Complexity Lab 2: Block Entropy Curves

[redacted]

last edited on June 24, 2011 10:48 AM by autoplectic

[redacted] ☑ Typeset

Print Worksheet Edit Text Undo Share Publish

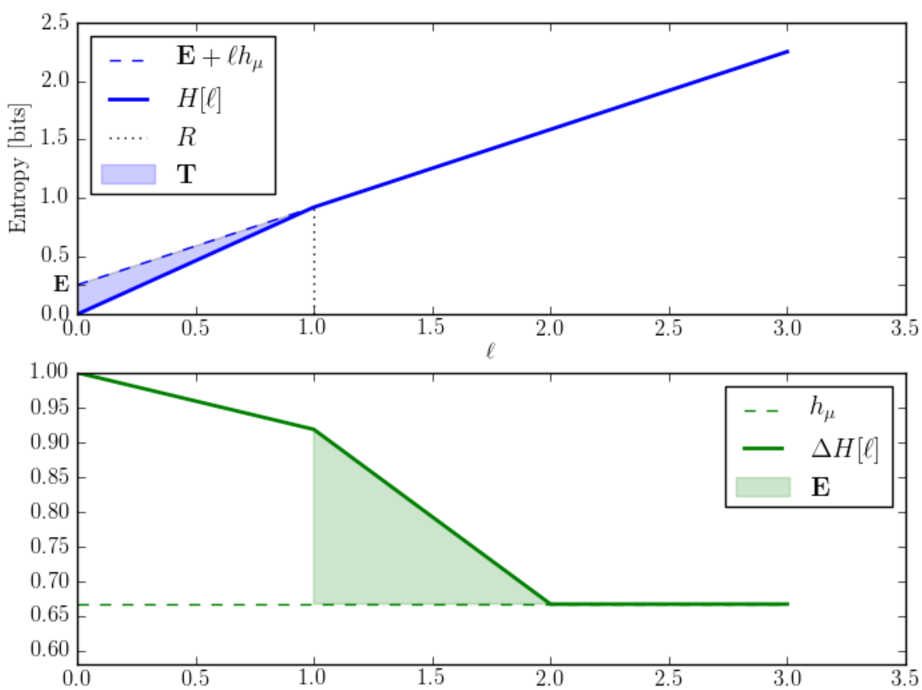
```
%hide
```

We're now going to look at various forms of information storage in processes. Since the fair and biased coins are trivial and memoryless, we will skip looking at them for now.

Here we are going to focus on two very similar, yet also very different, processes: the Golden Mean process and the Even process. We will be looking at block entropy curves again, but this time looking features of them that highlight information storage in the process: transient information in the block entropy curve and the excess entropy in both the block entropy curve and its derivative.

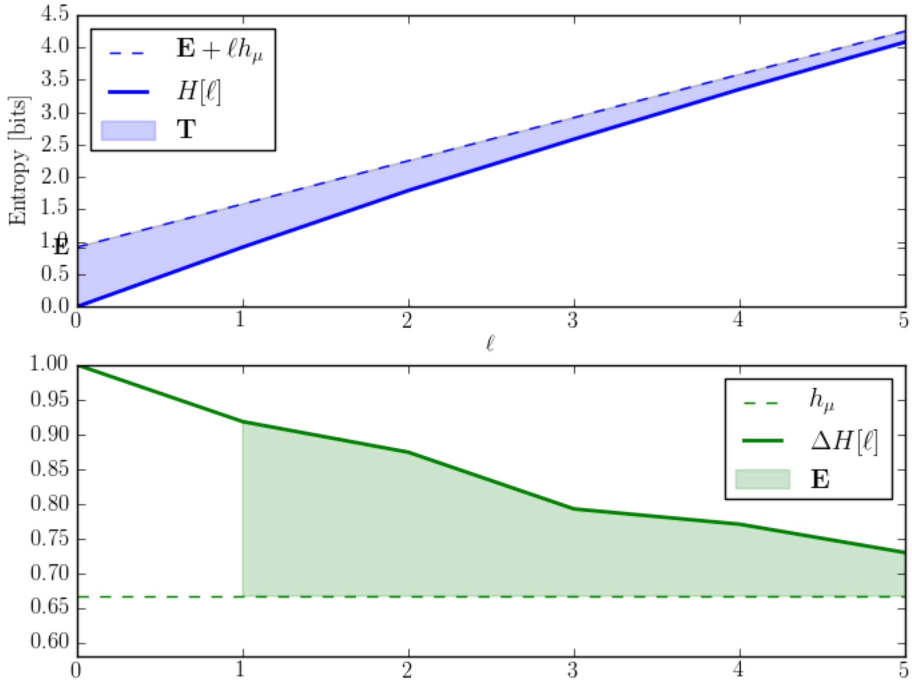
Let's look at a block entropy plot of the Golden Mean process:

```
block_entropy(goldenmean, 3)
```



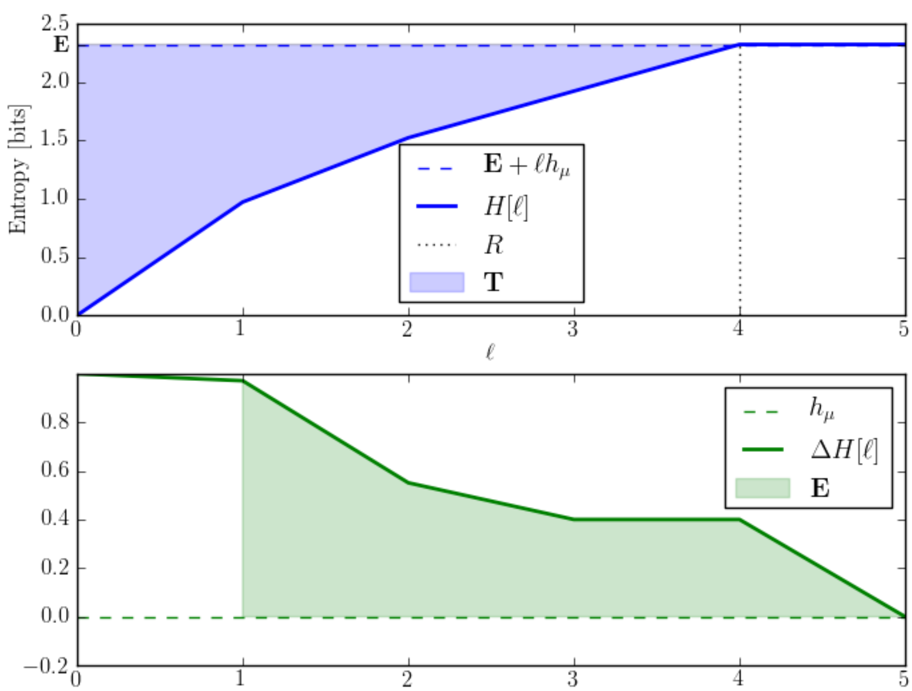
And now the even process:

```
block_entropy(even, length=5)
```



And finally a periodic process, for comparison:

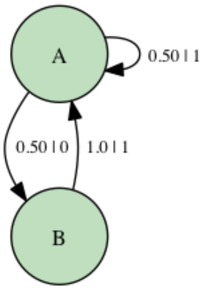
```
block_entropy(periodic)
```



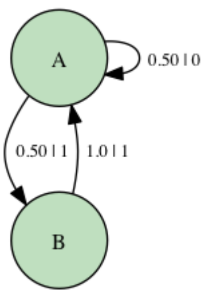
Next let's focus on just the golden mean and even processes, and highlight some similarities and differences that can be extracted using the process i-diagram atoms discussed earlier.

First, let's remind ourselves of what the generators for these two processes look like:

```
goldenmean.draw()
```




```
even.draw()
```



Next we will use a tool to create a table of different information quantities for the two processes, here we compute the entropy rate and the total correlation rate:

```
process_table([goldenmean, even, faircoin, periodic], ['hmu', 'rho'])
```

	hμ	ρμ
Golden Mean	0.66667	0.25163
Even	0.66667	0.25163
Coin, p = 0.5	1.00000	0.00000
Period-5 Process (11010)	0.00000	0.97095

Notice here that the entropy rate and total correlation rate are identical for the two processes. Those quantities do not differentiate the processes.

Let's take a closer look at the entropy rate by breaking it down into an ephemeral part r_μ and a structural part b_μ. Execute the cell when you are ready.

```
process_table([goldenmean, even, faircoin, periodic], ['rmu', 'bmu'])
```

	rμ	bμ
Golden Mean	0.45915	0.20752
Even	0.00000	0.66667
Coin, p = 0.5	1.00000	0.00000
Period-5 Process (11010)	0.00000	0.00000

Here we can see that these quantities are different for these two processes. Notice that for the even process b_μ is larger, implying that it is more structured despite having a similarly-structured generator to that of the golden mean process. We also notice that r_μ is zero, so all the randomness of the present symbol is relevant for future behavior.

Let's lastly look at the excess entropy, transient information, and Markov order for these two processes. Again, execute the cell when you are ready.

```
process_table([goldenmean, even, faircoin, periodic], ['E', 'T', 'R'])
```

evaluate

```
+--------------------------+-----------+-----------+---+
|                          |     E     |     T     | R |
+--------------------------+-----------+-----------+---+
| Golden Mean              |  0.25163 |  0.25163 | 1 |
| Even                     |  0.91830 |  3.16993 | ∞ |
| Coin, p = 0.5            |  0.00000 |  0.00000 | 0 |
| Period-5 Process (11010) |  2.32193 |  4.87291 | 4 |
+--------------------------+-----------+-----------+---+
```

Notice that the markov order for the even process is infinite, which couldn't be any more different from the golden mean which is markov order 1. For the golden mean, both **E** and **T** the same which is a consequence of R = 1. These values are also smaller than the information measures for the even process, implying that the golden mean process stores less information. **E** being large for the even process means that it shares more information between the past and the future, and **T** being large tells us that it is difficult to synchronize to.