Expect script for automating Aspera uploads to the EBI ENA

So I've been tinkering with a couple of expect scripts to automate the rather tedious task of uploading raw data files to the EBI ENA. Usually, this would require either using a FTP connection, or using the Webin submission route. However, we use our MISO LIMS to generate the required schema XMLs, and use the Aspera command line tool ascp (included in the plugin package) to upload the files themselves, so a way to do this in a batch fashion would be nice.

Automated script

aspera_expect

```
#!/usr/bin/expect

## set these to your dropbox id and password
set dropbox "era-drop-XX"
set pass "XXXXXXXXXXXXXX"

set fname [lindex $argv 0]
set files [open $fname]
set subs [read $files]

set run "[lindex $argv 1]"
set project "[lindex $argv 2]"

set direxist 0
set ident ""

set timeout -1

foreach line [split $subs \n] {
  if { "" != $line } {
    set seqfile [exec basename $line]
    set lst [split $line "/"]
    foreach p $lst {
      if {[regexp {[0-9][0-9][0-9]} $p]} {
        set ident $p
        puts $ident
      }
    }

    if { "" != $ident } {
      spawn ./ascp -QT -l80M -d $line $dropbox@webin.ebi.ac.uk:/$project/$run/$ident/.
    } else {
      spawn ./ascp -QT -l80M -d $line $dropbox@webin.ebi.ac.uk:/$project/$run/.
    }
    expect "$dropbox@webin.ebi.ac.uk\'s password:"
    send "$pass\r"
    expect eof
  }
}
```

This script requires a few elements (as well as expect being installed of course!):

- The ascp command line client, copied or symlinked to the same folder as the following script
- Your ENA dropbox username and password
- A simple text file containing the absolute pathnames of all files that you wish to upload
- The run folder name
- A project identifier which will become the parent folder in the ENA dropbox

Example 1

So, let's consider an input file "files_to_upload" that looks like this:

```
/path/to/runs/130101_SN001_001_ABLAHBLAH/Data/Intensities/BaseCalls/analysis/Project_X/Sample_Y
/Library_Z_NoIndex_L001_R1.fastq.gz
/path/to/runs/130101_SN001_001_ABLAHBLAH/Data/Intensities/BaseCalls/analysis/Project_X/Sample_Y
/Library_Z_NoIndex_L001_R2.fastq.gz
```

We have a paired end run with a single non-multiplexed library as resolved by some primary analysis, such as CASAVA, that you have carried out (hence the "analysis" folder). We can run the script like so:

```
[davey@tgac]$ ./aspera_expect files_to_upload 130101_SN001_001_ABLAHBLAH Project_X
```

All being well, you should see the terminal output that resembles the following:

```
spawn ./ascp -QT -l80M -d /path/to/runs/130101_SN001_001_ABLAHBLAH/Data/Intensities/BaseCalls/analysis/Project_X
/Sample_Y/Library_Z_NoIndex_L001_R1.fastq.gz era-drop-XX@fasp.era.ebi.ac.uk:/Project_X
/130101_SN001_001_ABLAHBLAH/.
era-drop-XX@fasp.era.ebi.ac.uk's password:
Library_Z_NoIndex_L001_R1.fastq.
gz
100% 1845MB 78.1Mb/s    03:18
Completed: 1889367K bytes transferred in 199 seconds
 (77465K bits/sec), in 1 file.
```

This will create the following folder structure on the remote ENA dropbox:

```
/Project_X/130101_SN001_001_ABLAHBLAH/Library_Z_NoIndex_L001_R1.fastq
/Project_X/130101_SN001_001_ABLAHBLAH/Library_Z_NoIndex_L001_R2.fastq
```

Example 2

This example is a little more complex, in that we have some demultiplexing output in file "more_files_to_upload":

more_files_to_upload

```
/path/to/runs/130101_SN002_002_BBLAHBLAH/Data/Intensities/BaseCalls/Qseqs/Demultiplexed/004/Fastq
/s_4_1_sequence.txt.gz
/path/to/runs/130101_SN002_002_BBLAHBLAH/Data/Intensities/BaseCalls/Qseqs/Demultiplexed/004/Fastq
/s_4_2_sequence.txt.gz
/path/to/runs/130101_SN002_002_BBLAHBLAH/Data/Intensities/BaseCalls/Qseqs/Demultiplexed/005/Fastq
/s_4_1_sequence.txt.gz
/path/to/runs/130101_SN002_002_BBLAHBLAH/Data/Intensities/BaseCalls/Qseqs/Demultiplexed/005/Fastq
/s_4_2_sequence.txt.gz
```

This represents the "old" way of CASAVA demultiplexing, i.e. we have raw files with the same name, but are distinguished by the parent folder name. Obviously, we need to preserve this folder structure when uploading to prevent overwrites. The script does this by attempting to find the numerical demultiplexed folder in the path (line 24 in the expect script code), and adding this to the upload path:

```
[davey@tgac]$ ./aspera_expect more_files_to_upload 130101_SN002_002_BBLAHBLAH Project_P
```

This will create the following folder structure on the remote ENA dropbox:

```
/Project_P/130101_SN002_002_BBLAHBLAH/004/s_4_1_sequence.txt.gz
/Project_P/130101_SN002_002_BBLAHBLAH/004/s_4_2_sequence.txt.gz
/Project_P/130101_SN002_002_BBLAHBLAH/005/s_4_1_sequence.txt.gz
/Project_P/130101_SN002_002_BBLAHBLAH/005/s_4_2_sequence.txt.gz
```

Notes

Supported sequencing platforms

This script can be used for any data file upload, but the examples above are obviously for Illumina.

Network bandwidth

You can tune the speed of the upload by modifying the "-l" flag in the ascp command:

```
-l80M
```

We have it set to 80MB/s, but you may want to increase/decrease this depending on your available bandwidth.

Demultiplexing folder detection

Parent folder detection can probably be improved over the simple regex:

```
regexp {[0-9][0-9][0-9]
```

I'd appreciate comments on this!