SE - Computational Methods for Text Stemmatology
Univ.-Prof. Dr Tara L Andrews, Dott.ssa.mag. Elisabetta Magnanti
Florian Klement & David Siegl

Difference Matrices and Leitfehler-detection: The longest common subsequence and measures of uniqueness
Converting and Adapting Dieter Bachmann's & Philipp Roelli's Perl Script to Python

Introduction:

Our aim with this project was to contribute to the larger whole of the stemmatological community in a structural way that would continue to carry a lasting impact on the future approach of digital researchers, by promoting convenient and approachable practices to allow for a low-threshold of entry. The topic we chose, the script by Roelli and Bachmann, carries weight through the utility it offers practitioners since 2010, yet parts of its merits are marred by the constraints imposed on its users due to the language barrier they might see themselves confronted by when having to use Perl, a scripting language that has fallen out of favour with the scientific community in recent years. Thus, our main effort lies in being able to allow the implementation by Roelli & Bachmann, which was written for a specific project in mind (i. e. comparing Latin texts), to reach a much broader audience of digital researchers from different backgrounds and empower them to adapt the quality of their results via individual adjustments fitting for their respective language of research and tradition. Our realisation of it offers more flexibility in terms of implementing additional features, like a parameterized version of the letters or terms to be substituted for one another, within the normalisation and general preprocessing steps, as well as additional freedom of choice in substitutions and connectivity by making use of Python's rich ecosystem and community. It was especially important for us to streamline and standardise the procedure of generating I/O within the creation of digital stemmas and its surrounding analyses.

The algorithm diff, created by Ned Konz, on which the works of Roelli and Bachmann build upon shares caveats in its application with more commonly used methods of stemmatology, as well as those borrowed from phylogenetics and still struggles with inconsistencies and lacunae. Diff operates on the basis of pairwise juxtaposition, whereas each manuscript is compared with another in order to compute a quantitative relatedness of the whole corpus. The basic notion of mathematical distance calculation is expressed via the so-called ‘edit distance’, referring to a given difference between two strings expressed in the number of edits or basic operations necessary to transform the first text into the second or vice versa. This distance metric can in the most general sense be applied on a character level but, as Roelli and Bachmann also point out, when it comes to stemmatological analysis, it seems more plausible to operate on a word level. - i. e. different manuscripts are considered as ordered lists of words in order to compare them. In a more practical sense the edit distance should therefore reflect the number of deletions or insertions made by a scribe, while the final result of the edit distance computation should in the end return the measure of distance connecting any two manuscripts in the shape of a matrix. Among others, Roelli’s & Bachmann’s implementation added to it further adaptations in order to bring the algorithm closer in line with the practices of scribal transmission and general features for the preprocessing of natural language textual witnesses. (Cf. Roelli & Bachmann, 2010, 315-317)

Our modification opted to drop diff altogether, given that the input format has been adapted from plain text, to the readily available shape of output automated collation tools like Collatex provide. In this shape, the alignment and standardisation of traditions is already taken care of and the act of comparison and weighting is a simpler procedure, which can do without requiring additional, external packages.

In order to evaluate the quality of the modified script compared to its original implementation, we selected several collated texts for comparison and made use of tools like the PHYLIP tree visualizer offered by Trex-online. More on this matter can be found in the following sections.

Description:

The adapted Python Script *lf_new5.py* provides the detection of Leitfehlers from a given text collation. The leitfehlers are pasted into a separate text document named "leitfehler_list.txt". Additionally it produces a lower-triangular distance matrix which can then be further used as input for PHYLIP and its various functionalities. The script starts by reading in the input, which consists of tabular data (in either .csv or .txt format) with its columns containing labels for the manuscripts followed by a line for every word in each manuscript. The script standardised the input by removing punctuation; it also stores the labels and contents of the manuscripts. Optionally the user can also provide the script with a file of regex or other substitution terms which are then parsed through the preprocessing pipeline as an additional step. The function called dodiff takes two arrays of words as an input and returns the numeric difference between the two arrays. For this it compares the contents of each manuscript to the contents of every other manuscript that precedes it in the array. The actual leitfehler calculation is done by counting the number of times each word appears in each manuscript and comparing these counts between pairs of manuscripts. Two rating functions iterate over each term for each pairing of texts, to see if either, both, or neither of the words is included in a manuscript. The script calculates a score for each word in the manuscripts based on the number of manuscripts in which it appears and its global frequency in the manuscripts. The scores are normalised by dividing them by the number of occurrences of each leitfehler candidate in the manuscripts.

Parameters:

For making the handling of our script compliant with the UNIX-Standards, we made use of the standard Python library *argparse* in order to implement a number of parameters which can directly be called and modified by the end user via a CLI. The different parameters are as follows:

-h / --help: This standard parameter provides the user with a documentation overview of the script and its different parameters.

-f /--file: This parameter is required and denotes the file path to the text collation in a tabular data format (either .csv or .txt).

-c / --cut: This parameter is optional and functions as a cut off threshold for the leitfehler detection score. Its default value = 0. A noticeable effect starts at roughly 400. For larger datasets, like Heinrichi, it is advisable to set a value preemptively to decrease the runtime considerably.

-d / --debug: This parameter is optional and can take either the binary value of 0 or 1. If the value = 0 then the script will print out only the matrix of the computed diff scores and if the value = 1 then the script will also create a separate text file called *leitfehler_list.txt* which contains potential leitfehlers and their scores.

-delim / --delim: This parameter is optional and takes as an input the separator of the collated input file. Its default value = comma but it can also handle other common delimiters such as tab.

-e / --encoding: This parameter is optional and denotes the textual encoding of the input file. Its default value = ‘utf-8’, for other encoding options and for the explicit format please refer to the base Python function open().

-r / --regex: This parameter is optional and provides the user the option to specify an additional filepath with regex expressions which will then be applied to the collated text within the preprocessing pipeline. The expected format of the regex file: every substitution expression consisting of two lines where
first line = matching pattern
second line = substitution pattern

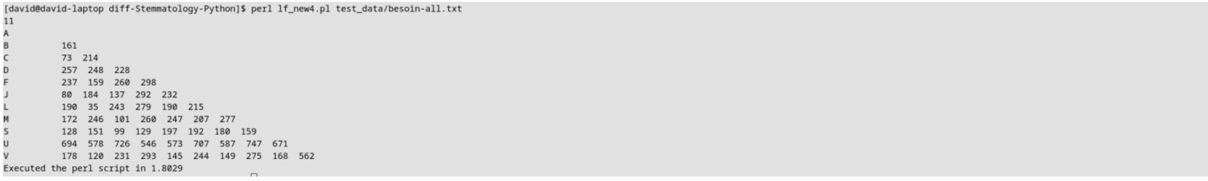
-sm / --scoremax: This parameter is optional and makes it possible the finetune the probability of the leitfehler detection. The expected input is of the type integer while its default value = 1.

Some benchmarks and comparisons:

Generally speaking the Python implementation outperformed its Perl ancestor by a moderate margin, especially in the case of the use of the Heinrichi text, where the difference is capable of accumulating more noticeable headway as the runtime goes up. In terms of the difference matrices, some subtle changes within the results can be seen, most notably due to our added preprocessing steps, as well as the subtle tweaks in the ranking of differences and our substitution of the diff algorithm with a simpler, more streamlined measure of comparison, entirely from within the base packages of python.

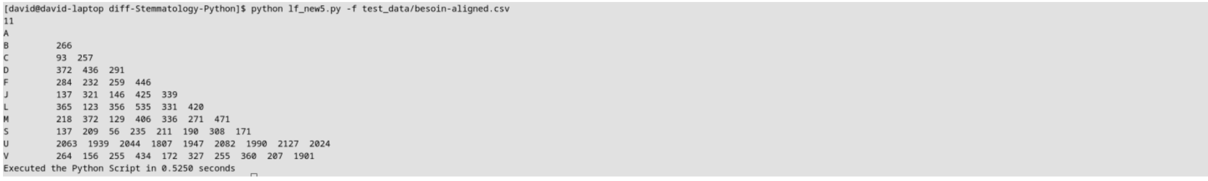
The following pictures show side by side comparisons of the lower half distance matrices of the Perl and Python scripts, as well as their runtime. Additionally we provide a visual comparison of a PHYLIP stemma created from the respective output, these were generated with the help of the trex-online tool of the Université du Québec à Montréal. (Cf. Trex-online)

The texts we used were chosen from the normalised and edited selection of datasets made available by the Helsinki Institute for Information Technology on their website for the Computer-Assisted Stemmatology Challenge. (Cf. Computer Assisted Stemmatology Challenge)



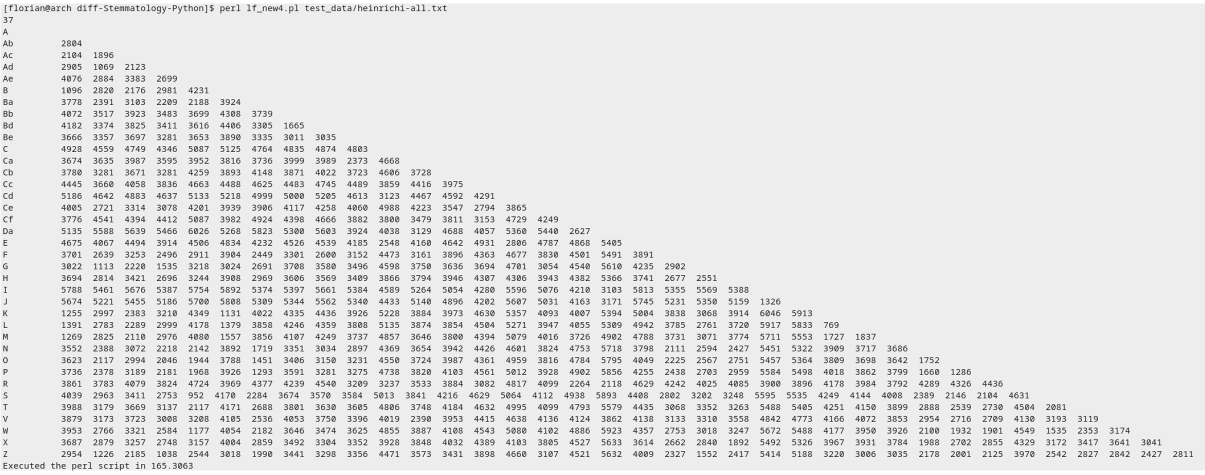
```
[david@david-laptop diff-Stemmatology-Python]$ perl lf_new4.pl test_data/besoin-all.txt
11
A
B         161
C         73  214
D         257  248  228
F         237  159  260  298
J         80  184  137  292  232
L         190  35  243  279  190  215
M         172  246  101  260  247  207  277
S         128  151  99  129  197  192  180  159
U         694  578  726  546  573  707  587  747  671
V         178  120  231  293  145  244  149  275  168  562
Executed the perl script in 1.8029
```

Vis. 1: Runtime and Distance Matrix for *Notre Besoin* with lf_new4.pl



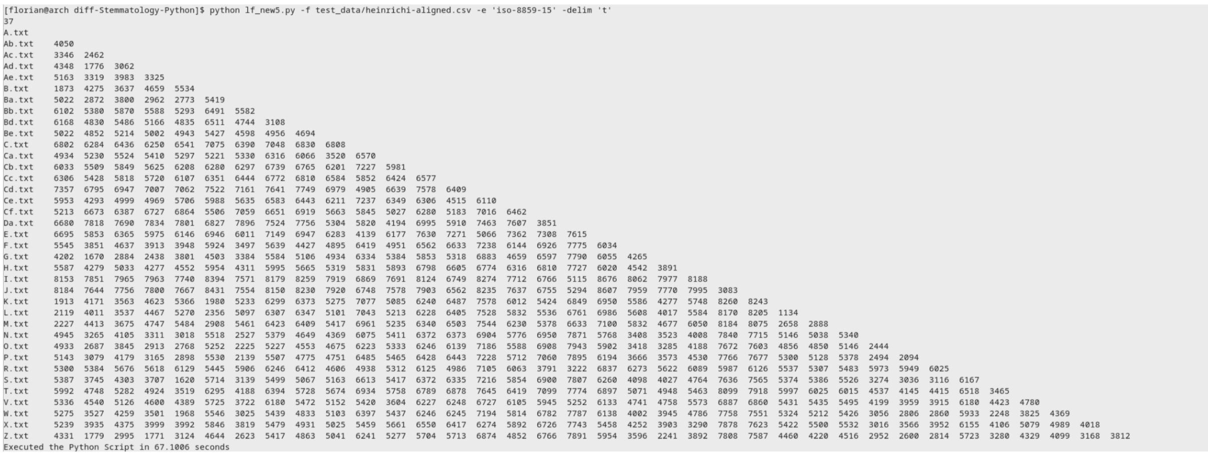
```
[david@david-laptop diff-Stemmatology-Python]$ python lf_new5.py -f test_data/besoin-aligned.csv
11
A
B         266
C         93  257
D         372  436  291
F         284  232  259  446
J         137  321  146  425  339
L         365  123  356  535  331  420
M         218  372  129  406  336  271  471
S         137  209  56  235  211  190  308  171
U         2063  1939  2044  1807  1947  2082  1990  2127  2024
V         264  156  255  434  172  327  255  360  207  1901
Executed the Python Script in 0.5250 seconds
```

Vis. 2: Runtime and Distance Matrix for *Notre Besoin* with *lf_new5.py*



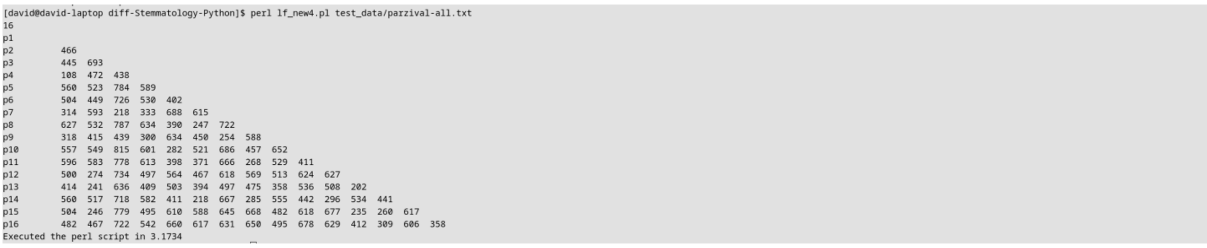
```
[florian@arch diff-Stemmatology-Python]$ perl lf_new4.pl test_data/heinrichi-all.txt
37
A
Ab        2804
Ac        2104  1896
Ad        2905  1069  2123
Ae        4076  2884  3383  2699
B         1096  2820  2176  2981  4231
Ba        3778  2391  3103  2209  2188  3924
Bb        4072  3517  3923  3483  3699  4308  3739
Bd        4182  3374  3825  3411  3616  4406  3305  1665
Be        3666  3357  3697  3281  3653  3890  3335  3011  3035
C         4928  4559  4749  4346  5087  5125  4764  4835  4874  4803
Ca        3674  3635  3987  3595  3952  3816  3736  3999  3989  2373  4668
Cb        3780  3281  3671  3281  4259  3893  4148  3871  4022  3723  4606  3728
Cc        4445  3660  4058  3836  4663  4488  4625  4483  4745  4489  3859  4416  3975
Cd        5186  4642  4883  4637  5133  5218  4999  5000  5205  4613  3123  4467  4592  4291
Ce        4005  2721  3314  3078  4201  3939  3906  4117  4258  4060  4988  4223  3547  2794  3865
Cf        3776  4541  4394  4412  5087  3982  4924  4398  4666  3882  3800  3479  3811  3153  4729  4249
Da        5135  5588  5639  5466  6026  5268  5823  5300  5603  3924  4038  3129  4688  4057  5360  5440  2627
E         4675  4067  4494  3914  4506  4834  4232  4526  4539  4185  2548  4160  4642  4931  2806  4787  4868  5405
F         3701  2639  3253  2496  2911  3904  2449  3301  2600  3152  4473  3161  3896  4363  4677  3830  4501  5491  3891
G         3022  1113  2220  1535  3218  3024  2691  3708  3580  3496  4598  3750  3636  3694  4701  3054  4540  5610  4235  2902
H         3694  2814  3421  2696  3244  3908  2969  3606  3569  3409  3866  3794  3946  4307  4306  3943  4382  5366  3741  2677  2551
I         5788  5461  5676  5387  5754  5892  5374  5397  5661  5384  4589  5264  5054  4280  5596  5076  4210  3103  5813  5355  5569  5388
J         5674  5221  5455  5186  5700  5808  5309  5344  5562  5340  4433  5140  4896  4202  5607  5031  4163  3171  5745  5231  5350  5159  1326
K         1255  2997  2383  3210  4349  1131  4022  4335  4436  3926  5228  3884  3973  4630  5357  4093  4007  5394  5004  3838  3068  3914  6046  5913
L         1391  2783  2289  2999  4178  1379  3858  4246  4359  3808  5135  3874  3854  4504  5271  3947  4055  5309  4942  3785  2761  3720  5917  5833  769
M         1269  2825  2110  2976  4080  1557  3856  4107  4249  3737  4857  3646  3800  4394  5079  4016  3726  4902  4788  3731  3071  3774  5711  5553  1727  1837
N         3552  2388  3072  2218  2142  3892  1719  3351  3034  2897  4369  3654  3942  4426  4601  3824  4753  5718  3798  2111  2594  2427  5451  5322  3909  3717  3686
O         3623  2117  2994  2046  1944  3788  1451  3406  3150  3231  4550  3724  3987  4361  4959  3816  4784  5795  4049  2225  2567  2751  5457  5364  3809  3698  3642  1752
P         3736  2378  3189  2181  1968  3926  1293  3591  3281  3275  4738  3820  4103  4561  5012  3928  4902  5856  4255  2438  2703  2959  5584  5498  4018  3862  3799  1660  1286
R         3861  3783  4079  3824  4724  3969  4377  4239  4540  3209  3237  3533  3884  3082  4817  4099  2264  2118  4629  4242  4025  4085  3900  3896  4178  3984  3792  4289  4326  4436
S         4039  2963  3411  2753  952  4170  2284  3674  3570  3584  5013  3841  4216  4629  5064  4112  4938  5893  4408  2802  3202  3248  5595  5535  4249  4144  4008  2389  2146  2104  4631
T         3988  3179  3669  3137  2117  4171  2688  3801  3630  3605  4806  3748  4184  4632  4995  4099  4793  5579  4435  3068  3352  3263  5488  5405  4251  4150  3899  2888  2539  2730  4504  2081
V         3879  3173  3723  3008  3208  4105  2536  4053  3750  3396  4019  2390  3953  4415  4638  4136  4124  3862  4138  3133  3310  3558  4842  4773  4166  4072  3853  2954  2716  2709  4130  3193  3119
W         3953  2766  3321  2584  1177  4054  2182  3646  3474  3625  4855  3887  4108  4543  5080  4102  4886  5923  4357  2753  3018  3247  5672  5488  4177  3950  3926  2100  1932  1901  4549  1535  2353  3174
X         3687  2879  3257  2748  3157  4004  2859  3492  3304  3352  3928  3848  4032  4389  4103  3805  4527  5633  3614  2662  2840  1892  5492  5326  3967  3931  3784  1988  2702  2855  4329  3172  3417  3641  3041
Z         2954  1226  2185  1038  2544  3018  1990  3441  3298  3356  4471  3573  3431  3898  4660  3107  4521  5632  4009  2327  1552  2417  5414  5188  3220  3006  3035  2178  2001  2125  3970  2542  2827  2842  2427  2811
Executed the perl script in 165.3063
```

Vis. 3: Runtime and Distance Matrix for *Heinrichi* with *lf_new4.pl*



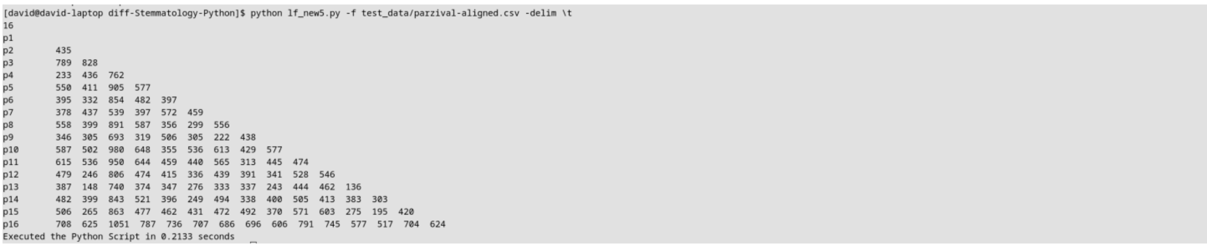
```
[florian@arch diff-Stemmatology-Python]$ python lf_new5.py -f test_data/heinrichi-aligned.csv -e 'iso-8859-15' -delim 't'
37
A.txt
Ab.txt    4050
Ac.txt    3346  2462
Ad.txt    4348  1776  3062
Ae.txt    5163  3319  3983  3325
B.txt     1873  4275  3637  4659  5534
Ba.txt    5022  2872  3800  2962  2773  5419
Bb.txt    6102  5380  5870  5588  5293  6491  5582
Bd.txt    6168  4830  5486  5166  4835  6511  4744  3108
Be.txt    5022  4852  5214  5002  4943  5427  4598  4956  4694
C.txt     6802  6284  6436  6250  6541  7075  6390  7048  6830  6808
Ca.txt    4934  5230  5524  5410  5297  5221  5330  6316  6066  3520  6570
Cb.txt    6033  5509  5849  5625  6208  6280  6297  6739  6765  6201  7227  5981
Cc.txt    6306  5428  5818  5720  6107  6351  6444  6772  6810  6584  5852  6424  6577
Cd.txt    7357  6795  6947  7007  7062  7522  7161  7641  7749  6979  4905  6639  7578  6409
Ce.txt    5953  4293  4999  4969  5706  5988  5635  6583  6443  6211  7237  6349  6306  4515  6110
Cf.txt    5213  6673  6387  6727  6864  5506  7059  6651  6919  5663  5845  5027  6280  5183  7016  6462
Da.txt    6680  7818  7690  7834  7801  6827  7896  7524  7756  5304  5820  4194  6995  5910  7463  7607  3851
E.txt     6695  5853  6365  5975  6146  6946  6011  7149  6947  6283  4139  6177  7630  7271  5066  7362  7308  7615
F.txt     5545  3851  4637  3913  3948  5924  3497  5639  4427  4895  6419  4951  6562  6633  7238  6144  6926  7775  6034
G.txt     4202  1670  2884  2438  3801  4503  3384  5584  5106  4934  6334  5384  5853  5318  6883  4659  6597  7790  6055  4265
H.txt     5587  4279  5033  4277  4552  5954  4311  5995  5665  5319  5831  5893  6798  6605  6774  6316  6810  7727  6020  4542  3891
I.txt     8153  7851  7965  7963  7740  8394  7571  8179  8259  7919  6869  7691  8124  6749  8274  7712  6766  5115  8676  8062  7977  8188
J.txt     8184  7644  7756  7800  7667  8431  7554  8150  8230  7920  6748  7578  7903  6562  8235  7637  6755  5294  8607  7959  7770  7995  3083
K.txt     1913  4171  3563  4623  5366  1980  5233  6299  6373  5275  7077  5085  6240  6487  7578  6012  5424  6849  6950  5586  4277  5748  8260  8243
L.txt     2119  4011  3537  4467  5270  2356  5097  6307  6347  5101  7043  5213  6228  6405  7528  5832  5536  6761  6986  5608  4017  5584  8170  8205  1134
M.txt     2227  4413  3675  4747  5484  2908  5461  6423  6409  5417  6961  5235  6340  6503  7544  6230  5378  6633  7100  5832  4677  6050  8184  8075  2658  2888
N.txt     4945  3265  4105  3311  3018  5518  2527  5379  4649  4369  6075  5411  6372  6373  6904  5776  6950  7871  5768  3408  3523  4008  7840  7715  5146  5038  5340
O.txt     4933  2687  3845  2913  2768  5252  2225  5227  4553  4675  6223  5333  6246  6139  7186  5588  6908  7943  5902  3418  3285  4188  7672  7603  4856  4850  5146  2444
P.txt     5143  3079  4179  3165  2898  5530  2139  5507  4775  4751  6485  5465  6428  6443  7228  5712  7060  7895  6194  3666  3573  4530  7766  7677  5300  5128  5378  2494  2094
R.txt     5300  5384  5676  5618  6129  5445  5906  6246  6412  4606  4938  5312  6125  4986  7105  6063  3791  3222  6837  6273  5622  6089  5987  6126  5537  5307  5483  5973  5949  6025
S.txt     5387  3745  4303  3707  1620  5714  3139  5499  5067  5163  6613  5417  6372  6335  7216  5854  6900  7807  6260  4098  4027  4764  7636  7565  5374  5386  5526  3274  3036  3116  6167
T.txt     5992  4748  5282  4924  3519  6295  4188  6394  5728  5674  6934  5758  6789  6878  7645  6419  7099  7774  6897  5071  4948  5463  8099  7918  5997  6025  6015  4537  4145  4415  6518  3465
V.txt     5336  4540  5126  4600  4389  5725  3722  6180  5472  5152  5420  3604  6227  6248  6727  6105  5945  5252  6133  4741  4758  5573  6887  6860  5431  5435  5495  4199  3959  3915  6180  4423  4780
W.txt     5275  3527  4259  3501  1968  5546  3025  5439  4833  5103  6397  5437  6246  6245  7194  5814  6782  7787  6138  4002  3945  4786  7758  7551  5324  5212  5426  3056  2806  2860  5933  2248  3825  4369
X.txt     5239  3935  4375  3999  3992  5846  3819  5479  4931  5025  5459  5661  6550  6417  6274  5892  6726  7743  5458  4252  3903  3290  7878  7623  5422  5500  5532  3016  3566  3952  6155  4106  5079  4989  4018
Z.txt     4331  1779  2995  1771  3124  4644  2623  5417  4863  5041  6241  5277  5704  5713  6874  4852  6766  7891  5954  3596  2241  3892  7808  7587  4460  4220  4516  2952  2600  2814  5723  3280  4329  4099  3168  3812
Executed the Python Script in 67.1006 seconds
```

Vis. 4: Runtime and Distance Matrix for *Heinrichi* with *lf_new5.py*



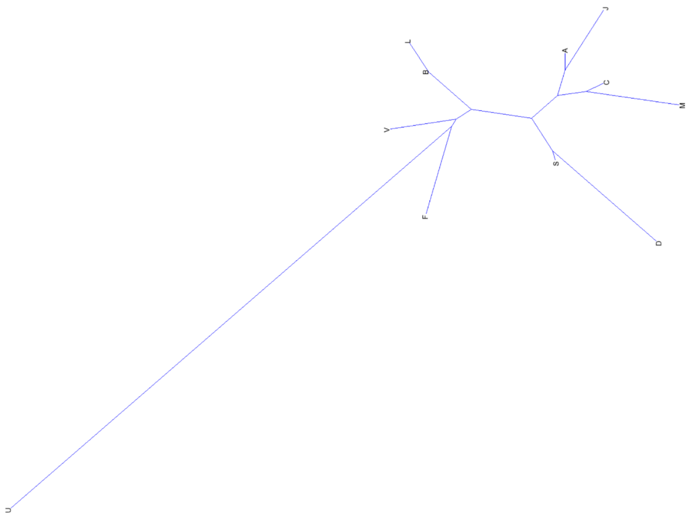
```
[david@david-laptop diff-Stemmatology-Python]$ perl lf_new4.pl test_data/parzival-all.txt
16
p1
p2          466
p3          445  693
p4          108  472  438
p5          560  523  784  589
p6          504  449  726  530  402
p7          314  593  218  333  688  615
p8          627  532  787  634  390  247  722
p9          318  415  439  300  634  450  254  588
p10         557  549  815  601  282  521  686  457  652
p11         596  583  778  613  398  371  666  268  529  411
p12         500  274  734  497  564  467  618  569  513  624  627
p13         414  241  636  409  503  394  497  475  358  536  508  202
p14         560  517  718  582  411  218  667  285  555  442  296  534  441
p15         504  246  779  495  610  588  645  668  482  618  677  235  260  617
p16         482  467  722  542  660  617  631  650  495  678  629  412  309  606  358
Executed the perl script in 3.1734
```

Vis. 5: Runtime and Distance Matrix for *Parzival* with *lf_new4.pl*

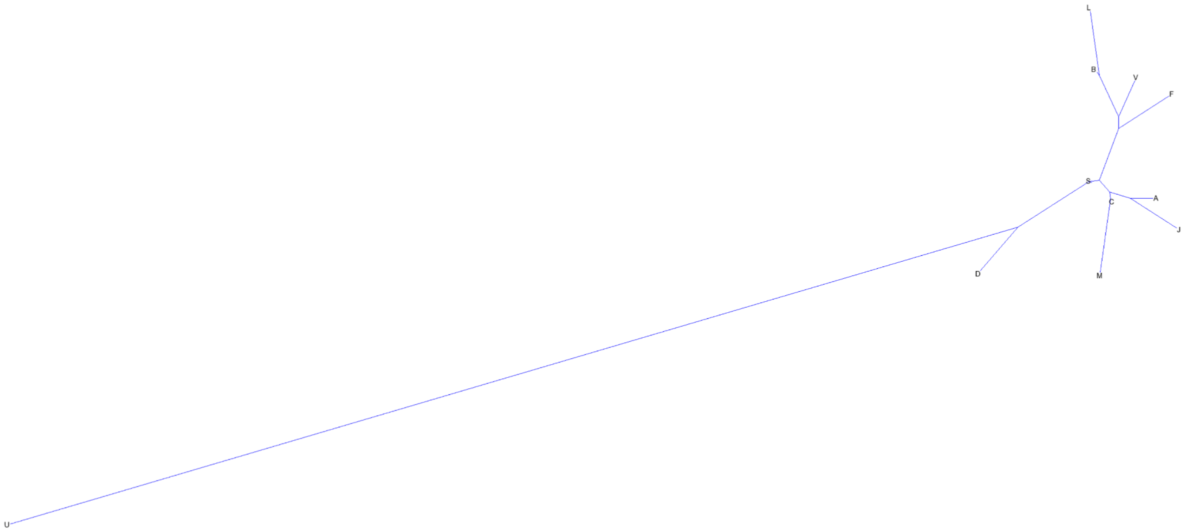


```
[david@david-laptop diff-Stemmatology-Python]$ python lf_new5.py -f test_data/parzival-aligned.csv -delim \t
16
p1
p2         435
p3         789  828
p4         233  436  762
p5         550  411  905  577
p6         395  332  854  482  397
p7         378  437  539  397  572  459
p8         558  399  891  587  356  299  556
p9         346  305  693  319  506  305  222  438
p10        587  502  980  648  355  536  613  429  577
p11        615  536  950  644  459  440  565  313  445  474
p12        479  246  806  474  415  336  439  391  341  528  546
p13        387  148  740  374  347  276  333  337  243  444  462  136
p14        482  399  843  521  396  249  494  338  400  505  413  383  303
p15        506  265  863  477  462  431  472  492  370  571  603  275  195  420
p16        708  625  1051  787  736  707  686  696  606  791  745  577  517  704  624
Executed the Python Script in 0.2133 seconds
```

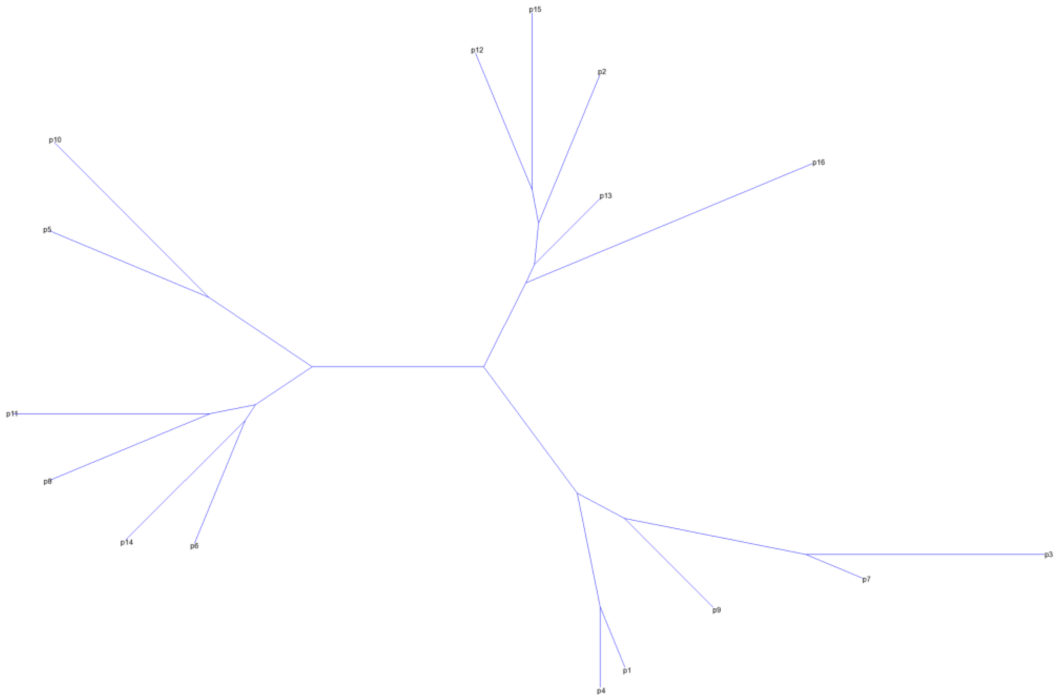
Vis. 6: Runtime and Distance Matrix for *Parzival* with *lf_new5.py*



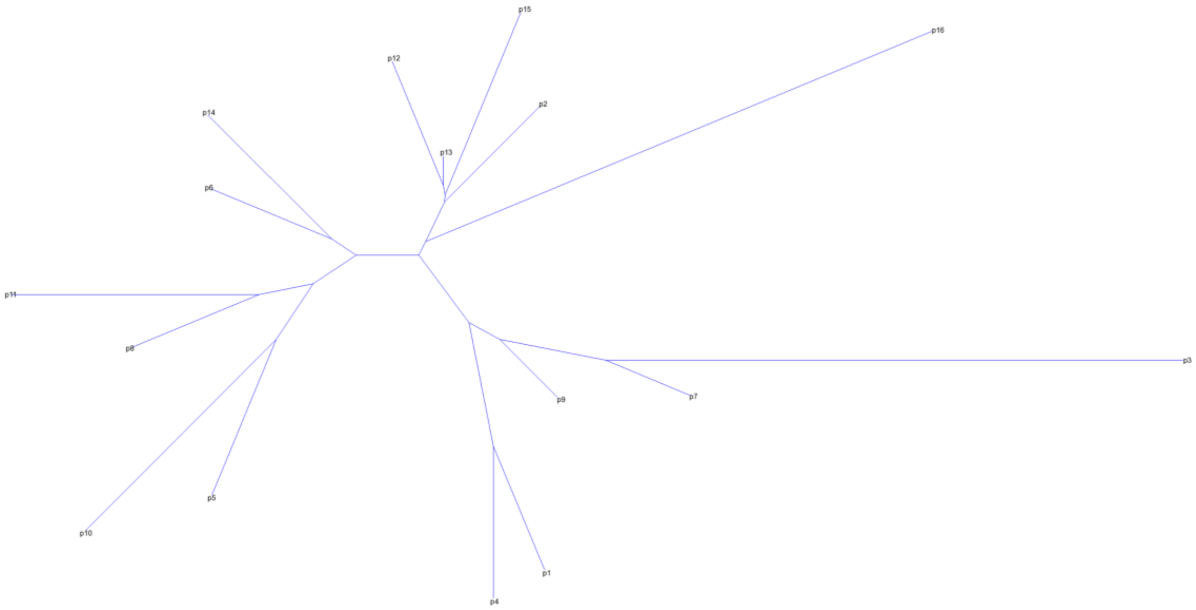
Vis. 7: PHYLIP Stemma for *Notre Besoin* with output from *lf_new4.pl*



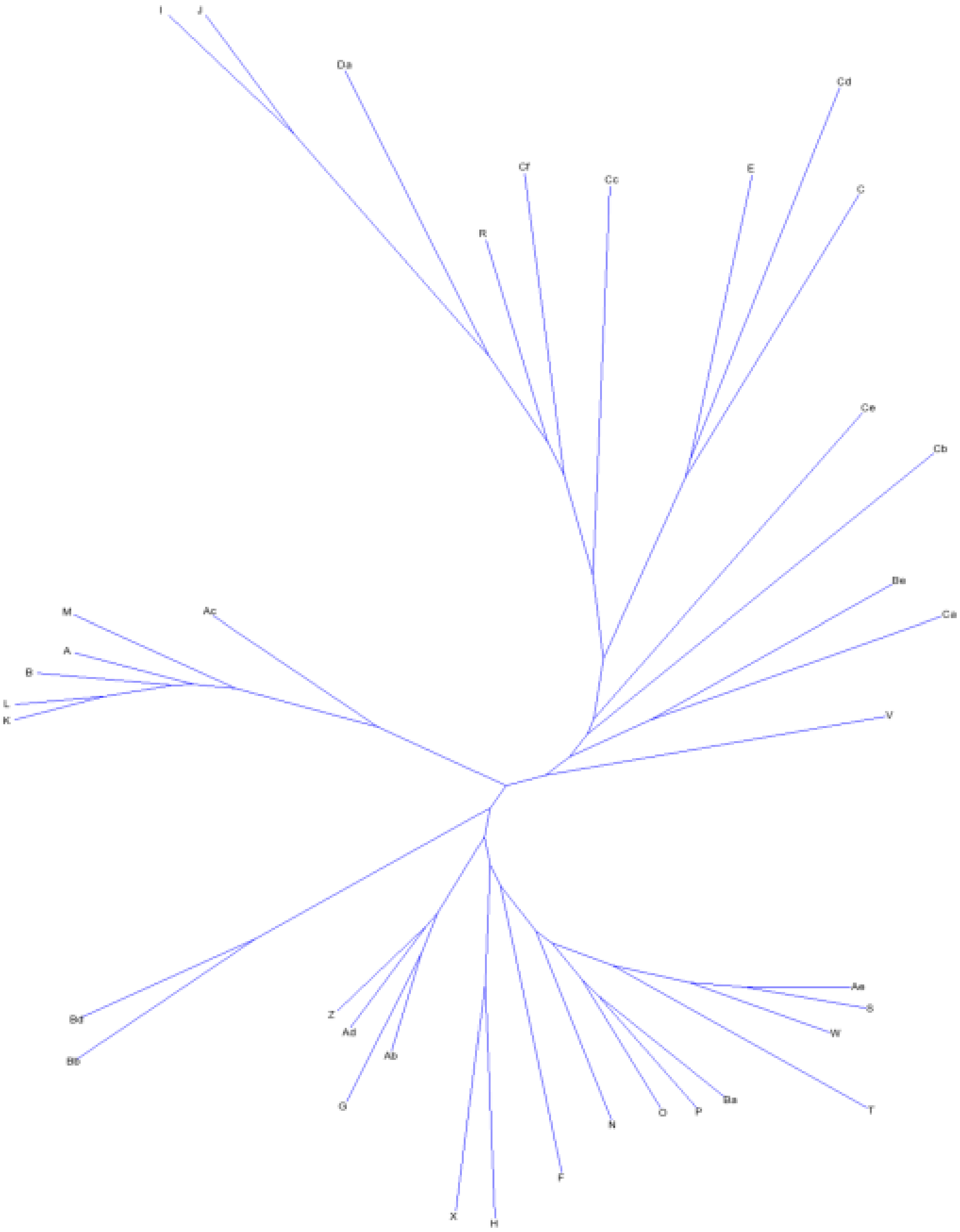
Vis. 8: PHYLIP Stemma for *Notre Besoin* with output from *lf_new5.py*



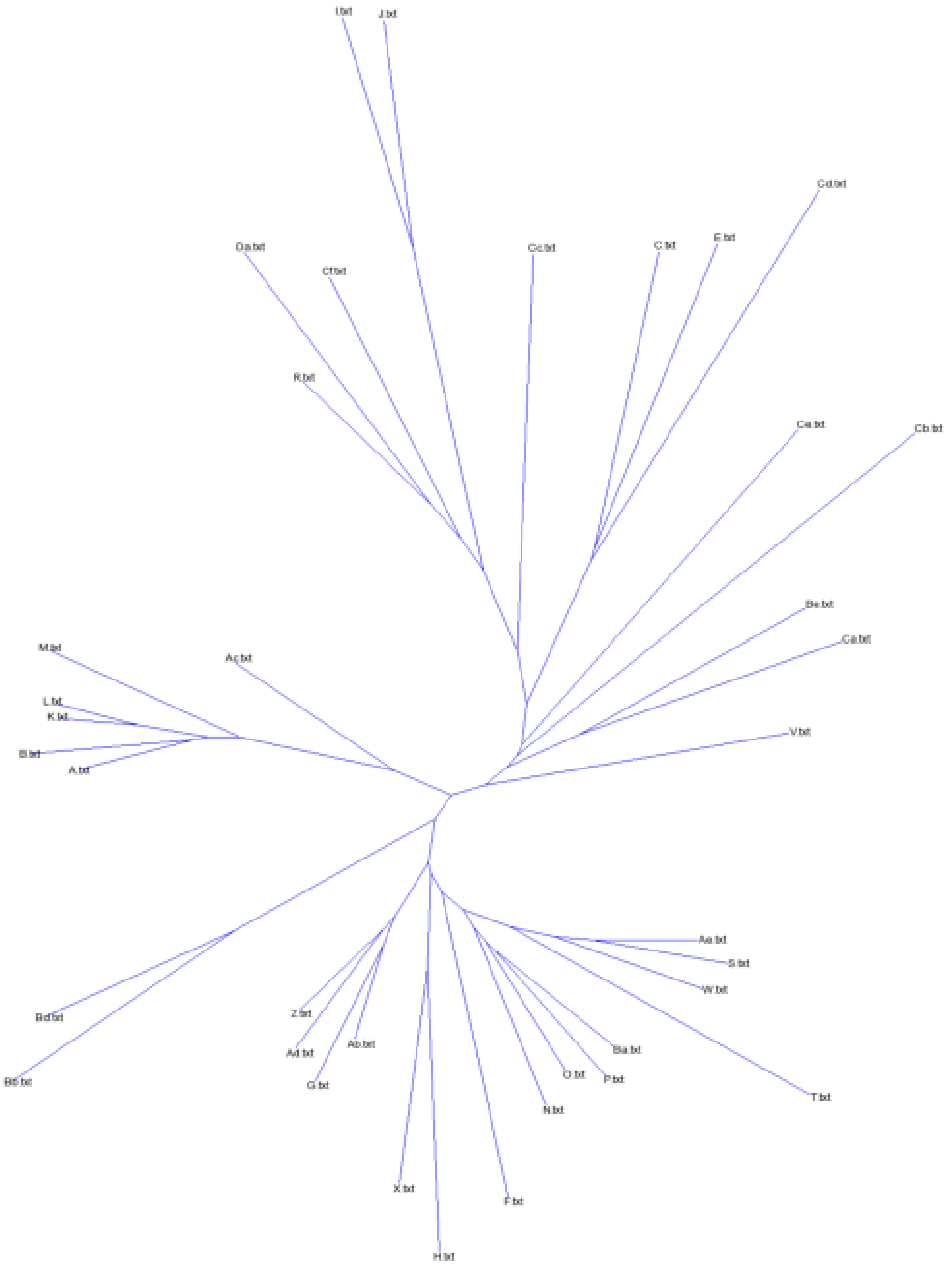
Vis. 9: PHYLIP Stemma for *Parzival* with output from *lf_new4.pl*



Vis. 10: PHYLIP Stemma for *Parzival* with output from *lf_new5.py*



Vis. 11: PHYLIP Stemma for *Heinrichi* with output from *lf_new4.pl*



Vis. 12: PHYLIP Stemma for *Heinrichi* with output from *lf_new5.py*

References

Baret et al., P. (2006). Testing Methods on an Artificially Created Textual Tradition. In *The Evolution of Texts. Confronting Stemmatological and Genetical Methods* (Vol. XXIV–XXV., pp. 255-283). Istituti Editoriali e Poligrafici Internazionali.

Christiansen, T., Foy, B., Wall, L., & Orwant, J. (2012). *Programming Perl: Unmatched Power for Text Processing and Scripting*. O'Reilly Media, Incorporated.

Computer-Assisted Stemmatology Challenge, https://www.cs.helsinki.fi/u/ttonteri/casc/data.html. Accessed 5 February 2023.

Howe et al., C. J. (2012). Responding to Criticisms of Phylogenetic Methods in Stemmatology. *Studies in English Literature 1500-1900*, *52*(1), 51-67.

Hunt, J. W., & McIlroy, M. D. (1976). An Algorithm for Differential File Comparison. *Bell Laboratories*, *41*, 1-8.

PerlPhrasebook. (2012, 4 26). PerlPhrasebook - Python Wiki. Retrieved 12 8, 2022, from https://wiki.python.org/moin/PerlPhrasebook

Roelli, P., & Bachmann, D. (2010). Towards Generating A Stemma Of Complicated Manuscript Traditions. Petrus Alfonsi's Dialogus. *Revue d'histoire des textes*, *5*, 307-321.

Trex-online, http://www.trex.uqam.ca/index.php?action=phylip. Accessed 5 February 2023.