

Reading and writing files

**V. Orgogozo
Jan 6 2012**

Our goal today : transform a text file into a Google Earth .kml file

```
Dive Date Lat Lon Depth Notes
Tiburon 596 19-Jul-03 36 36.12 N 122 22.48 W 1190 holotype
JSL II 1411 16-Sep-86 39 56.4 N 70 14.3 W 518 paratype
JSL II 930 18-Aug-84 40 05.03 N 69 03.01 W 686 Youngbluth (1989)
Ventana 1575 11-Mar-99 36 42.24 N 122 02.52 W 767
Ventana 1777 16-Jun-00 36 42.60 N 122 02.70 W 934
Ventana 2243 9-Sep-02 36 42.48 N 122 03.84 W 1001
Tiburon 515 24-Nov-02 36 42.00 N 122 01.98 W 1156
Tiburon 531 13-Mar-03 24 19.02 N 109 12.18 W 1144
Tiburon 547 31-Mar-03 24 14.04 N 109 40.02 W 1126
JSL II 3457 26-Sep-03 40 17.77 N 68 06.68 W 862 Francesc Pages (pers.comm)
```

Marrus_claudanielis.txt



Marrus_claudanielis.kml

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
  <Document>

    <Placemark>
      <name>Tiburon 596</name>
      <description>Tiburon 596      19-Jul-03      36 36.12 N 122 22.48 W 1190 holotype</description>
      <Point>
        <altitudeMode>absolute</altitudeMode>
        <coordinates>-122.374667, 36.602000, -1190</coordinates>
      </Point>
    </Placemark>
    <Placemark>
      <name>JSL II 1411</name>
      <description>JSL II 1411      16-Sep-86      39 56.4 N 70 14.3 W 518 paratype</description>
      <Point>
        <altitudeMode>absolute</altitudeMode>
        <coordinates>-70.238333, 39.940000, -518</coordinates>
      </Point>
    </Placemark>
    <Placemark>

    (...)

  </Placemark>

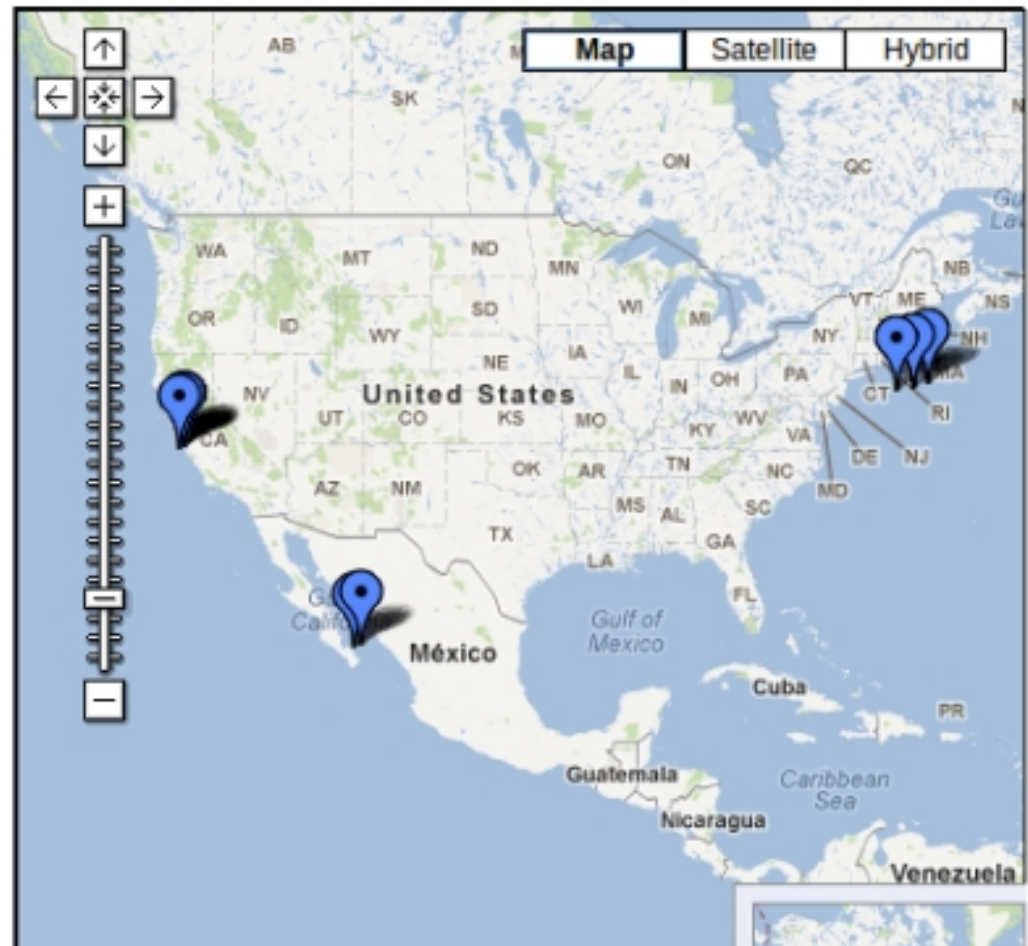
</Document>
</kml>
```

A .kml file can be visualized with Google Earth or Google Map

Type your KML in here

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com
/kml/2.2">
<Document>
<Placemark>
<name>Tiburon 596</name>
<description>Tiburon 596
19-Jul-03 36 36.12 N 122
22.48 W 1190 holotype</description>
<Point>
<altitudeMode>absolute</altitudeMode>
<coordinates>-122.374667, 36.602000,
-1190</coordinates>
</Point>
</Placemark>
<Placemark>
<name>JSL II 1411</name>
<description>JSL II 1411
16-Sep-86 39 56.4 N 70 14.3
W 518 paratype</description>
```

Show it on the map!



<http://display-kml.appspot.com/>

Always think first about the strategy

to save time and anticipate complex steps

Certain issues might have a big effect on the way the program should be written

(1) check the input file and the output file

Marrus_claudanielis.txt

Dive	Date	Lat	Lon	Depth	Notes
Tiburon 596	19-Jul-03	36 36.12 N	122 22.48 W	1190	holotype
JSL II 1411	16-Sep-86	39 56.4 N	70 14.3 W	518	paratype
JSL II 930	18-Aug-84	40 05.03 N	69 03.01 W	686	Youngbluth (1989)
Ventana 1575	11-Mar-99	36 42.24 N	122 02.52 W	767	
Ventana 1777	16-Jun-00	36 42.60 N	122 02.70 W	934	
Ventana 2243	9-Sep-02	36 42.48 N	122 03.84 W	1001	
Tiburon 515	24-Nov-02	36 42			
Tiburon 531	13-Mar-03	24 19			
Tiburon 547	31-Mar-03	24 14			
JSL II 3457	26-Sep-03	40 17			

Marrus_claudanielis.kml

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
<Document>

<Placemark>
  <name>Tiburon 596</name>
  <description>Tiburon 596 19-Jul-03 36 36.12 N 122 22.48 W 1190 holotype</description>
  <Point>
    <altitudeMode>absolute</altitudeMode>
    <coordinates>-122.374667, 36.602000, -1190</coordinates>
  </Point>
</Placemark>
<Placemark>
  <name>JSL II 1411</name>
  <description>JSL II 1411 16-Sep-86 39 56.4 N 70 14.3 W 518 paratype</description>
  <Point>
    <altitudeMode>absolute</altitudeMode>
    <coordinates>-70.238333, 39.940000, -518</coordinates>
  </Point>
</Placemark>
<Placemark>

(...)

</Placemark>

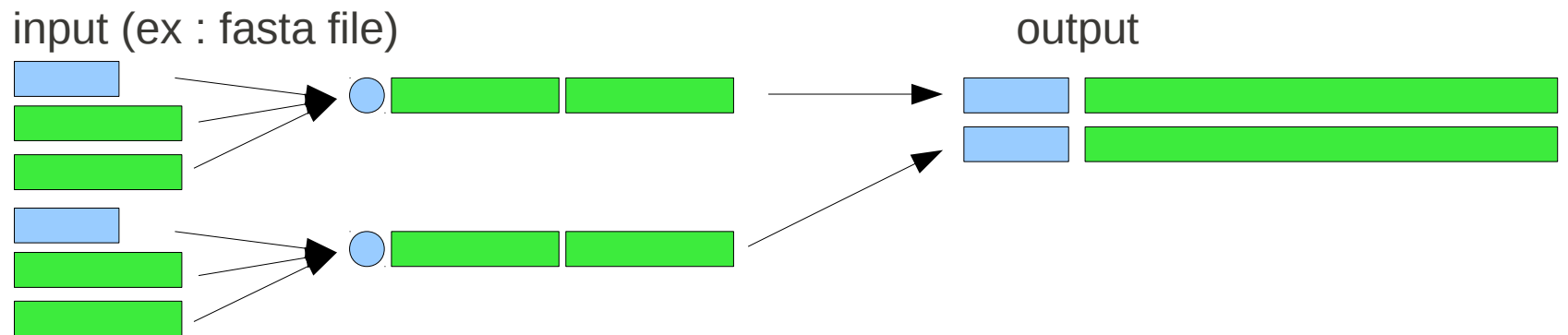
</Document>
</kml>
```

(2) which method should we use to transform the input file into the output file?

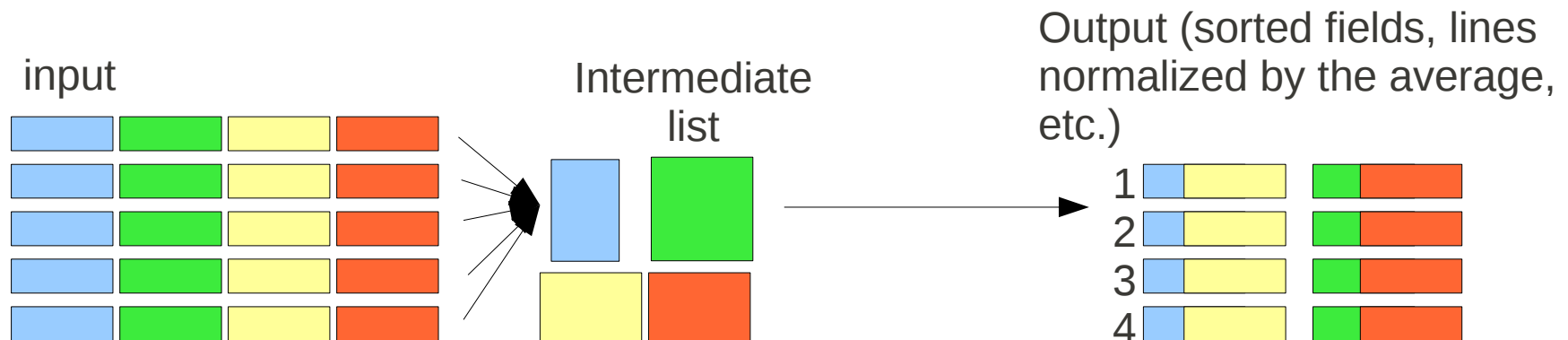
(a) *each line can be processed in the order it occurs*



(b) *several lines must be combined into an output line*



(c) *all the lines must be processed before the output file is created*



Open latlon_1.py

```
#!/usr/bin/env python

# Set the input file name
# (The program must be run from within the directory
# that contains this data file)
InFileName = 'Marrus_claudanielis.txt'

# Open the input file for reading
InFile = open(InFileName, 'r')

# Initialize the counter used to keep track of line numbers
LineNumber = 0

# Loop through each line in the file
for Line in InFile:
    # Remove the line-ending characters
    Line=Line.strip('\n')
    # Print the line
    print LineNumber,":", Line

    # Index the counter used to keep track of line numbers
    LineNumber = LineNumber + 1

# After the loop is completed, close the file
InFile.close()
```

Function open
(file name, file mode)
'r' = read mode

The **InFileName** and the
InFile variables are different

*To create a list variable
containing all the lines of the
file:*

```
FileList = inFile.readlines()
```

Method `.strip()`
removes line endings (space,
tab, end of line)

Open latlon_1.py

```
#!/usr/bin/env python

# Set the input file name
# (The program must be run from within the directory
# that contains this data file)
InFileName = 'Marrus_claudanielis.txt'

# Open the input file for reading
InFile = open(InFileName, 'r')

# Initialize the counter used to keep track of line numbers
LineNumber = 0

# Loop through each line in the file
for Line in InFile:
    # Remove the line-ending characters
    Line=Line.strip('\n')
    # Print the line
    print LineNumber,":", Line

    # Index the counter used to keep track of line numbers
    LineNumber = LineNumber + 1

# After the loop is completed, close the file
InFile.close()
```

Better to start numbering with 0

Always close the file when done

Function open
(file name, file mode)
'r' = read mode

The **InFileName** and the **InFile** variables are different

To create a list variable containing all the lines of the file:
FileList = inFile.readlines()

Method .strip()
removes line endings (space, tab, end of line)

make the file executable, execute it in the examples folder

```

virginie@Darwin:~/Documents/BioInfo-cours/pcfb/examples$ ./latlon_1.py
0 : Dive   Date Lat  Lon  Depth  Notes
1 : Tiburon 596  19-Jul-03  36 36.12 N 122 22.48 W  1190 holotype
2 : JSL II 1411  16-Sep-86  39 56.4 N  70 14.3 W  518 paratype
3 : JSL II 930   18-Aug-84  40 05.03 N 69 03.01 W  686 Youngbluth (1989)
4 : Ventana 1575   11-Mar-99  36 42.24 N 122 02.52 W  767
5 : Ventana 1777   16-Jun-00  36 42.60 N 122 02.70 W  934
6 : Ventana 2243   9-Sep-02  36 42.48 N 122 03.84 W  1001
7 : Tiburon 515   24-Nov-02  36 42.00 N 122 01.98 W  1156
8 : Tiburon 531   13-Mar-03  24 19.02 N 109 12.18 W  1144
9 : Tiburon 547   31-Mar-03  24 14.04 N 109 40.02 W  1126
10 : JSL II 3457  26-Sep-03  40 17.77 N 68 06.68 W  862 Francesc Pages
(pers.comm)
virginie@Darwin:~/Documents/BioInfo-cours/pcfb/examples$

```

**Modify the script so
that the header line is
not printed**

HINT :
Use if ... :
within the loop

```

#!/usr/bin/env python
(...)

# Initialize the counter used to keep track of line numbers
LineNumber = 0

# Loop through each line in the file
for Line in InFile:
    # Remove the line-ending characters
    Line=Line.strip('\n')
    # Print the line
    print LineNumber,":", Line

    # Index the counter used to keep track of line numbers
    LineNumber = LineNumber + 1

# After the loop is completed, close the file
InFile.close()

```


(3) split the line into data fields

Method `.split()`

Produces a list of strings = the values occurring between the delimiters, the delimiters are thrown away

`.split()` considers space and tab as delimiters

`.split('\t')` considers only tab as delimiters

Open `latlon_2.py`

What happens if you use `Line.split('\t')` instead of `Line.split(' ')` ?

```
#!/usr/bin/env python
(...)

# Initialize the counter used to keep track of line numbers
LineNumber = 0

# Loop through each line in the file
for Line in InFile:
    if LineNumber > 0:
        # Remove the line ending characters
        Line=Line.strip('\n')

        # Separate the line into a list of its tab-delimited components
        LineList=Line.split('\t')

        # Print the line
        # print LineNumber,":", LineList
        # print LineList[4], LineList[2], LineList[3]
        print "Depth: %s\tLat: %s\t Lon:%s" % (LineList[4], LineList[2],
LineList[3])

        # Index the counter used to keep track of line numbers
        LineNumber = LineNumber + 1

(...)
```

(4) write the new output file

Open latlon_3.py

Function open
(file name, file mode)
'w' = write mode
'a' = append mode

Be careful of not
deleting existing files

>> versus >

```
#!/usr/bin/env python  
(...)
```

```
LineNumber = 0
```

```
1 # Open the output file for writing -Do this *before* the loop, not inside it  
  OutFileName=InFileName + ".kml"
```

```
2 OutFile=open(OutFileName,'w') # You can append instead with 'a'
```

```
# Loop through each line in the file  
for Line in InFile:
```

```
    # Skip the header, line # 0
```

```
    if LineNumber > 0:
```

```
        # Remove the line ending characters
```

```
        Line=Line.strip('\n')
```

```
        ElementList=Line.split('\t')
```

```
        # Use the % operator to generate a string
```

```
        # We can use this for output both to the screen and to a file
```

```
        OutputString = "Depth: %s\tLat: %s\t Lon:%s" % \  
            (ElementList[4], ElementList[2], ElementList[3])
```

```
        # Can still print to the screen then write to a file
```

```
        print OutputString
```

```
3 # Unlike print statements, .write needs a linefeed  
  OutFile.write(OutputString+"\n")
```

```
# Index the counter used to keep track of line numbers
```

```
LineNumber = LineNumber + 1
```

```
# After the loop is completed, close the files
```

```
InFile.close()
```

```
4 OutFile.close()
```