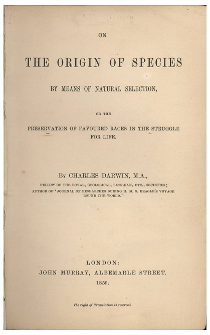
The 100 most common words in On the Origin of Species

Determine the frequencies of all words in Darwin's "On the origin of Species" and print the 100 most common words. Compare these words with the most common words in English (use Wikipedia, notice that there all forms of a verb are combined, i.e.: **is**, **are**, **was,** etc. all combine into **be**). Which words in the top 100 are typical for this book?



ON

THE ORIGIN OF SPECIES

BY MEANS OF NATURAL SELECTION,

OR THE

PRESERVATION OF FAVOURED RACES IN THE STRUGGLE FOR LIFE.

By CHARLES DARWIN, M.A.,

FELLOW OF THE ROYAL, GEOLOGICAL, LINNÆAN, ETC., SOCIETIES;
AUTHOR OF 'JOURNAL OF RESEARCHES DURING H. M. S. BEAGLE'S VOYAGE ROUND THE WORLD.'

LONDON:
JOHN MURRAY, ALBEMARLE STREET.
1859.

The right of Translation is reserved.

Data

http://www.gutenberg.org/cache/epub/2009/pg2009.txt
https://en.wikipedia.org/wiki/Most_common_words_in_English

Suggestions

Read the file line by line, use a regular expression to replace all non-word characters by spaces: `line = re.sub("\W"," ",line)`
Split the line in words and use a dictionary to count the occurrence of each word (think about upper and lower case versions of a word). Print the dictionary and use the Linux sort and head or tail commands to get the top 100.

Possible extensions

-In Wikipedia it says: "*The Reading Teacher's Book of Lists* claims that the first 25 words make up about one-third of all printed material in English, and that the first 100 make up about one-half of all written material". Does this also hold true for "On the Origin of Species"?
-To better compare your list with that in Wikipedia, you would have to combine all forms of a verb into one (is, are, were -> be). Try to find a way to do that in your script for the verbs in your top 100.

Side note:

Here you can see an animation of the changes from the first to the sixth edition of the book: https://fathom.info/traces/

Protein classification using Prosite patterns

In week 1 you saw that Prosite patterns are powerful tools to find protein domains, and that they can be converted into python regular expressions. Write a python script that takes as command line arguments a Prosite pattern and a FASTA file, and then prints out the proteins for which the sequence matches the Prosite pattern.

Data

http://prosite.expasy.org/
The **sp_human_single_line.fasta** file from the first week (see Blackboard).

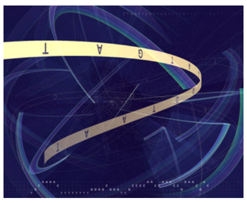
Suggestions

import the python modules sys and re. Create a python script that reads the input file line by line. Use a regular expression to convert the given Prosite pattern into a python regular expression.

Possible extensions

-Enhance the script so that it is able to find all Prosite patterns for a certain protein sequence. The Prosite patterns can be found in this file: ftp://ftp.expasy.org/databases/prosite/prosite.dat
-Enhance the script so that it can find all Prosite patterns in all proteins in the **sp_human_single_line.fasta** file.

Building a BLAST website



Using the Flask framework that you encountered in week 5, combined with the running the BLAST program from the command line, create a website that allows a user to enter a protein sequence and then click a button to have that sequence searched in a BLAST database (for instance with the human proteins from SwissProt). The result of the BLAST search should then be presented on the website, for instance in tabular format.

Suggestions

Flask web framework, the **blastp** and **makeblastdb** programs. A MySQL database is not needed. You can make a text input field in HTML using something like this (see also the slides from week 5, day 4):

```
<input type="text" name="sequence">
```

If you put this in the HTML form, you will get an additional key in the **request.args** dictionary called "sequence".

Possible extensions

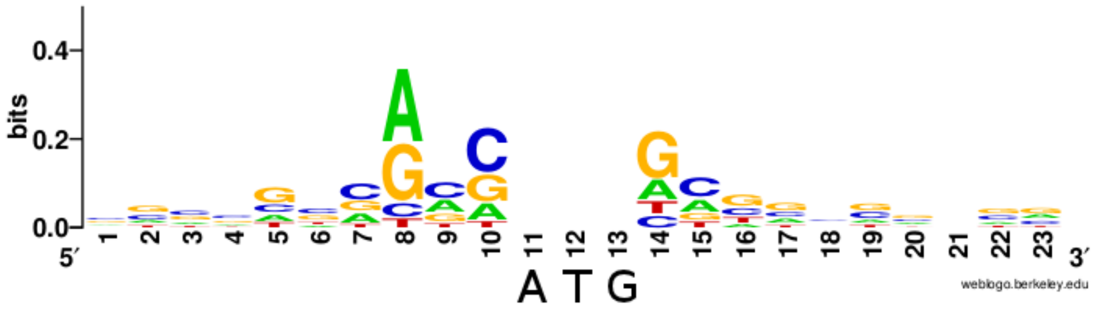
-Allow the user to set a cut-off for the E-value
-Use another one of BLAST's output formats that is not tabular, for instance the default output format
-Create hyperlinks for the found proteins to the UniProt website
Example: http://www.uniprot.org/uniprot/1433E_HUMAN

Side note

Here you can find an artist's impression of the BLAST algorithm, which was used in the Hulk movie:
http://benfry.com/genomevalence/

The Kozak consensus sequence

Marilyn Kozak discovered that there is a preference for certain nucleotides at specific positions right before and after a start codon. She discovered that by inspecting the initiation codons in 699 mRNAs from different vertebrates. The weblogo image below was generated by inspecting 10,000 human mRNAs (https://en.wikipedia.org/wiki/Kozak_consensus_sequence). With what you have learned in the past weeks, you can redo this analysis using the human coding sequences and the upstream flanks of the coding sequences. Determine the frequencies of each nucleotide at specific positions upstream and downstream of the initiation codon.



Data

You can download the coding sequences and their upstream flanks (two different queries) from Ensembl's biomart: http://www.ensembl.org/biomart/martview/
Alternatively you could use this set of predicted coding sequences, these do not include the upstream region:
ftp://ftp.ensembl.org/pub/release-86/fasta/homo_sapiens/cdna/Homo_sapiens.GRCh38.cdna.abinitio.fa.gz

Suggested tools

wget, grep, cut, sort, uniq, or a python script that counts the frequencies of the different nucleotides at specific positions after the start codon. Although the input files contain multiline FASTA sequences, you probably only have to look at the first line after the ">" line.

Possible extensions

-You could compare different species, for instance from different kingdoms (plants, animals, fungi, bacteria)
-You could create a WebLogo image like the one above using the weblogo tool.
-Test how far from the initiation codon you need to go to have no more bias.

Tracking data of 20 Common Cranes

Track 20 common cranes (*Grus grus*) as they migrate across Europe. Determine where they go to breed, and where to spend the wintertime, and in which time of the year they migrate. By taking the coordinates of tens of thousands of measurements per animal you can get a detailed insight in how active these animals are throughout the year(s).

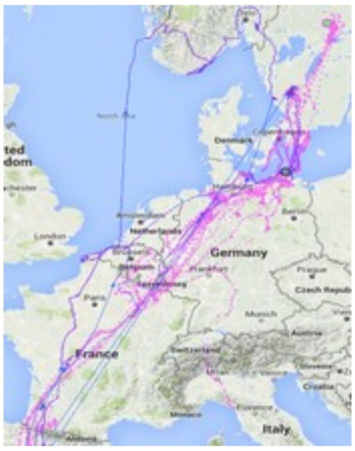
Data

We've used the data before, in weeks 1, 2 and 3. The data is derived from Movebank (https://www.movebank.org/).

Dataset: http://www.bioinformatics.nl/courses/BIF-50806/Examples/Common_Crane_tracking/GPS_telemetry_of_Common_Cranes_Sweden.csv.gz

Suggested Tools

There is an example Notebook with code available from Blackboard (under Projects). You can find strategies for calculating differences between coordinates (in m or km) and between timestamps. Plotting e.g. km migrated per week, can be done using simple plotting tools described in W4D3 (Flowers). For making KMLs, have a look at the 'Dinosaurs' practical (W4D4).



Possible extensions

a) Plot the coordinates on Google Maps or G. Earth. Instead of individual placemarks, you can also draw lines that mark the route of the cranes.
b) Create a database where you divide the data into multiple tables to decrease redundancy and allow efficient querying of the data.

Identifying species using DNA barcoding

Identifying species based on their morphology (outer appearance) is often challenging, and, interestingly, a very specialized skill. Knowing species composition is important in many fields. For instance, different species of mosquito may have different properties in their ability to transmit malaria. Molecular identification can be far easier automated and, provided there is a global database of information to compare, can be far more accurate. Molecular identification strategies are nowadays common in microbial studies, and increasingly applied to other fields as well, for instance in counting the abundance of fish larvae, almost impossible to do high-throughput with conventional means, but very important to guide fisheries management. You can make an automated reporting tool that extends on the example of W4D5 by taking the best hits and then, after aligning them, clusters the sequences. Additional analysis and visualization can be done by making a phylogenetic tree.

Data

Mosquito or fish DNA barcoding databases (www.ibol.org). The Mosquito database is smaller and simpler in structure (all based on Cytochrome Oxidase 1 sequences, the fish database has often other DNA barcodes too and generally is larger). The Mosquito data is the same as in W4D5 'Combining Tools'.
Fish data: http://www.bioinformatics.nl/courses/BIF-50806/PCB_project_data/Fish_barcoding_data.tsv.gz

Suggested Tools

Example code can be taken from the practical in W4D5 ('combining tools'). Simple clustering can be done by making your own python code, e.g. by taking into account the distribution of sequence similarities reported by Blast. Alignments can be made using Clustal or Muscle. Phylogeny can be made using either graphical tools (e.g. MEGA) or command line (e.g. RaxML, with BioPython for visualization). For the extension, you can use example code from W5 (databases and web programming).

Possible extensions

a) Determine the geographic location of the sequences that are most similar.
b) Create a database of the data where you make multiple tables to efficiently store the data and make it easy to query on certain traits.
c) Create a web frontend where you can submit a DNA sequence to retrieve a blast result that is done on-the-fly.

Identify flower species based on color profiles

Identify which of four species (pansy, sunflower, daisy, or tigerlillie) any of around 200 images belongs. You will do this by using the color histograms of the images to make clusters based on similarity in color use in the images.

Data

The images come from this website:
http://www.robots.ox.ac.uk/~vgg/data/flowers/17/
A selection of four species with distinct color profiles were made based on that dataset, which can be downloaded from Myers:
http://www.bioinformatics.nl/courses/BIF-50806/PCB_project_data/Selection_Flowers.zip

Suggested Tools

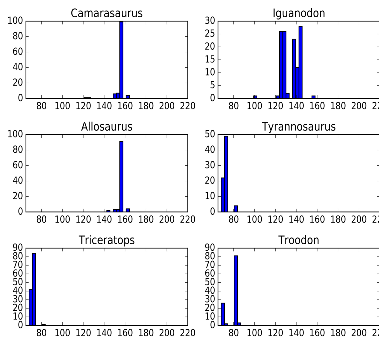
Example code can be found in the exercises of W4D3. Automate the various steps and make a matrix of similarities based on the color. Vary some of the parameters, such as the number of bins in the histograms, or the zoom level in the image, to see if your results improve. You can choose to program a simple clustering algorithm yourself, or use one that you find in, e.g., SciPy or Scikit-Learn.

Possible extensions

a) Test multiple clustering or 'machine learning' algorithms found in Scikit-learn (if you're interested, additional example code can be provided).
b) More fancy identification strategies can be applied, e.g. based on available 'masks' for the flowers which makes it possible to better identify the color profiles against background. Example code can be provided if you are interested.

Dinosaur diversity in space and time

We are all familiar with many of the iconic dinosaur species – T. rex, *Brontosaurus*, *Triceratops*, *Stegosaurus*, etc. However, a T.rex eating a *Stegosaurus* is actually as temporally unlikely as a lawyer being eaten by a T. rex. Discover the dynamics of dinosaur diversity by analyzing the paleontological database. Create summary information for (selected or all) dinosaur genera, track diversity of dinosaurs through time, and analyze the occurrence of species and genera in space as well. Make the information discoverable by using Google Maps or Earth, and tweak the code provided earlier to work with different colors for the placemarks, for instance.



Data

Data derived from PaleoDB (paleobiodb.org). Dinosaur data can downloaded according to W4D4 'Dinosaurs!' exercise.
Note: if you have preferences for other taxa, e.g. because you believe you could derive more interesting or robust statistics, e.g. on diversity through time, you are free to select whatever you like!

Suggested Tools

Example code can be taken from W4D4. Simple plotting code and translation to KML can be found there too. Most things should be doable in plain Python. Try to 'remember' certain (counts of) datapoints from a collection of datapoints by storing them in dictionaries and lists, so that at the end they can be aggregated and analyzed.

Possible extensions

a) Go nuts and be creative with the data. See what patterns you can derive from the data beyond the obvious.
b) Create a database of the data that can be queried through Python
c) Create a dynamic website where you allow selecting of a Dinosaur species or genus to be analyzed in real time.