search for all the files that need subsetting (using the `glob` module) and iterating through these.

```python
# Import glob module
import glob

# Get list of files matching pattern '*_utm.kea'
inputImageList = glob.glob("*_utm.kea")

# Iterate through files
for inputImage in inputImageList:
    print('Subsetting: ' + inputImage)

    # Set name for output image by replacing 'kea' with '_sub.kea'
    outputImage = inputImage.replace('.kea', '_sub.kea')
    print('Saving to: ' + outputImage)

Pete Bunting and Daniel Clewley
The Remote Sensing and GIS Software Library
Open subset script

- Copy (cp) the 1_subset_multiple.py script from the ‘scripts’ folder to your data folder:
  ```shell
  cp ../Scripts/1_subset_multiple.py ./
  ```
- Open the file and go through the contents, looking up imageutils.subset on www.rsgislib.org and the glob module in the Python standard library documentation.
- You can also get information about functions by opening another terminal window starting ipython and typing:
  ```python
  from rsgislib import imageutils
  help(imageutils.subset)
  ```
Run subset script

- Run using:
  python 1_subset_multiple.py
Image subset

Compare the subset to the original in TuiView
False colour composites provide a good way of visualising data from multiple years. To create a composite image, from all dates use the `stackImageBands` command in RSGISLib:

```python
imageList = ['N06W053_JERS1_96_HH_utm_sub.kea',
             'N06W053_PALSAR_07_HH_utm_sub.kea',
             'N06W053_PALSAR_10_HH_utm_sub.kea']

bandNamesList = ['1996', '2007', '2010']

outputImage = 'N06W053_96-10_stack.kea'

imageutils.stackImageBands(imageList, bandNamesList, outputImage, None, 0, 'KEA', rsgislib.TYPE_32FLOAT)
```
Run composite script

- Copy (cp) the 2_stack_bands.py script from the ‘scripts’ folder to your data folder:
  
cp ../Scripts/2_stack_bands.py ./

- Open the file and go through the contents.

- Run using:
  
  python 2_stack_bands.py
View Composite Image

Open the composite image in TuiView
Apply Lee Filter

To reduce speckle prior to segmenting the image, apply a Lee filter, available through the imagefilter module.

```python
inputImage = 'N06W053_96-10_stack.kea'
outputImageBase = 'N06W053_96-10_stack'

filters = []
filters.append(imagefilter.FilterParameters(filterType = 'Lee',
                                           fileEnding = 'lee', size=3, nLooks=3))
imagefilter.applyfilters(inputImage, outputImageBase,
                         filters, 'KEA', 'kea', rsgislib.TYPE_32FLOAT)
```
Run filtering script

- Copy (cp) the 3_lee_filter.py script from the 'scripts' folder to your data folder:
  ```bash
cp ../Scripts/3_lee_filter.py ./
  ```
- Open the file and go through the contents.
- Run using:
  ```bash
  python 3_lee_filter.py
  ```
View Filtered Data

Compare to the original in TuiView
The data are supplied as digital numbers (DN), to convert to $\sigma^0$ (in dB) the following equation is used:

$$\sigma^0 = 10 \log_{10}(DN^2) + C$$  \hspace{1cm} (1)

Where C is -84.66 for JERS-1 data and -83.0 for PALSAR data.
Calibrate Data - Band Maths

Apply the calibration using the \texttt{bandmath} function from the \texttt{imagecalc} module.

```python
inStack = 'N06W053_96-10_stack_lee.kea'
jers1Temp = 'jers1_96_temp_dB.kea' format = 'KEA'

dn > 0?20*log10(dn)-84.66:0'

bandDefns = []
bandDefns.append(BandDefn('dn', inStack, 1))
imagecalc.bandMath(jers1Temp, jers1Cal, 'KEA',
rsgislib.TYPE_32FLOAT, bandDefns)
```

Need to apply to each band individual, due to different calibration requirements for JERS-1 and PALSAR and re-stack.
Run filtering script

- Copy (cp) the 4_calibrate_data.py script from the ‘scripts’ folder to your data folder:
  ```
  cp ../Scripts/4_calibrate_data.py ./
  ```
- Open the file and go through the contents.
- Run using:
  ```
  python 4_calibrate_data.py
  ```
View Calibrated Data

Now the data are calibrated, open them in TuiView and use the Query tool (+) to get the value of each pixel in all bands and display a plot of the trend in $\sigma^0$ over time.
Use `gdal_translate` to create a KMZ file of the composite.

```
gdal_translate -of KMLSUPEROVERLAY -scale \nN06W053_96-10_stack_lee_dB.kea \nN06W053_96-10_stack_lee_dB.kmz
```

Where the `-scale` flag is used to scale the values from 0 - 255.
Outline

Investigate Datasets
Pre-processing
Summary

Reproject the data to UTM
Subset the image to a region of interest
Create a composite image
Apply Lee filter
Convert to dB

View in GoogleEarth

Note: More advanced image enhancement is available in RSGISLib

Pete Bunting and Daniel Clewley
The Remote Sensing and GIS Software Library
So far we have covered:

- Some of the utility programs included with GDAL
- The RSGISLib Python interface and a selection of commands
- Data visualisation using TuiView
Tomorrow we will cover:

- Image segmentation
- Classification
- Change detection
- More advanced Python