OEIS Cluster Evaluation

Joshua Moss

Alex Teush

Submitted to Dr. Or Zuk as a final project in the course 52311 – Modern Statistical Data Analysis.

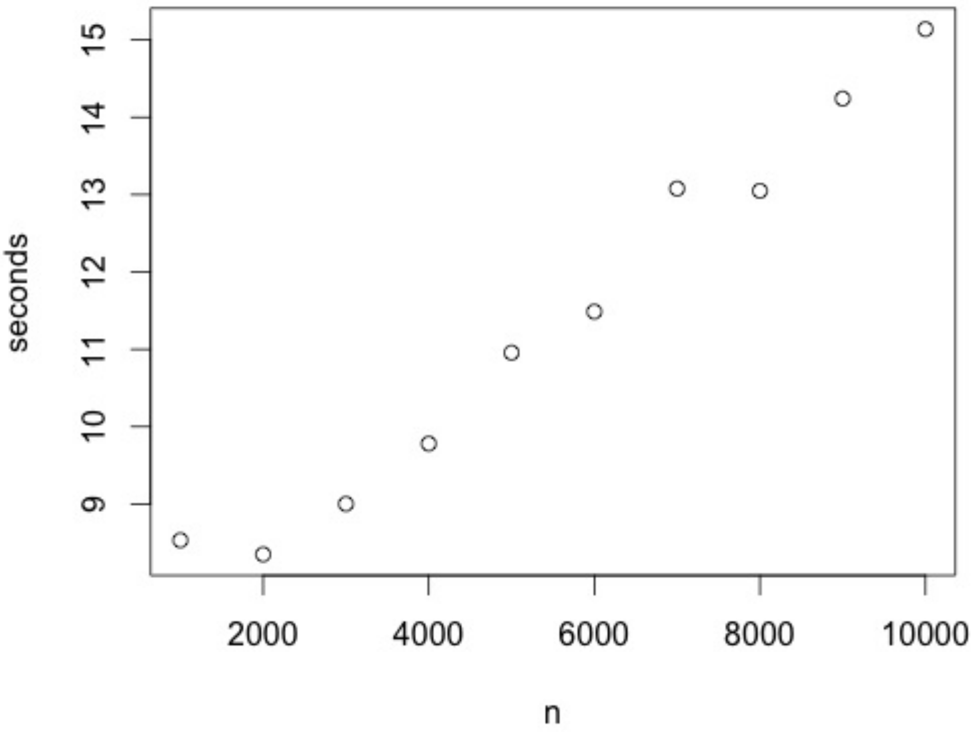
This is an analysis of the clustering made by us. We chose to cluster the OEIS' sequences using two methods: The k-means algorithm and a variant of hierarchical clustering.

Algorithm:

Hierarchical Clustering Algorithm:

We used a heuristic approach to hierarchical clustering. Given a list or matrix X with length n, a distance function and a hierarchical clustering approach (single, complete, average), in a first step, s random elements of X are clustered based on the distance function and the clustering method into k clusters. In the second step, the remaining elements are attributed to the original clusters using a voting method. For each element, up to v representatives are randomly selected from each cluster. The element is then compared to these representatives using the clustering method provided. The algorithm can be run using the function vote_clust in the attached R code.

The algorithm's complexity is O(n). The first step takes $s^2 \log s$ calculations (typical of hierarchical clustering algorithms). The remaining n-s sequences are voted up to kv times. Therefore – the whole number of moves is up to $s^2 \log s + (n-s)kv$. Since s, v and k are fixed (we used s=100, v=10), $\lim_{n\to\infty} (s^2 \log s + (n-s)v) = \lim_{n\to\infty} n \cdot v \Rightarrow O(n)$. the model's complexity can also be seen in the plot below, which was produced by running the algorithm with the parameters s=100,v=10, k=10 using single linkage and Euclidian distance on the OEIS sequences:



seconds
15
14
13
12
11
10
9
2000
4000
6000
8000
10000
n

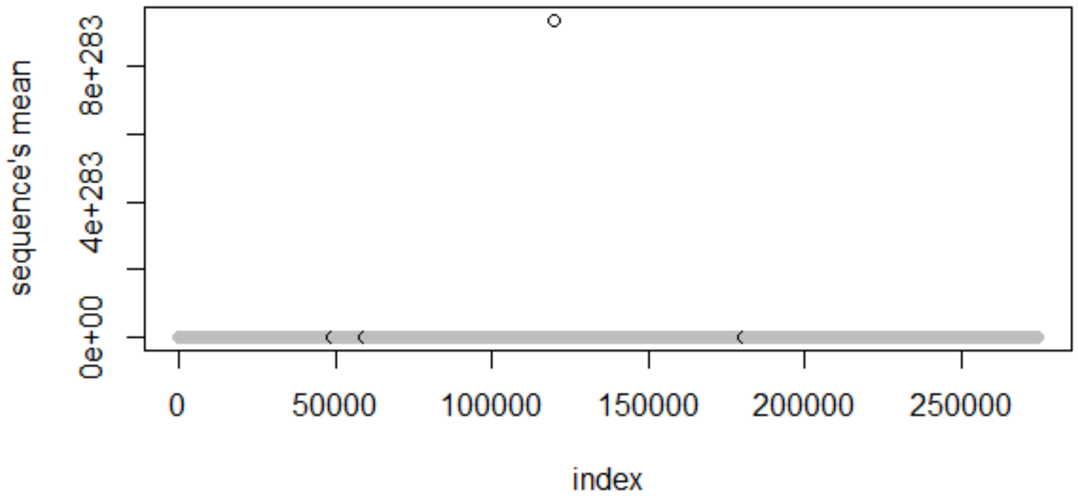
k-means algorithm:

The k-means algorithm was used on features, suggested by other classmates, which describe the sequences and create for each OEIS sequence a $\mathbb{R}^{20}$-vector. The features we've included were: series' index in the OEIS, sequences' mean, variance, median, length, skewness, kurtosis, log(abs(mean)), log(variance), number of sign changes, mean and median of sequences' differences, number of even elements, percentage of primes, squared and cubed numbers, sequences' maximum/sequences' mean, mean of the sequence modulo 2, 3 and 5. Further Explanations of the algorithm can be found in the R code attached.

The sequences were represented as a 272,544X20 order matrix (272,544 – Number of non-empty sequences in the OEIS stack). We operated the R function kmeans on it. The number of clusters was set by us to be 100, in order to compare our work to the work of other students.

The k-means process took 68 iterations.

It is sufficient to observe the clusters' sizes in order to appreciate that this is a very bad clustering. Among the 100 clusters, 98 contain from 1 to 41 sequences, one cluster contains 1258 sequences, and another one contains 272,259. Our Hypothesis was that this is due to the fact that some OEIS sequences have very large elements, which causes them to be very far from other sequences in a subset of the feature space. This can indeed be seen in the plot below:



The sequence with the maximal mean is A119555:

```
1.900000e+01  6.190000e+02  3.589900e+04  3.
301819e+06 4.685441e+284
```

Therefore, we chose to subtract the features Index, Mean, Variance, Median, Differences' Mean, Differences' Median and Sequence's length. We can see the difference in the clusters' sizes:

```
  [1]  6394  3318  2668  1362  1893  1127  1
498  2379   313  2286  2033  4631  2130
 [14]   136   465  1583   390  3511  2743
977  4469  1253  1311 13282  2948   800
 [27]  6846  2597  1871   682  4612  1584  5
218  6450  4796  2153  4200  3250  4762
 [40]  1499   691  5360   752  1140  3926  4
005   459  3985  1627  9289  1887  1347
 [53]  7484  1286  4709   826  3120   606  3
949   463  1700   375  3730  2177  3412
 [66]   901  4749  3721   788  3971   445  1
654  3794  1245  1107   947  5484  1731
 [79]  3688  1650  4545  4583  1583   489
722  3229  1057  1704  1231   812  3201
```

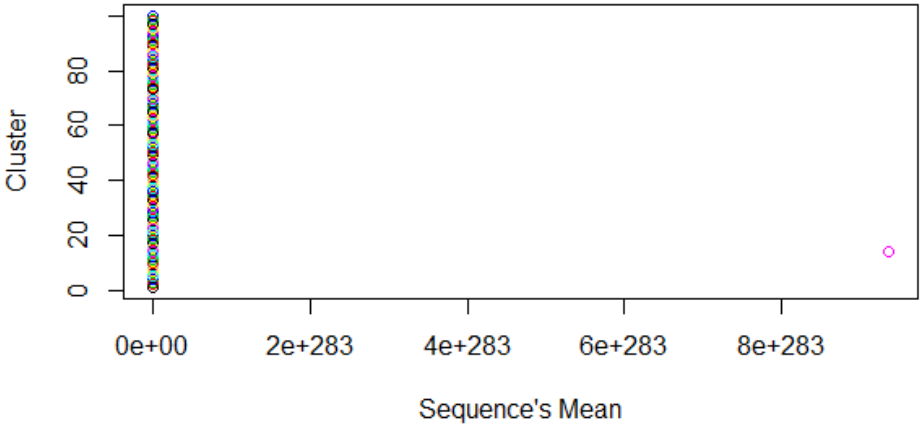
```
 [92]  3847   302  1589  2768  6799  4944  4
522  2752  1265
```

We chose to evaluate the model, without some of the features.

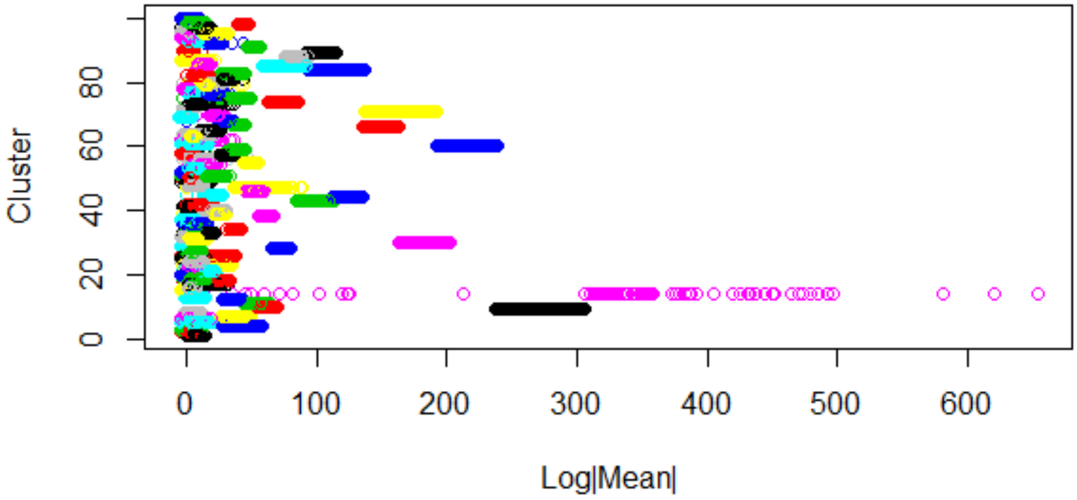
Features Examination

Our method will be to look at features and check whether different clusters assign different features' values. Mainly, we have looked if there's a 1:1 map between the feature's values and clusters. I.e – whether different feature's values go to different clusters. We've done this using the plot package in R. In our paper we present only notable features.

Plotting the sequences' mean against our clusters will give us:



It doesn't tell us much. Though, we can observe that when plotting Log|Mean| against the cluster, we found ourselves in-front of a map, that's nearly 1:1 when values exceed 100.



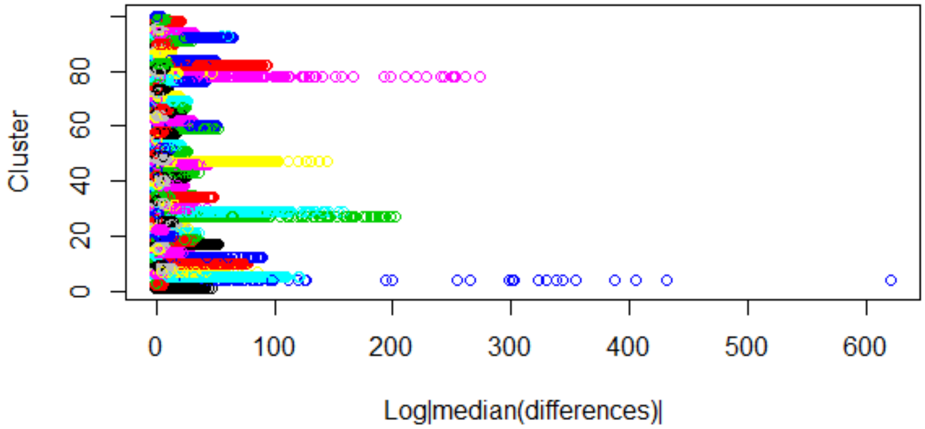
Since the log function is monotonous, the algorithm separates Mean and Variance values as well. The map between log|mean|, log|variance|(x ax) and clusters (y ax) is pretty much 1:1 where x-values larger than 100.

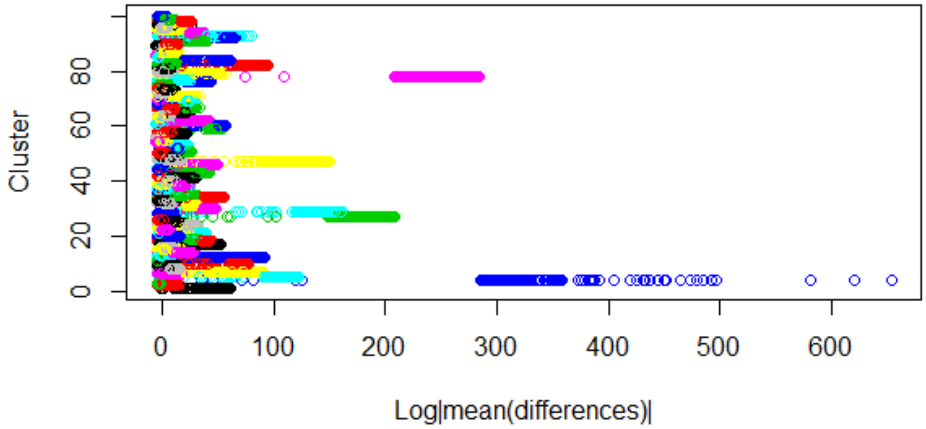
We can see a 1:1 map, to some level, when the x values are bigger than 50. Different log|mean|s go to different clusters.

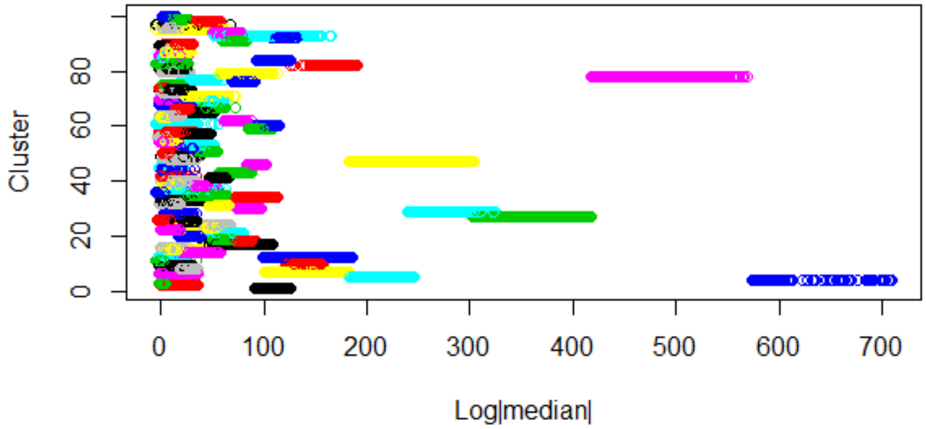
We conclude that the algorithm's clusters separates the Log|Mean| and Log|Variance| features, at least when log(mean), log(variance) > 100. Henceforth – the Mean and Variance features are separated as well.

Looking at the Differences' Mean or Differences' Median features (we observed the log of those features for the same reasons explained earlier):

--plots can be seen in the next page--


Cluster
0
20
40
60
80
0
100
200
300
400
500
600
Log|median(differences)|


Cluster
0
20
40
60
80
0
100
200
300
400
500
600
Log|mean(differences)|


Cluster
0
20
40
60
80
0
100
200
300
400
500
600
700
Log|median|

Other approaches we've used to evaluate our clusters were:

1. To check notable sequences' clusters and see whether other sequences in the same cluster are related.
2. To sample 1000 sequences and check the ratio between sequences linked by the OEIS and sequences appearing in the same cluster.

In the first method, we thought of several famous sequences and looked at other sequences in the same clusters. We sampled ten sequences from each cluster, then went to OEIS.com and looked which values that sequence included. The sequences we have observed were the prime numbers (A000040), perfect numbers (A000396), Fibonacci numbers (A000045), the zero sequence (A000004), and Catalan numbers (A000108).
In most cases, we didn't find a link between the sequences in the same cluster, and between the sequences we have observed. This can be due to the number of clusters we have used. By average, the number of sequences in each cluster was about 27,200. Therefore, it is likely that every cluster contained many un-related sequences.

Applying method 2, we sampled 1000 sequences from the OEIS stack. For every sequence we used the functions getUrl, and getLinks, in order to mine all the sequences linked to it in the sequence's OEIS page (all the links that are "A" followed by a number) and counted how many linked sequences are in the same cluster. In addition, we computed the ratio between the linked sequences and the cluster's size. This was done since one could not cluster all the sequences into one cluster and offer the model as a 100% matching percentage model. We also compared the matching percentage between sequences in context – 5 to 8 sequences that appear in every sequence's OEIS page – and sequences that are in the same cluster. This was a more difficult task, due to the fact that whereas the number of linked sequences tends to be about 14-18, the number of sequences in context is about 5-7, and so by nature, there will be fewer matches.

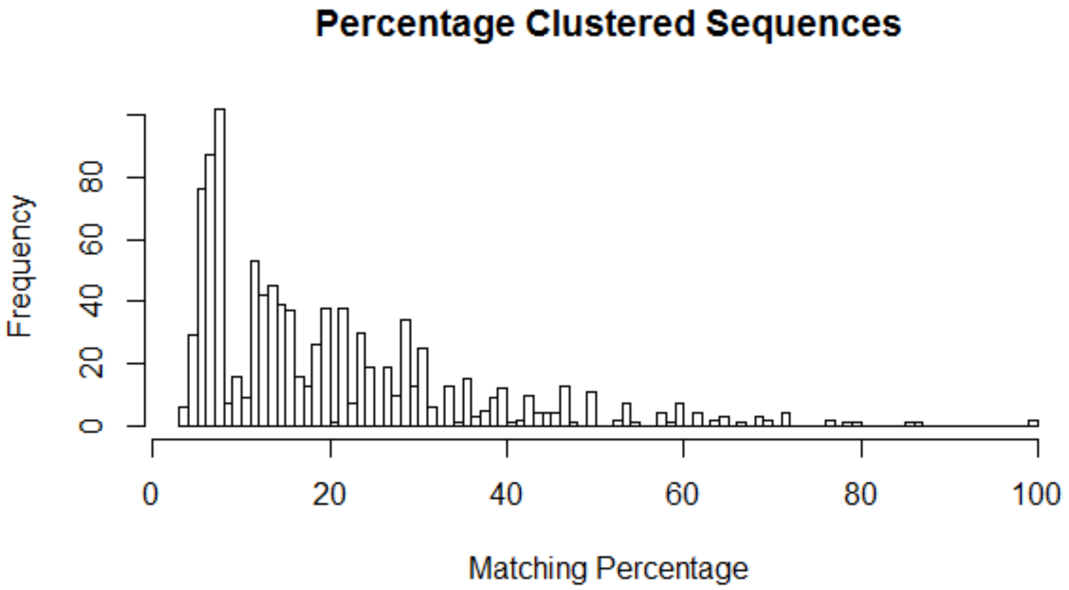
We used the testing process in four models:

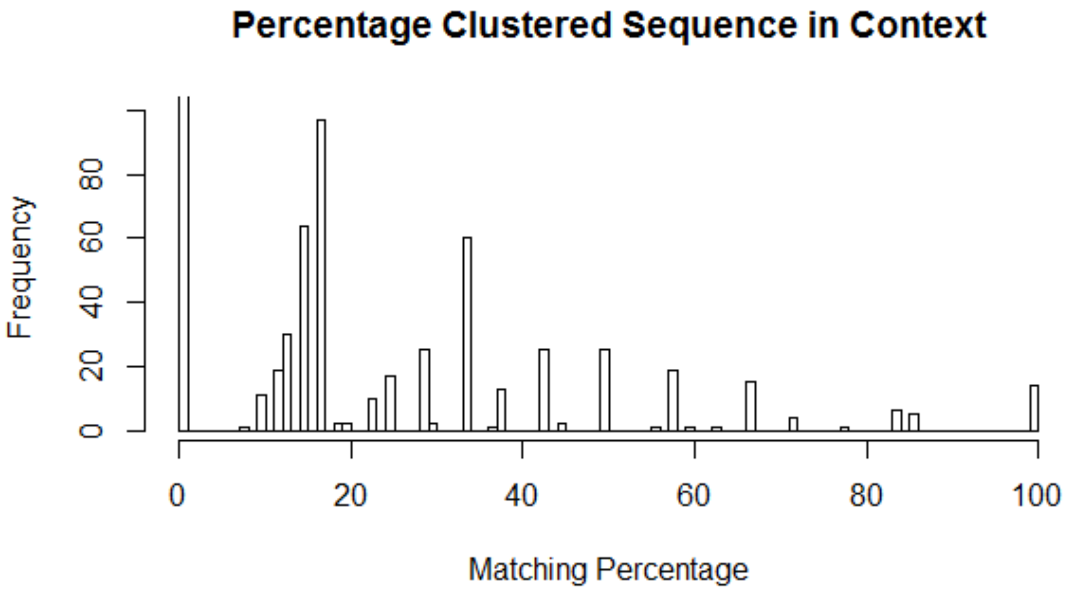
1. The k-means based clustering model with the features as input (k=100)
2. The hierarchical clustering based model with the features as input (k=100,v=1,s=10000,single linkage)
3. The labeling found in the repository
4. A random model

For every model we produced a histogram of the matching percentage between linked and clustered sequences, and between sequences in context and clustered sequences.

The histograms can be seen in the next pages. They were made by the r package hist. We chose by default to set the “breaks” argument to 100. In the sequences in context histograms, the y-axis was set to range between 0 and 100, because the majority of sequences’ matching percentage was zero in all models, whereas the true number of elements without sequences in context in the same cluster can be found in the tables attached.

The k-means algorithm:

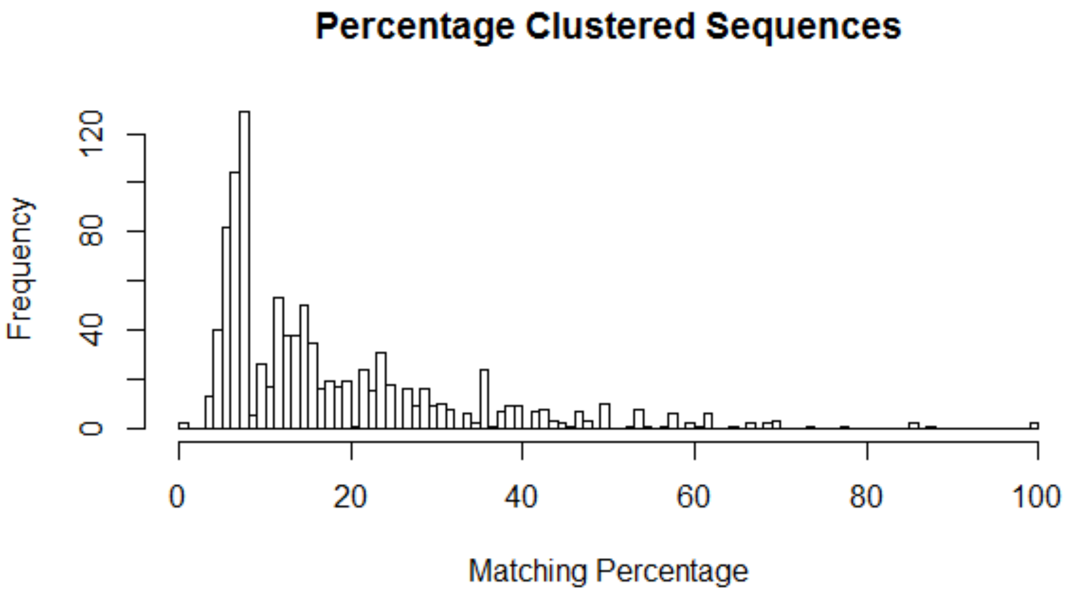


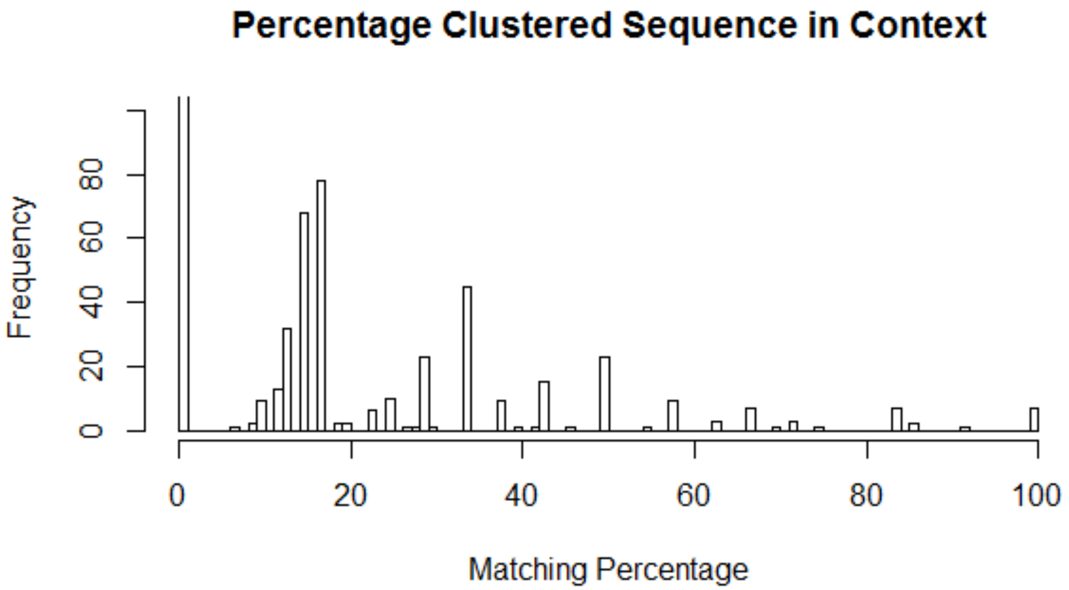


	Linked Sequences Matching Percentage	Sequence in Context Matching Percentage
Mean	19.67507	14.62027
Variance	235.6952	456.8905
Median	14.28571	0
Mode	7	0

Elements without sequences in context in the same cluster	527
Mean pctage between the linked sequences and cluster's size	0.1007042
Median pctage between linked sequences and cluster's size	0.07411067

Hierarchical Clustering model:

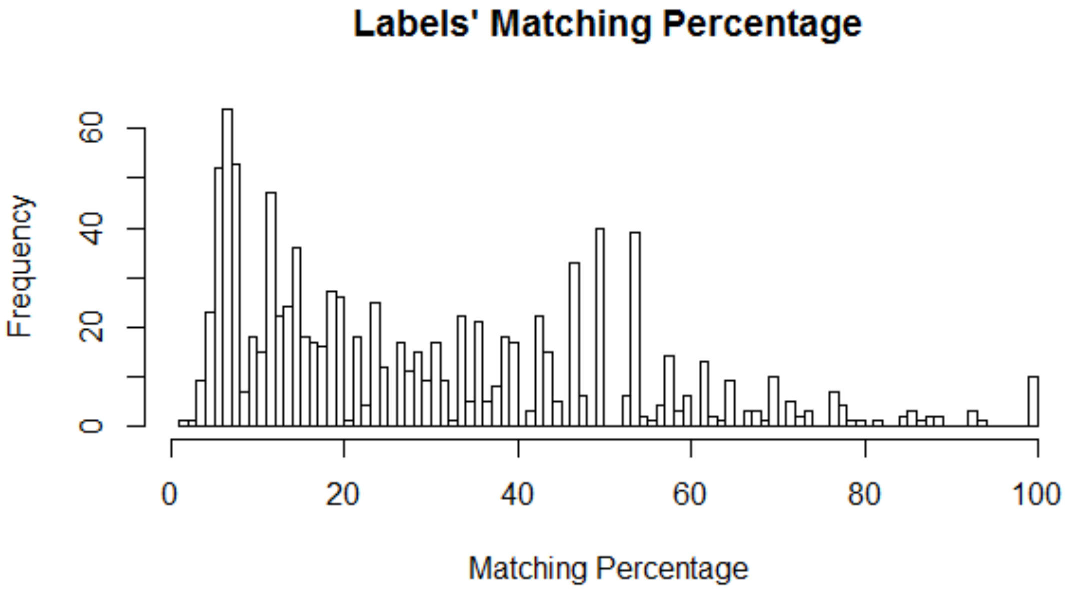


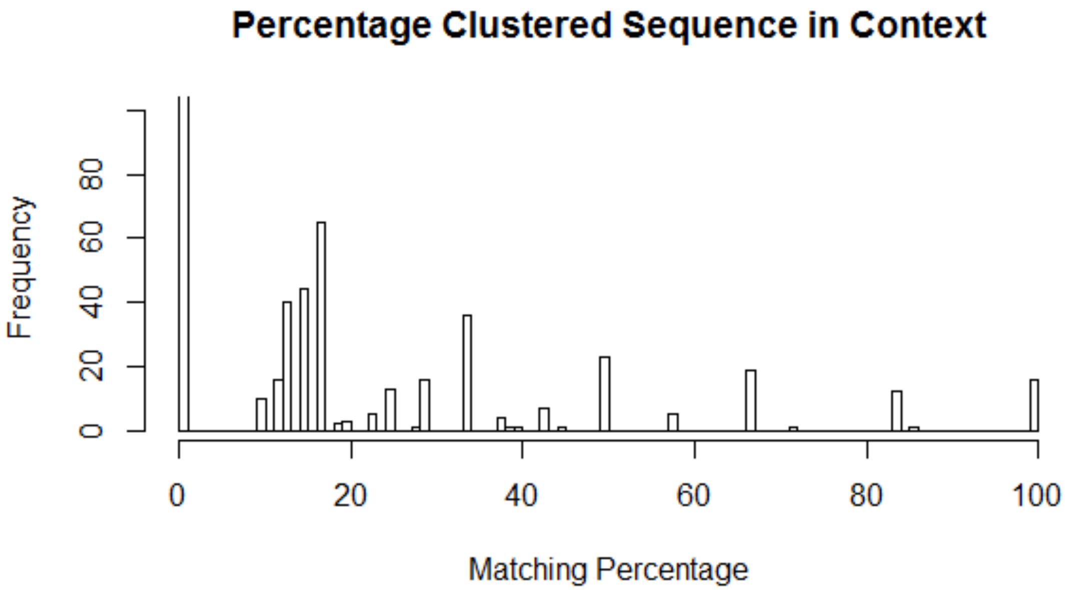


	Linked Sequences Matching Percentage	Sequence in Context Matching Percentage
Mean	17.57898	11.01527
Variance	216.0431	349.8747
Median	12.5	0
Mode	7	0

Elements without sequences in context in the same cluster	614
Mean pctage between the linked sequences and cluster’s size	0.08395844
Median pctage between linked sequences and cluster’s size	0.05668934

Labels model:

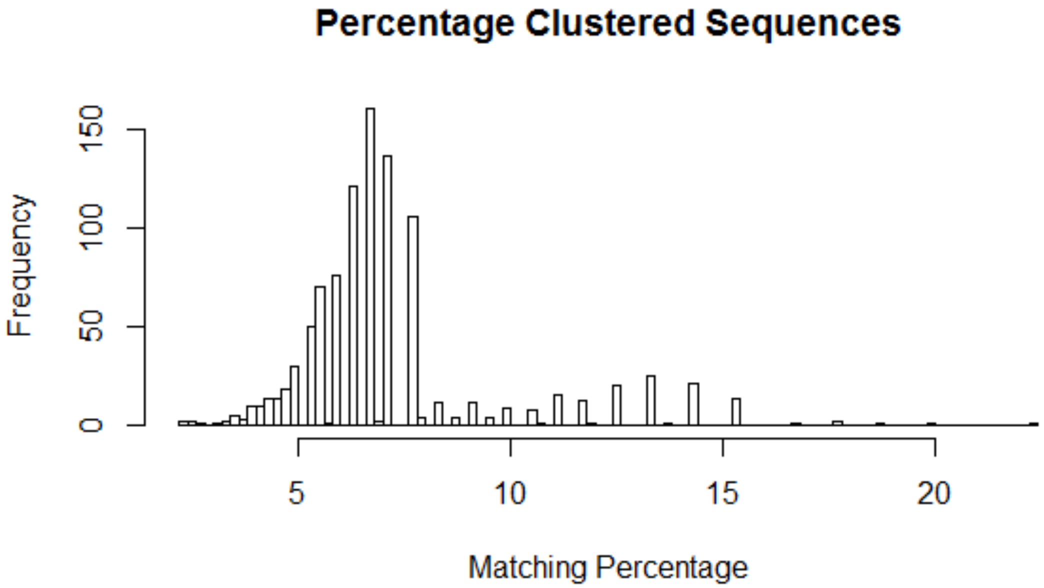


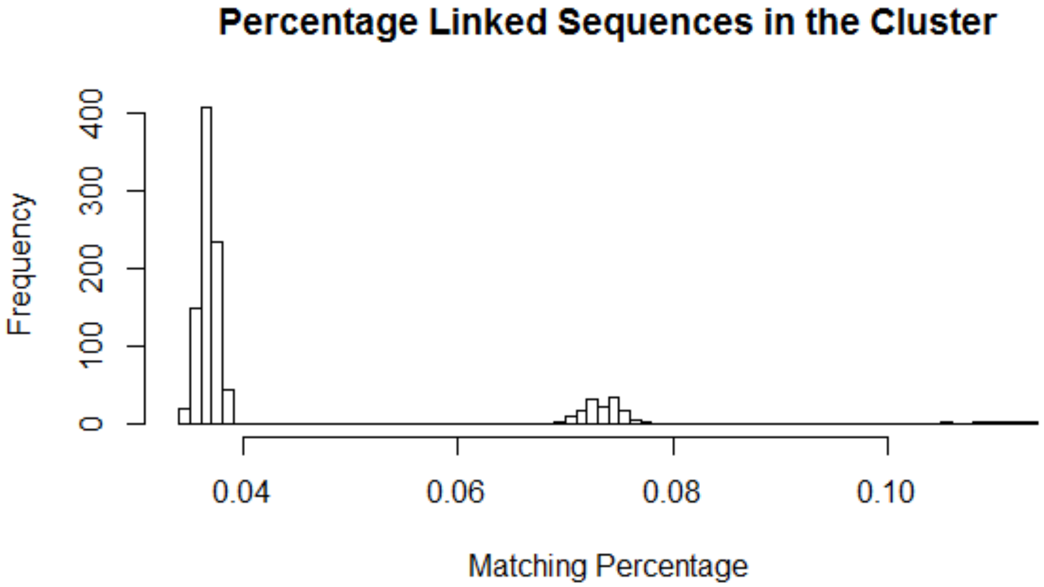


	Linked Sequences Matching Percentage	Sequence in Context Matching Percentage
Mean	29.00535	10.73718
Variance	216.0431	425.2142
Median	23.07692	0
Mode	7	0

Elements without sequences in context in the same cluster	658
Mean pctage between the linked sequences and cluster’s size	0.164526
Median pctage between linked sequences and cluster’s size	0.1035599

Random Model:





	Linked Sequences Matching Percentage	Sequence in Context Matching Percentage
Mean	7.236008	0.8324
Variance	6.673586	12.09566
Median	6.666667	0
Mode	6	0

Elements without sequences in context in the same cluster	944
Mean pctage between the linked sequences and cluster's size	0.04240099
Median pctage between linked sequences and cluster's size	0.03675119

Further Anaysis of matching between clusters and linked sequences:

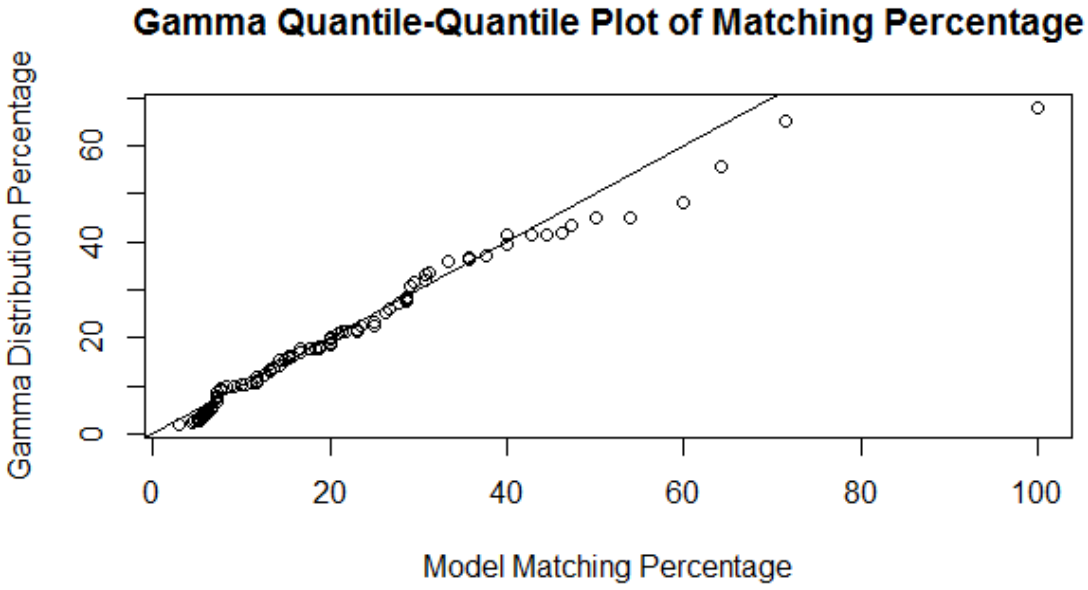
Both k-means and hierarchical clustering histograms are reminiscent of Gamma distribution, with “breaking points” in which the histogram get relatively low values, compared to it’s neigbours. That’s more significant in the crossrefs histograms. We assumed that this is happening due to the fact that the matching property has to be a rational number, in which the denominator can receive only few values (in “sequences in context” it’s bounded by 5 and 8). In order to estimate the distribution of number of lined sequences to get in the cluster, we chosed to use the method of moments estimation:

According to that method, we’ll set X=number of clustered linked sequences.

$X \sim \Gamma(\alpha, \lambda)$. Hence:

$$\bar{X}_{1000} = \frac{\alpha}{\lambda}, \frac{1}{1000}\sum_{i=1}^{1000}(X_i - \bar{X})^2 = \frac{\alpha}{\lambda^2} \Rightarrow \alpha = 1.643, \lambda = 0.0835$$

If we’ll compare the histogram against Gamma(1.643,0.0835) It’ll give us:



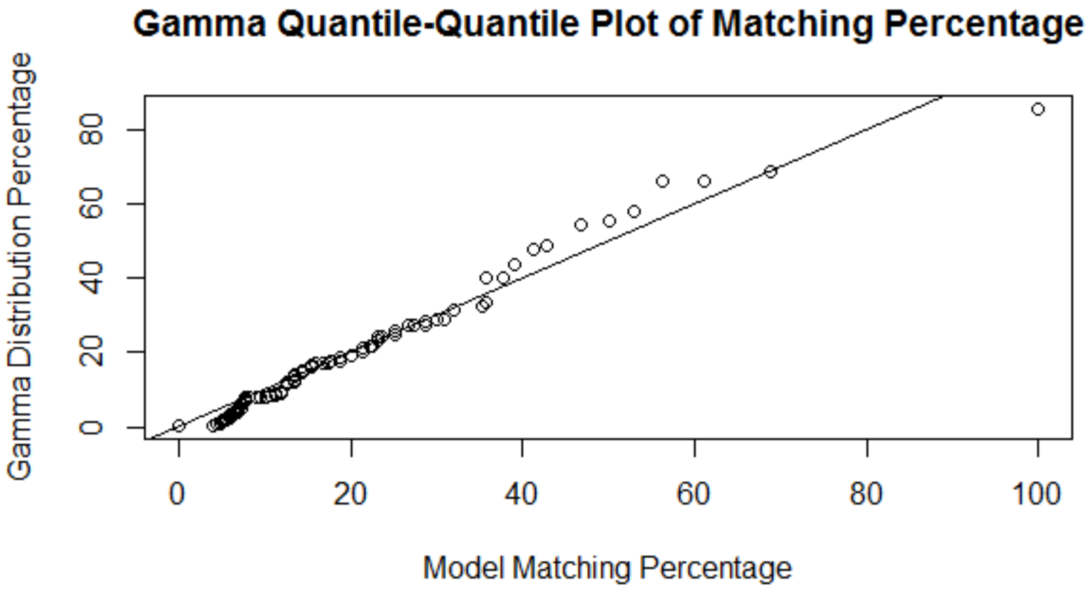
Using the same method for the hierarchical clustering will give us:

$$\bar{X}_{1000} = \frac{\alpha}{\lambda} = 17.58;$$

$$\frac{1}{1000}\sum_{i=1}^{1000}(X_i - \bar{X})^2 = \frac{\alpha}{\lambda^2}$$

$$\Rightarrow \alpha = 1.43, \lambda = 0.0814$$

The qq-plot can be found on the next page.



In conclusion, we see that there is a notable difference between the random model and all the other models. The labels model gives higher matching percentages between clustered sequences and linked ssequences, whereas when comparing the matching between clustered and sequences in context, all models give about the same results, with a slight advantage to the k-means clustering (14.62%) over the others. Given a sequence, all models not necessarily match sequences in context or linked sequences to the sequence's cluster. however, they do tend to group the linked and/or sequences in context in relatively few clusters. The main problem with the models, is the clusters being too large, henceforth incomparable to the OEIS' links and crossrefs. It appears to be that 100 is a too-small number for the OEIS clustering.

On the other hand, 100-clusters models proved to be more efficient in comparison to other models we tried. When we set the k to be 10,000 (that should be an average of 27.2 sequences in each clusters. Not so far from the average number of linked sequences – 16) the mean of matching percentage between linked and clustered sequences was 8.81, whereas the matching percentage between clustered sequences and crossrefs was 2.97.

The mean proportion between the linked sequences that also appeared in the sequence’s cluster and the cluster’s size was 5.2. that’s by far better the the 100-means mean of proportion – 0.1.

Interesting Clusters and Sequences:

After seeing that 100-means clustering was too general, meaning that a cluster containing about 27,200 will have many sequences with a weak linkage between them, we thought of looking at models of 10,000 clusters. Another motivation was that many of our 10000-means clusters are an inclusion of our hierarchical clustering and 100-means clusters, or their intersections. So by evaluating them we can save space and time, yet still evaluate our original models clusters. This analysis was performed using the k-means algorithm.

Interesting examples that we have found:

A000796   Decimal expansion of Pi (or, digits of Pi).

```
This sequence clustered with 56 sequences:

796  10503  14565  73017  73732  78890  8584
9  86036  86056  93204  93731
94090  94961 102521 104541 104689 111770 113
399 117016 117234 133613 133842
134972 143440 144810 157214 157294 159467 16
4293 174815 174968 175478 175639
176103 176104 176323 176403 178647 179450 18
5579 195059 197723 201130 202473
240976 244648 246768 247318 247718 254979 25
6166 256853 258405 259149 261345
261508
```

All of those sequences are decimal expansions of numbers, constants, limits etc. for example:

$$\frac{1}{\sqrt{2}}, w = \lim_{n\to\infty} n\phi - \sum_{k=1}^{n}\frac{F(k+1)}{F(k)}, \sum_{n=0}^{\infty}{\frac{1}{9}}^{2^n},$$

The only exception is - A261345 (Number of distinct prime divisors among the numbers k^2 + 1 for k in 1 <= k <= n). That’s a surprising result. We didn’t give the sequence’s name as a feature.

We suggest that it may be because all decimal expansions consist of only elements from 1 to 9.

We wondered whether there would be much difference upon evaluating the binary expansion of Pi (A004601). It lead us to many un-related sequences. Such as:

Consider the last letter of each of the English words zero, one, two, three, four, five, ... Write down 0 for a vowel or "y", 1 for a consonant. (A059437)

A000004 The zero sequence.

```
This sequence clustered with 66 sequences:
4      12    7395   10692   10701   10709   10716
10722  10727  10731  10734
10850  10851  10852  10853  10854  10855  10
856  10857  10858  10859  10860
10861  10862  10863  10864  10865  10866  10
867  10868  10869  10870  10871
37017  58445  58446  72288  76337 115453 118
329 121977 122036 144134 174817
175274 245206
```

This cluster contains almost only constant sequences of elements (or, relatively large prime numbers, for which the differences between the elements are very small. This includes one-element sequences like-

Squares composed of digits {0,5,6}, not ending with zero.(A059446), that contained the sole element 5000060065066660656065066555556.

This result is a not such bad one. The OEIS returns only the first three results under “crossrefs”.

A000045 Fibonacci numbers: F(n) = F(n-1) + F(n-2) with F(0) = 0 and F(1) = 1.

```
This sequence clustered with 30 sequences:
45   1351   2965   4691   10029   10752   10754
10757  13986  14291  45794
50192  52284  52943  58354  65678  77419  83
198  95354 108906 113166 117760
121653 135701 157894 165407 167808 187070 23
3525 234368
```

Despite it’s being one of the most famous sequences, the clustering of the fibonnaci numbers wasn’t that good. A10752, A10754, A10757 are sequences related to triangles.

A001351 are Associated Merssene numbers (not to be confused with Mersenne primes). We haven’t found a link between them and Fibonnaci numbers.

Other interesting sequence clustered with fibonacci numbers are:

A002965 - Interleave denominators (A000129) and numerators (A001333) of convergents to sqrt(2) – It makes sense that a sequence $(a_n)_{n=1}^{\infty}$ in which $\frac{a_{n+1}}{a_n} \to 1.41$ should be clustered with a sequence $(a_n)_{n=1}^{\infty}$ such that $\frac{a_{n+1}}{a_n} \to 1.61$.
A014291 - Imaginary Rabbits: imaginary part of a(0)=I; a(1)=-I; a(n)=a(n-1)+I*a(n-2), where $I = \sqrt{-1}$ - It’s reminiscent of the original context in which the sequence was introduced – as a way to describe the reproduction of immortal rabbits.

A095354 – Number of primes p such that $Fib(n+1) \leq p \leq Fib(n+2)$.

A167808 –Numerator of x(n)=x(n-1)+x(n-2), x(0)=0, x(1)=1/2.

Some sequences were expansions of rational functions:

A052943 $\left(\frac{1-x^2}{1-2x^2-x^3+x^5}\right)$, A117760 $\left(\frac{1}{1-\mathrm{x}-\mathrm{x}^3-\mathrm{x}^5-\mathrm{x}^7}\right)$, A165407 $\left(\frac{1}{1-\mathrm{x}-\mathrm{x}^3 * \mathrm{c}(\mathrm{x}^3)}\right)$
c(x) – generating function of Catalan numbers.

There is a likeness between those expansions and the Fibonacci sequence. Foe example –

```
1, 1, 1, 2, 3, 5, 8, 13, 21, 33, 53, 85,
136, 218, 349, 559, 895, 1433, 2295, 3675,
5885, 9424
```

Some clustered sequences are defined by recursion (for example A050192, A233525).

Another intersting sequence in the cluster was A234368 - Floor(AGM(1, Fibonacci(n))), where AGM denotes the arithmetic-geometric mean.

A119555 Primes in the sequence f(n) = f(n-1)+((-1)^n)*n!, with f(0)=0.

This is the OEIS sequence with the largest elements.

It clusters with 11 sequences:

19437 104536 114784 119555 120850 139120 144957 145572 162591 172145 173058

Other sequences in the cluster are recursive sequences involving factorials (A019437)
or sequences (often recursive ones) of very big primes (A104536, A114784, A120850, 139120, 144957, A162591).

A000040 The prime numbers.

This sequence clusters with 96 sequences:
40 25584 38614 38616 38618 40161 42966 49543 49545 49549 49551
49555 49561 49569 49573 49585 50260 50757 51701 51860 52085 57447
57448 58853 63884 63904 68863 70159 76805 77359 79152 80191 82011
82646 84331 85400 85402 86472 86498 86518 87685 91265 94516 94744
94746 94751 100725 100726 101044 101595 102348 106118 107801 107802 107803
107804 107805 107806 107807 107808 107809 107810 107811 107812 107813 107814
108546 113029 115232 118753 119615 119993 127566 129543 137458 152076 161929
165671 167773 169647 176162 176164 176165 178209 216437 216883 216884 216885
216886 238242 240960 244862 258429 262694 265750 265757

Notable sequences:

A38614,38616,38618,76905 – primes not containing the digits 6,8,0 and the number 13, respectively.
A40161, 42966, 49543, 49545, 49549, 49551, 49555, 49561, 49569, 49573, 49585, 58853, 216883, 216884, 216885, 216886 – primes p such that x^q=2 has a solution mod p, for various primes q.

A surprising result is the sequence A070159 - Numbers n such that phi(n)/(sigma(n)-n) is an integer. All most all the elements are prime numbers.

A107801-107814 - a(1)=prime(q), for n>=2 a(n) = smallest prime not previously used which contains a digit from a(n-1) for different primes q.

A176162,A176164,A176165 - Primes p such that (p-2)/q is not a prime number for q=5,7,11.

A238242 - Primes p such that p^2+p+41 is also prime – A famous sequence.

We can assume from the prime numbers example that large clusters consist of several groups of very similar sequences.

A250000 Peaceable coexisting armies of queens: the maximum number m such that m white queens and m black queens can coexist on an n X n chessboard without attacking each other.

```
This sequence clusters with 52 sequences:
840   6250   8840  10361  10671  27673  3052
9  45504  46708  47815  51602  51758  52437
52442  52443  57241  71536  76838  87778  89
794  89891  89892  94445  96219  96340  9811
9
98403  98472  99438 106169 111259 114248 119
602 123849 132346 135515 140837 153802 16174
6
185306 211179 218951 218971 239104 242737 24
3274 243555 243562 243786 250000 253569 2597
90
263161 266739 267484 268546 268547
```

A somewhat exotic sequence. The algorithm returns interesting results:

A51758 - Consider problem of placing A051755(n) queens on an n X n board so that each queen attacks precisely 2 others. Sequence gives number of solutions up to square symmetry.
Where- A051755(n): Consider problem of placing N queens on an n X n board so that each queen attacks precisely 2 others. Sequence gives maximal number of queens.

A087778 - Decimal expansion of Avogadro's constant.

Many results did were from the field of graph theory. Including:

A243786, 243562, 243555, 243274, 211179, 185306, 161746, 135515, 132346 and many more.

In addition, we have looked at the largest cluster:

The largest cluster contains 197 sequences:

```
1241    4392    4393    4994    9976    9978  305
31  35832  35833  35834  35835  49394  53729
55476  60917  62143  62152  62263  63817  67
427  68204  75909  77231  82022  89274  9037
3
97192 101632 107523 107524 107561 107562 111
598 111780 112485 113921 131521 132869 14090
6
141008 141010 141011 154308 162830 163177 16
3187 163526 163548 163995 164025 164639 1646
64
164964 164970 165369 165456 165973 165979 16
5980 166420 166421 166422 166613 166614 1666
15
166903 167079 167080 167081 167225 167226 16
7235 167697 167698 167699 167941 167942 1679
43
168703 168704 168705 168751 168752 168753 16
8799 168800 168801 168847 168848 168849 1688
95
168896 168897 168943 168944 168945 168991 16
8992 168993 169039 169040 169041 169087 1690
88
```

```
169089 169135 169136 169137 169183 169184 16
9185 169231 169232 169233 169279 169280 1692
81
169327 169328 169329 169375 169376 169377 16
9423 169424 169425 169471 169472 169473 1695
19
169520 169521 169567 169568 169569 170035 17
0036 170037 170083 170084 170085 170131 1701
32
170133 170179 170180 170181 170227 170228 17
0229 170275 170276 170277 170323 170324 1703
25
170371 170372 170373 170419 170420 170421 17
0467 170468 170469 170515 170516 170517 1705
63
170564 170565 170611 170612 170613 170659 17
0660 170661 170707 170708 170709 180585 1832
42
186547 197088 203283 221339 223073 230803 23
0836 230897 231053 231243 235340 263433 2642
80
268884 269015
```

Looking at sequence’s indexes, we can see that the cluster contains many different groups of sequences, as we have seen observing the prime numbers overview (in some extent, also in the Fibonacci sequence). A4292, 4293, 4294 are expansions of the rational functions $\frac{(1+x)^2}{1-18x+x^2}, \frac{1+2x+x^2}{1-26x+x^2}, \frac{1+2x+x^2}{1-34x+x^2}$ respectively. A9976, 9978 are powers of 32 and 34. A35832-35835 are Coordination sequence for lattice D*_m (with edges defined by l_1 norm = 1) for m=92, 94, 96, 98, 100.

We can see the pattern of mapping many groups of connected sequences to one cluster again and again throughout the clustered sequences. Almost every line has at least a couple of successive indexes, that often (though not always) indicate a connection between sequences. It is worth mentioning that when we run the algorithm we omitted the index feature, out of fear that the algorithm would not cluster together linked sequences due to indexing.

It is important to comment, that many of those sets of sequences in the cluster, are only a part of a larger group. You can take for example the A35832-35835 above. It is a subset of a larger sequence family taking from A35797 (Coordination sequence for lattice D*_24 (with edges defined by l_1 norm = 1)) to A35835 (Coordination sequence for lattice D*_100 (with edges defined by l_1 norm = 1))). We have seen the same thing dealing the prime numbers – many sequences were from the family of primes p such that x^q=2 has a solution mod p, for various primes q. We gave 16 examples of it, whereas in reality the OEIS contains more than a hundred sequences relating to that family (the elements from A49543 to A49596 are sequences of the type, and it's only a fraction of all such sequences). Although it makes sense that those sequences would be in the same cluster with the sequence of all primes without any characteristic, the algorithm in some sense leaves us with "gaps and holes". A solution we can offer, is to compare the 10,000-means model to models of lesser clusters, and see whether other related sequences can be found.

Another attempt made by us is to observe the clusters density, where cluster's density is the sum of squared distances between the elements to the cluster's center. The kmeans function returns that value as 'withinss'. Unfortunately, that did not give us much. It's more affected by the size of elements. The command `which.max(results4$withinss)` returns 8268 – the second sequence we have observed. In spite of the the cluster being sparse, there's a strong similarity between the results.