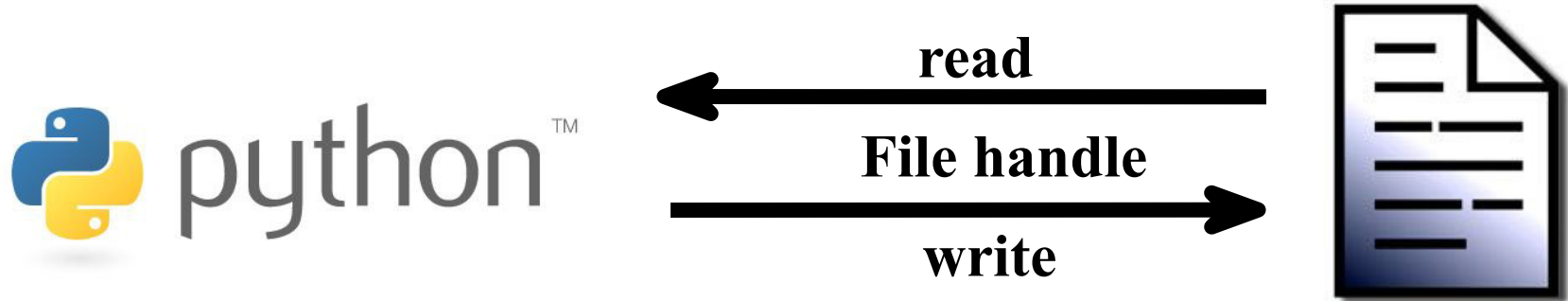


File handling

Concept of file handling

- “connection” from Python to file
- Read = Transfer of data **from** file
- Write = Transfer of data **into** file



Read a file (parsing)

“connection” from Python to file File in working directory

1 `f = open("test.txt", "r") # "r" ist default`
`lines = f.readlines()`
3 `f.close()`

Close “connection”

Function for reading all lines

oder

2 `with open("test.txt", "r") as f: # "r" ist default`
`lines = f.readlines()`

“connection” from Python to file

Reading a file (big data)

```
1 with open( "text.txt", "r" ) as f: #"r" is default
2     line = f.readline() #reads next line
3     while line: #until end of file is reached
4         print(line)
5         line = f.readline() #reads next line
```

- Advantage: only one line is read and processed at a time
- NGS data (e.g. FASTQ/SAM/BAM/VCF) are usually several GB in size => RAM limitations
- Very long sequence (e.g. genome sequences) in FASTA might be too large for available RAM


Analyze file - example

- How many lines are in AtCol0_Exons.fasta? (large file!)
- Under UNIX: `head <FILENAME>`

```
>AT1G01010.1|exon-1 | 1-283 | chr1:3631-3913 FORWARD LENGTH=283
AAATTATTAGATATACCAAACCAGAGAAAACAAATACATAATCGGAGAAATACAGATTACAGAGAGCGAGAGAGATCGAC
GGCGAAGCTCTTTACCCGGAAACCATTGAAATCGGACGGTTTAGTGAAAATGGAGGATCAAGTTGGGTTTGGGTTCCGTC
CGAACGACGAGGAGCTCGTTGGTCACTATCTCCGTAACAAAATCGAAGGAAACACTAGCCGCGACGTTGAAGTAGCCATC
AGCGAGGTCAACATCTGTAGCTACGATCCTTGGAAC TTGCGCT
>AT1G01010.1|exon-2 | 366-646 | chr1:3996-4276 FORWARD LENGTH=281
TCCAGTCAAAGTACAAATCGAGAGATGCTATGTGGTACTTCTTCTCTCGTAGAGAAAACAACAAAGGGAATCGACAGAGC
AGGACAACGGTTTCTGGTAAATGGAAGCTTACCCGAGAATCTGTTGAGGTCAAGGACCAGTGGGGATTTTGTAGTGAGGC
CTTTCGTGGTAAGATTGGTCATAAAAGGGTTTTTGGTGTTCTCGATGGAAGATACCCTGACAAAACCAAATCTGATTGGC
TTATCCACGAGTTCCACTACGACCTCTTACCAGAACATCAG
>AT1G01010.1|exon-3 | 856-975 | chr1:4486-4605 FORWARD LENGTH=120
AGGACATATGTCATCTGCAGACTTGAGTACAAGGGTGATGATGCGGACATTCTATCTGCTTATGCAATAGATCCCACTCC
CGCTTTTGTCCCCAATATGACTAGTAGTGCAGGTTCTGTG
```

(multiple) FASTA

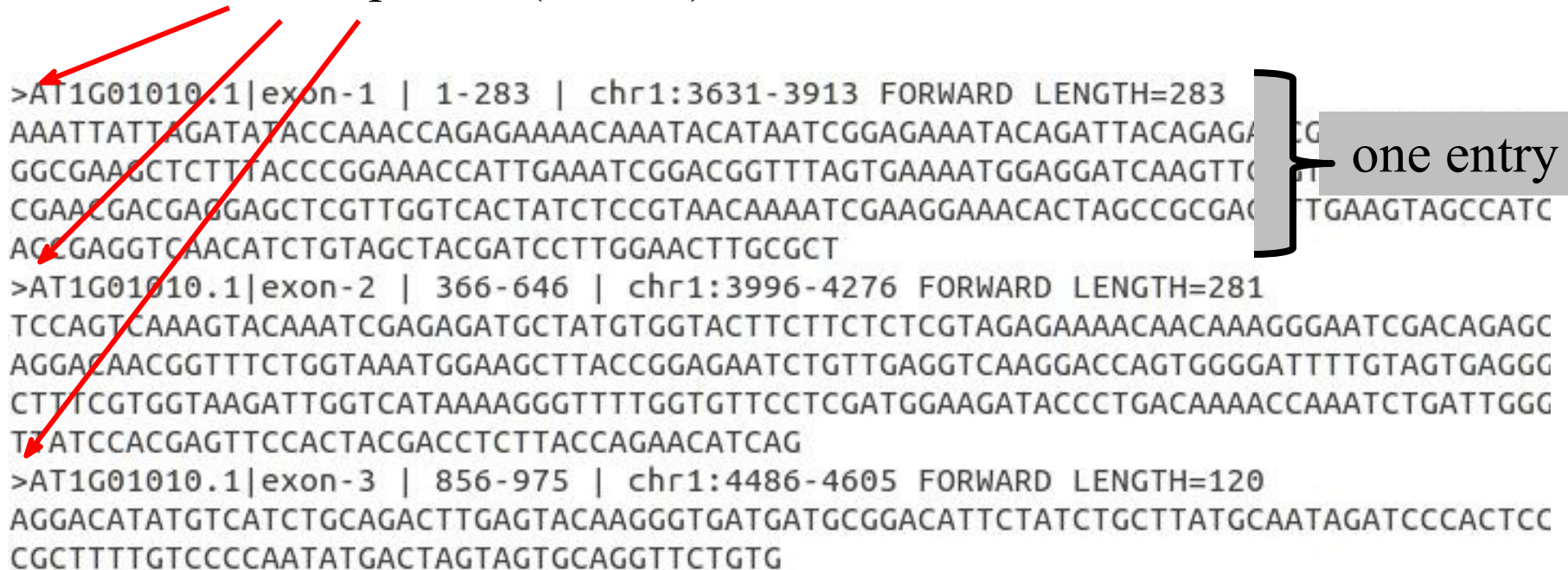
Name of sequence (header): line starts with ‘>’



```
>AT1G01010.1|exon-1 | 1-283 | chr1:3631-3913 FORWARD LENGTH=283
AAATTATTAGATATACCAAACCAGAGAAAACAAATACATAATCGGAGAAATACAGATTACAGAGAGCGAGAGAGATCGAC
GGCGAAGCTCTTTACCCGGAAACCATTGAAATCGGACGGTTTAGTGAAAATGGAGGATCAAGTTGGGTTTGGGTTCCGTC
CGAAACGACGAGGAGCTCGTTGGTCACTATCTCCGTAACAAAATCGAAGGAAACACTAGCCGCGACGTTGAAGTAGCCATC
ACGAGAGGTCAACATCTGTAGCTACGATCCTTGGAACCTTGCGCT
>AT1G01010.1|exon-2 | 366-646 | chr1:3996-4276 FORWARD LENGTH=281
TCCAGTCAAAGTACAAATCGAGAGATGCTATGTGGTACTTCTTCTCTCGTAGAGAAAACAACAAAGGGAATCGACAGAGC
AGGACAACGGTTTCTGGTAAATGGAAGCTTACCCGGAGAATCTGTTGAGGTCAAGGACCAGTGGGGATTTTGTAGTGAGGC
CTTTCGTGGTAAGATTGGTCATAAAAGGGTTTTGGTGTTCTCTCGATGGAAGATACCCTGACAAAACCAAATCTGATTGGC
TATCCACGAGTTCCACTACGACCTCTTACCAGAACATCAG
>AT1G01010.1|exon-3 | 856-975 | chr1:4486-4605 FORWARD LENGTH=120
AGGACATATGTCATCTGCAGACTTGAGTACAAGGGTGATGATGCGGACATTCTATCTGCTTATGCAATAGATCCCACTCC
CGCTTTTGTCCCCAATATGACTAGTAGTGACAGTTCTGTG
```


(multiple) FASTA

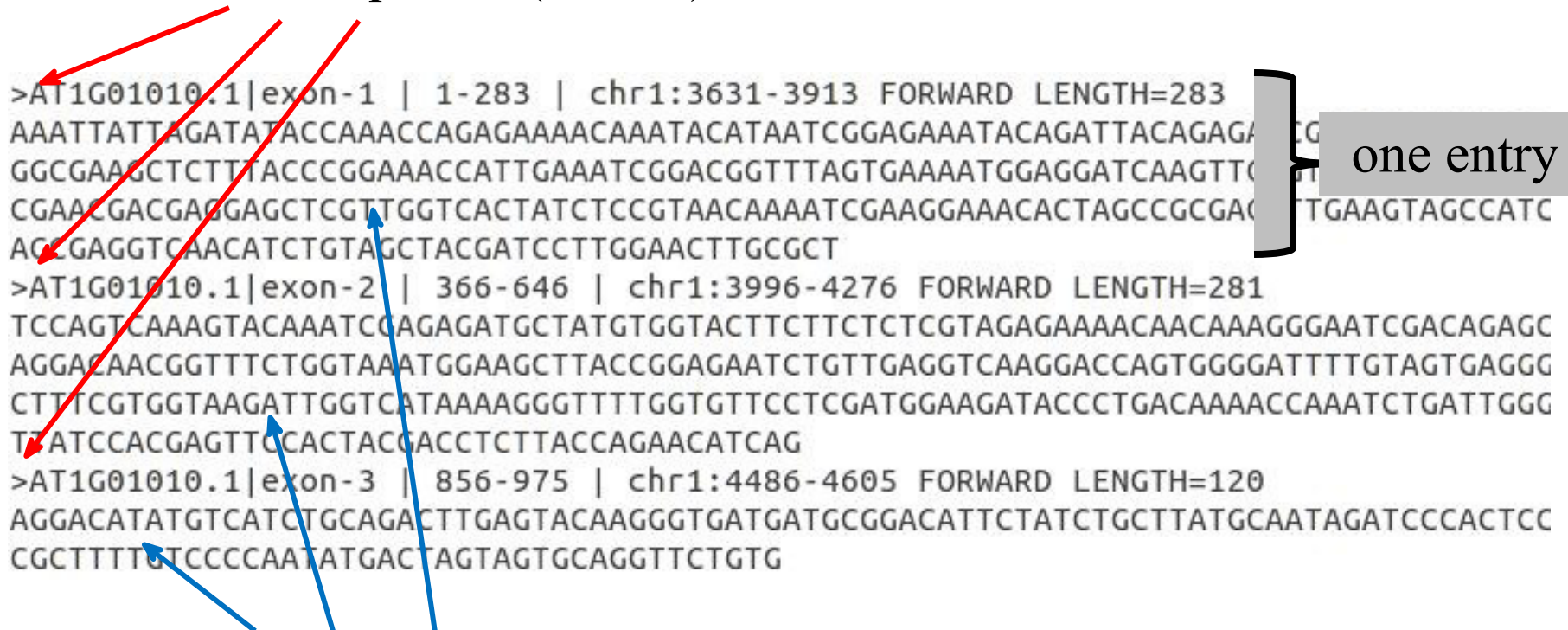
Name of sequence (header): line starts with ‘>’



```
>AT1G01010.1|exon-1 | 1-283 | chr1:3631-3913 FORWARD LENGTH=283
AAATTATTAGATATACCAAACCAGAGAAAACAAATACATAATCGGAGAAATACAGATTACAGAGA
GGCGAAGCTCTTTACCCGGAAACCATTGAAATCGGACGGTTTAGTGAAAATGGAGGATCAAGTTC
CGAAGGACGAGGAGCTCGTTGGTCACTATCTCCGTAACAAAATCGAAGGAAACACTAGCCGCGAG
ACGAGAGGTCAACATCTGTAGCTACGATCCTTGGAACCTTGCGCT
>AT1G01010.1|exon-2 | 366-646 | chr1:3996-4276 FORWARD LENGTH=281
TCCAGTCAAAGTACAAATCGAGAGATGCTATGTGGTACTTCTTCTCTCGTAGAGAAAACAACAAAGGGAATCGACAGAGC
AGGACAACGGTTTCTGGTAAATGGAAGCTTACCCGAGAATCTGTTGAGGTCAAGGACCAGTGGGGATTTTGTAGTGAGGC
CTTTCGTGGTAAGATTGGTCATAAAAGGGTTTTGGTGTTCTCTCGATGGAAGATACCCTGACAAAACCAAATCTGATTGGC
TATCCACGAGTTCCACTACGACCTCTTACCAGAACATCAG
>AT1G01010.1|exon-3 | 856-975 | chr1:4486-4605 FORWARD LENGTH=120
AGGACATATGTCATCTGCAGACTTGAGTACAAGGGTGATGATGCGGACATTCTATCTGCTTATGCAATAGATCCCACTCC
CGCTTTTGTCCCCAATATGACTAGTAGTGACAGTTCTGTG
```

(multiple) FASTA

Name of sequence (header): line starts with ‘>’



```
>AT1G01010.1|exon-1 | 1-283 | chr1:3631-3913 FORWARD LENGTH=283
AAATTATTAGATATACCAAACCAGAGAAAACAAATACATAATCGGAGAAATACAGATTACAGAGA
GGCGAAGCTCTTTACCCGGAAACCATTGAAATCGGACGGTTTAGTGAAAATGGAGGATCAAGTTC
CGAAACGACGAGGAGCTCGTTGGTCACTATCTCCGTAACAAAATCGAAGGAAACACTAGCCGCGAG
ACGAGAGGTCAACATCTGTAGCTACGATCCTTGGAAGTTGCGCT
>AT1G01010.1|exon-2 | 366-646 | chr1:3996-4276 FORWARD LENGTH=281
TCCAGTCAAAGTACAAATCCAGAGATGCTATGTGGTACTTCTTCTCTCGTAGAGAAAACAACAAAGGGAATCGACAGAGC
AGGACAACGGTTTCTGGTAAATGGAAGCTTACCCGAGAATCTGTTGAGGTCAAGGACCAGTGGGGATTTTGTAGTGAGGC
CTTTCGTGGTAAGATTGGTCATAAAAGGGTTTTGGTGTTCTCTCGATGGAAGATACCCTGACAAAACCAAATCTGATTGGC
TATCCACGAGTTCCACTACGACCTCTTACCAGAACATCAG
>AT1G01010.1|exon-3 | 856-975 | chr1:4486-4605 FORWARD LENGTH=120
AGGACATATGTCATCTGCAGACTTGAGTACAAGGGTGATGATGCGGACATTCTATCTGCTTATGCAATAGATCCCACTCC
CGCTTTTGTCCCCAATATGACTAGTAGTGCAGGTTCTGTG
```

one entry

Sequence lines (no limit!)

Analyze file - example

```
1 with open( "/vol/apbiokurs/data/AtCol0_Exons.fasta", "r" ) as f:
2     line = f.readline() #reading first line
3     line_counter = 0 #counting lines
4     while line:
5         line_counter += 1 #counting lines
6         line = f.readline()
7     #number in line_counter needs to be converted to string:
8     print("File contains " + str( line_counter ) + " lines")
```

Exercises C – Part1

- 1.1) Count number of sequences (= number of headers) in /vol/apbiokurs/data/AtCol0_Exons.fasta! (add file to your Drive: **link** in chat!)
- 1.2) Count number of sequence lines!
- 1.3) Count number of characters in document! (How many per line?)
- 1.4) How long are all contained sequences combined?
- 1.5) Calculate the average sequence length in this file!

And back again... writing into file!

Read:

```
1  
2 with open( "test.txt", "r" ) as f:  # "r" (read) ist default  
3     lines = f.readlines()  
4  
5  
6  
7
```

difference: r = read; w = write

Write:

```
8  
9 with open( "test2.txt", "w" ) as out:  
10     out.write( "hello world!" )
```

Writes a string into a file

- If output file does not exist, it will be created!
- File handle (f and out) can have any name!

Read & write

```
1 with open( "test.txt", "r" ) as f: #open file to read
2     with open( "test2.txt", "w" ) as out: #open file to write
3         line = f.readline() #read from first file
4         while line:
5             #this would be the place to apply filters
6             out.write( line )
7             line = f.readline()
```

Exercises D – Part1

- 1.1) Read the file `AtCol0_Exons.fasta` and write all headers (starting with '>') into a new file!
- 1.2) Read the file `AtCol0_Exons.fasta` and write the following:
 - Line if it is a header
 - Length of line if it is a sequence line
- 1.3) Calculate the number of sequences, the cumulative length, and the average length in the new file! Are they matching the values of the original file?
- 1.4) Write sequences into a new file if their length is a multiple of 10!

```

datei = open("/content/drive/MyDrive/Python_course_2021_data/AtCol0_Exons.fasta" , "r")
lines = datei.readlines()
datei.close()
counter = 0
for line in lines:
    if line.startswith(">AT1"):
        counter += 1

print("number of seqs: " + str(counter))

#cumulative length and average sequence length

cum_len = 0
number_seqs = 0
header = False
for line in lines:
    line = line.strip()
    if line.startswith(">AT1"):
        header = True
    elif line.startswith(">") and not line.startswith(">AT1"):
        header = False
    elif header == True:
        cum_len += len(line)

print("cum_len: " + str(cum_len))
print("average len: " + str(cum_len/counter))

```


White space characters

- New line ('`\n`') und tab ('`\t`') are special characters

```
print "hello\tworld!\nhello\tworld!\n"
```

- Python interprets these characters in print statements, but functions like `readline()` and `write()` do not!

=> New line needs to be added “manually” to each new line

```
1 with open( "test_file.txt", "w" ) as out:  
2     out.write( "first test" )  
3     out.write( "second test" )  
4     out.write( "third test\n" )  
5     out.write( "fourth test\n" )  
6     out.write( "fifth test" )  
7     out.write( "sixth test" )
```

How many lines does
this file have?

strip()

- Removes white space characters from borders of a string (often used for new lines at the line end):

```
1 line = ">name_of_first_seq\n"
2 print( line )
3 #>name_of_first_seq
4 # [empty line generated by \n ]
5 line = line.strip()
6 print( line )
7 #>name_of_first_seq
```

split()

- Separates a string at each given occurrence of the given substring (e.g. tab, comma, ...)
- Generates list of strings

```
1 #tab-delimited file
2 line = "spalte1\tspalte2\tspalte3\tspalte4\n"
3 #line should be splitted at tabs
4 columns = line.strip().split('\t')
5 print (columns)
6 #["spalte1", "spalte2", "spalte3", "spalte4"]
```


join()

- Combines strings of a list by putting a given substring between them (e.g. underline)
- Important: all elements of list need to be strings!

```
1 #tab-delimited file
2 line = "spalte1\tspalte2\tspalte3\tspalte4\n"
3 #line should be splitted at tabs
4 columns = line.strip().split('\t')
5 print(columns)
6 #["spalte1", "spalte2", "spalte3", "spalte4"]
7
8 new_line = "_".join( columns )
9 print(new_line)
10 #spalte1_spalte2_spalte3_spalte4
```

Exercises D – Part2

- 2.1) Read the file AtCol0_Exons.fasta and write the following:
 - Only ArabidopsisGenelIdentifier (e.g. AT1G01010)
 - Gene identifier, exon name, and exon length (tab-delimited)



```
>AT1G01010.1|exon-1 | 1-283 | chr1:3631-3913 FORWARD LENGTH=283
AAATTATTAGATATACCAAACCAGAGAAAACAAATACATAATCGGAGAAATACAGATTACAGAGAGCGAGAGAGATCGAC
GGCGAAGCTCTTTACCCGGAAACCATTGAAATCGGACGGTTTAGTGAAAATGGAGGATCAAGTTGGGTTTGGGTTCCGTC
CGAACGACGAGGAGCTCGTTGGTCACTATCTCCGTAACAAAATCGAAGGAAACACTAGCCGCGACGTTGAAGTAGCCATC
AGCGAGGTCAACATCTGTAGCTACGATCCTTGGAAC TTGCGCT
>AT1G01010.1|exon-2 | 366-646 | chr1:3996-4276 FORWARD LENGTH=281
TCCAGTCAAAGTACAAATCGAGAGATGCTATGTGGTACTTCTTCTCTCGTAGAGAAAACAACAAAGGGAATCGACAGAGC
AGGACAACGGTTTCTGGTAAATGGAAGCTTACCGGAGAATCTGTTGAGGTCAAGGACCAGTGGGGATTTTGTAGTGAGGG
CTTTCGTGGTAAGATTGGTCATAAAAGGGTTTTGGTGTTCCCTCGATGGAAGATACCCTGACAAAACCAAATCTGATTGGG
TTATCCACGAGTTCCACTACGACCTCTTACCAGAACATCAG
>AT1G01010.1|exon-3 | 856-975 | chr1:4486-4605 FORWARD LENGTH=120
AGGACATATGTCATCTGCAGACTTGAGTACAAGGGTGATGATGCGGACATTCTATCTGCTTATGCAATAGATCCCACTCC
CGCTTTTGTCCCCAATATGACTAGTAGTGCAGGTTCTGTG
```