Using genomic representations and machine learning algorithms for predicting structural information about gene clusters' products

Author: Victor Rodriguez Pastor

2425005R

Supervisor: Simon Rogers

College of Medical, Veterinary & Life Sciences

A report submitted in partial fulfilment of the requirements for the

MSc Bioinformatics Degree at The University of Glasgow

19th August 2019

Content

1. Abstract

Biosynthetic gene clusters are groups of genes found together in the genome that biosynthesise secondary metabolites which can be used for a wide range of applications, especially in the field of medicine: immunosuppressants, antibiotics, etc. With the recent development of computational power and bioinformatics tools, new gene clusters in bacteria, fungi and plants have been discovered. However, most of their products remain still unknown.

The challenge now is to be able to discover new structural information about gene clusters' products. Throughout this project, two genomic representations were used to check if it was possible to predict structural fingerprints (binary vector representations in which the presence or absence of certain substructures is encapsulated) from genomic data. Both k-mers and Pfam counts were used as the input for two different machine learning algorithms: Neural Network and IOKR (Input Output Kernel Regression).

Each algorithm first learned from 825 gene clusters, for which it knew the expected output. Then, for each new unseen input, the algorithm was provided a set of 206 possible output vectors (among which was the expected vector), which it ranked according to their similarity with the predicted output vector. Given these ranked estimations, the frequency with which the expected output was set to be within the Top 1, Top 3, Top 5, etc; was recorded. Additionally, the performance was measured as objectively as possible by averaging the results from five cross-validation subsets.

IOKR was found to consistently generate better predictions than the neural network. Regarding the input vectors, Pfam counts seem to faithfully summarise the sort of enzymes coded by a gene cluster, which is ultimately correlated with its product. Finally, different approaches for further research were discussed.

2. Introduction

Machine learning has become one of the most powerful techniques in medicine and bioinformatics due to the development of computational power in recent years. It has been used in a wide range of applications: from combating antimicrobial resistance (Macesic *et al.*, 2017), through classifying its stages and predicting the survival likelihood of cancer (Montazeri *et al.*, 2016) to predicting the human phenotype (Basile and Ritchie, 2018).

However, the commonly overoptimistic perspectives on the performance of artificial intelligence (AI) have led to confusion about what can actually be achieved using machine learning methods (Park *et al.*, 2019). It is essential to measure the prediction capability of different algorithms to discriminate which methods will potentially generalise well. In other words, machine learning methods are expected to exhibit similar performance with new datasets as well (Miotto *et al.*, 2018).

Gene clusters are groups of genes close to each other in the genome that are related in a common pathway through a series of consecutive reactions necessary to synthesise a certain metabolite (Chavali and Rhee, 2018). Most of them are thought to be present in microorganisms; however, gene clusters have also been detected in plants (Schläpfer *et al.*, 2017) and fungi (Lind *et al.*, 2017) over the last years. Some of those clusters altogether with the compound they produce can be found in databases like MIBiG (Medema *et al.*, 2015).

These products are commonly antibiotics, anti-cancer agents, insecticides, etc (Chavali and Rhee, 2018). Despite their utmost importance in public health care, most of them have remained undiscovered until the recent development of bioinformatics tools, based on machine learning methods, capable of predicting DNA sequences likely to contain Biosynthetic Gene Clusters (BGCs) which produce secondary metabolites (Adamek *et al.,* 2016).

Other DNA-sequence-based machine learning algorithms are aimed at predicting RNA splicing (Zhang *et al.,* 2019; Signal *et al.*, 2017), DNA accessibility, DNA binding, histone modification, DNA methylation (Chen *et al.*, 2019) and RNA binding, among others (Avsec *et al.*, 2019). However, little work has been done in the prediction of the product of a gene cluster based on the genetic information present in those genes (Villebro *et al.,* 2019; Skinnider *et al.*, 2017).

A neural network is a machine learning technique based on how the human brain works. It comprises several layers of neurons whose individual units are activated or not depending on the input from the previous layer. Each neuron has several parameters which are updated after each iteration to reduce the value of a loss function. This way, the algorithm tries to progressively minimise the difference between the expected output and the predicted output for a certain configuration of parameters (LeCun *et al.*, 2015). Neural networks can be used for different purposes in the field of bioinformatics due to their potential and their flexibility depending on the architecture, loss function, optimiser, etc (Min *et al.,* 2017; Celesti *et al.,* 2018).

Secondly, Input Output Kernel Regression (IOKR) is a machine learning algorithm (Brouard *et al.*, 2016) which uses a function to correlate a structured input (for example a BGC) directly with a structured output (for instance, the product of a BGC). It does not explicitly predict an output vector, instead, it chooses the most likely one from a set of possible vectors with which it must be provided beforehand.

Using machine learning approaches, it would be ideal to predict what the product of a certain gene cluster is; however, in most cases this is hard, so a representative output was chosen, fingerprints (Muegge and Mukherjee, 2016). Fingerprints can be used as a systematic representation of chemical compounds, obtaining a binary vector for each molecule in which it is stated the presence (1) or absence (0) of several molecular properties; for example, the existence of certain substructures (Yin *et al.*, 2019). Fingerprints are widely used for virtual screening using fingerprint similarity to find new candidate drugs with similar catalytic activity (Cereto-Massagué *et al.*, 2015).

For solving this problem with a machine learning algorithm, not only does one need an output fingerprint vector for each BGC but also, one must find ways of converting the genetic information from a gene cluster into a proper input vector (Dawit and Henkel, 2017). Some of these representations have *'a priori'* little biological meaning such as DNA and amino acid k-mers (Chen *et al.*, 2012) Others, for example Pfam counts, are based on the likelihood of finding certain protein domains in the CDS coded by the DNA sequence, which is ultimately related with the sort of enzymes that synthesise the cluster's product (Eddy, 2009).

Throughout this project, two genomic representations (k-mers and Pfam counts) were used as the input of two machine learning algorithms: Neural Network and IOKR (Input Output Kernel Regression). The aim was to check if it was possible to predict structural fingerprints, which encapsulate the presence or absence of certain substructures in gene cluster's products.

3. Materials and methods

3.1. Software and data used

The project was carried out on the programming language *Python 3.7.3*. The code was written using the platform *Jupyter notebook*. Rather than starting from scratch, several Python libraries were installed and imported: *Numpy*, useful for working with arrays and matrixes; *Keras*, an open source neural network; *SeqIO*, used for parsing GenBank files; *Scikit-learn*, which provides a range of machine learning algorithms with which to do benchmarking, and methods for plotting dimensionality reduction graphs; *RDKit*, for obtaining the fingerprints for each BGC; and *Os* for interacting with files from the computer. The code used in this project was stored on GitHub (*https://github.com/victorrgez/2425005RSummerProject*).

Windows Subsystem for Linux (WSL) with an Ubuntu environment was used since *Hmmer 3.2.1* (software used for detecting Pfam domains) cannot work on Windows. The Biosynthetic Gene Clusters' data was obtained from MIBiG (Medema *et al.*, 2015), a repository which provides the DNA sequence, the amino acid sequence and the Smile Strings of the products for each cluster, being those products divided into several categories: Polyketides, Alkaloids, Saccharides, etc.

There are many possible sets of fingerprints. CDK fingerprints were used for optimising the machine learning algorithms. These fingerprints were characterised using a list of 306 substructures obtained from the manual of CDK, a library similar to RDKit *(http://cdk.github.io/cdk/1.4/docs/api/org/openscience/cdk/fingerprint/SubstructureFingerprinter.html. Accessed on 12th June 2019)*

3.2. Methods

3.2.1. Input vectors

Several possible inputs for machine learning algorithms were considered. For obtaining DNA k-mer counts, each sequence was scanned through while annotating how many times did each k-mer appear in the sequence of each cluster. This process was repeated for k-mers of different lengths, considering that the number of possible k-mers scales exponentially with respect to their length. Therefore, the higher the length of the k-mer, the more computationally expensive the input is to store and run with a machine learning algorithm. The number of possible DNA k-mers is 4^n, being 'n' the length, since there are four different possible nucleotides in each position. The same process was carried out with amino acid k-mers, being the number of possible k-mers 20^n, as there are 20 different amino acids in the sequence.

Another more biologically-based input used was Pfam counts. These vectors summarise the type of Pfam domains present in each BGC, which is ultimately related with the secondary metabolite they produce. The software *HMMER* was employed to search for Pfam domains in the amino acid sequence of each cluster. The output files from *HMMER* were parsed in order to find out which Pfam domains were likely to be present in each BGC. Consequently, 6159 different Pfam domains were annotated in total. Each cluster was associated with an input vector of length 6159 in which each position corresponded to the presence (1) or absence (0) of each Pfam domain.

3.2.2. Output vectors

Regarding the output prediction for machine learning algorithms, it would have been desirable to predict what the product of a certain gene cluster is; however, in most cases this is hard, so an easier output was used, fingerprints. The fingerprint of each product summarised the presence (1) or absence (0) of 306 substructures (according to the list obtained from CDK).

RDKit was used to compute which substructures were present in the product of each cluster. The method "*HasSubstructMatch*" returned a 1 or a 0 for each pair of a Smile String and a Smart String, indicating whether the substructure was found in the product being analysed or not.

3.2.3. Machine learning algorithms

Two different machine learning algorithms were used to predict fingerprints. In both cases, the dataset of 1031 BGCs was first randomised and then divided into 5 different groups (0-206, 207-412, 413-618, 619-824 and 825-1031). In each experiment, 4 of these groups were included in the training set and the other one corresponded to the test set. By doing so, 5 different combinations of the groups for the training and the test sets were used as crossvalidation (given that the dataset is relatively small) and the averages of the results were computed in each experiment.

The first algorithm used was a Keras Neural Network. It comprises several layers with a number of units that act similarly to neurons in the human brain. On learning from the training set (that is, inputs for which the output fingerprints are known), the system changes the weights and the bias for each unit in order to progressively obtain a smaller difference with respect to the expected output.

The experiments involved measuring the ability of the neural network to discriminate between the expected output vector and the other fingerprints belonging to the test set in each experiment (206 vectors). However, the output vector of length 306, predicted by the neural network for each input from the test set, is not likely to match completely with any of fingerprints from the test dataset. Instead, the similarity between the output vector and those in the test dataset was computed individually and ranked to find out which output vector in the dataset was more similar to the one being predicted.

$$\boldsymbol{Gaussian\ Kernel}\ (\boldsymbol{vector\ 1}, \boldsymbol{vector\ 2}) = \boldsymbol{exp}(-\Upsilon * \boldsymbol{Euclidean\ distance}^{\boldsymbol{2}})$$

Figure 1. Gaussian kernel formula for two vectors. The value depends on the parameter ϒ and on the squared Euclidean distance between the two vectors. These vectors must have the same dimensions, for instance, two fingerprints or two Pfam-counts vectors.

This similarity was calculated with a Gaussian Kernel (figure 1) which depends on ϒ (a parameter that can be changed) and the squared Euclidean Distance between the vectors being compared (the more similar two vectors are, the smaller the Euclidean distance is). Since the expected vector is known, it was annotated in which position it was ranked according to the similarity scores obtained.

Consequently, for each experiment it was computed the percentage of times in which the expected output vector was ranked first, Top 3, Top 5, Top 10 and Top 25 for each input vector from the test set.

The second machine learning algorithm used in this project was the Input Output Kernel Regression (IOKR) algorithm (Brouard *et al.*, 2016). As described by Brouard *et al.*, this method uses a function that correlates a structured input (for instance a BGC) directly with a structured output (the product of the BGC). Using one kernel (figure 1) for the input vector and another one for the output, IOKR will compute a score for each possible pair. Then, it will choose which fingerprint out of a set of possible output vectors is the most likely to be the expected one:

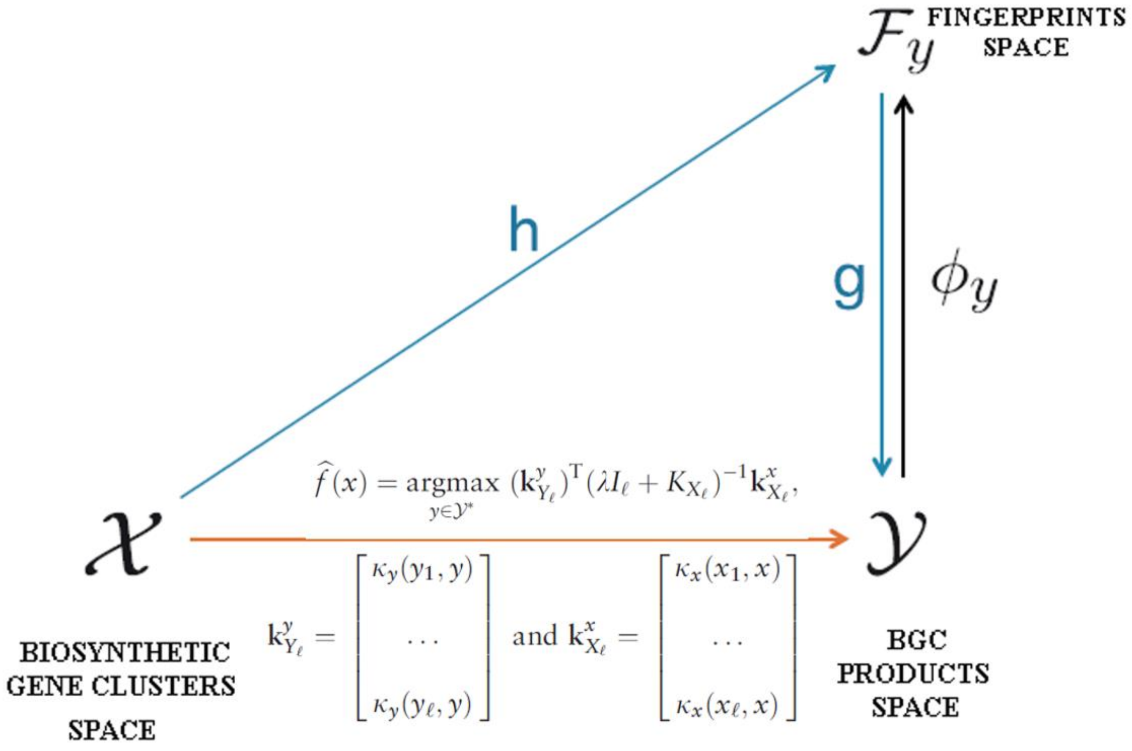

$\mathcal{F}_{\mathcal{Y}}$
FINGERPRINTS SPACE
h
g
$\phi_{\mathcal{Y}}$
$\widehat{f}(x) = \underset{y\in\mathcal{Y}^*}{\operatorname{argmax}}\ (\mathbf{k}^{y}_{Y_\ell})^{\mathrm{T}}(\lambda I_\ell + K_{X_\ell})^{-1}\mathbf{k}^{x}_{X_\ell},$
$\mathcal{X}$
$\mathcal{Y}$
$\mathbf{k}^{y}_{Y_\ell} = \begin{bmatrix} \kappa_y(y_1, y) \\ \dots \\ \kappa_y(y_\ell, y) \end{bmatrix}$ and $\mathbf{k}^{x}_{X_\ell} = \begin{bmatrix} \kappa_x(x_1, x) \\ \dots \\ \kappa_x(x_\ell, x) \end{bmatrix}$
BIOSYNTHETIC GENE CLUSTERS SPACE
BGC PRODUCTS SPACE

Figure 2. IOKR algorithm. F(x) correlates directly a structured input (BGCs) with a structured output (compounds). Instead of predicting a fingerprint, IOKR selects the most likely fingerprint (the one with the highest score, 'argmax') for each input (in this project, the test dataset comprised 206 possible output fingerprints). Lambda (λ) is a parameter than can be modified, I is the identity matrix and K_x is the Gram Matrix of the input vectors (Pfam counts for example). This central part of the formula is always the same (for each subset of the experiment). On the other hand, k_x is computed for each input vector and k_y for each fingerprint. Both k_x and k_y are column vectors in which the Kernel value (figure 1) between all the vectors in the training set and the one being tested is computed pairwise. Modified from Brouard *et al.*, 2016.

With this method, predicting a certain fingerprint as the output for each input vector is no longer needed. The algorithm will simply compute a score for each pair of input/output vectors. The higher the score, the more likely for a certain output vector to be the expected one. The fingerprints from the test set were chosen as the dataset of possible outputs (206). Similar to the aforementioned process with the Keras neural network, the possible output vectors were ranked for each input of the test set. Afterwards, it was computed the percentage of times in which the expected output vector was ranked first, within the Top 3, etc.

There are three different parameters in the IOKR formula that were changed in order to improve the performance of the algorithm: λ, Υ_x (belongs to the kernel used in K_x) and Υ_y (in the kernel formula of K_y). Given that the uncertainty derived from the initial order of the clusters (which was randomised at the start of each experiment) was greater than the score difference between different combinations of parameters, in the last experiment, the same randomised initial order was maintained for all settings. This procedure was carried out three times and the results were averaged.

3.2.4. Approaches for further research

As an extension of the project, several types of fingerprints were used with the algorithms that had already been optimised with CDK fingerprints. The purpose of this section was to analyse the potential of other fingerprints that might help fill the gaps previously identified with CDK fingerprints. That is, it would be encouraging to find other substructures that act as markers for other categories, for which lower scores were obtained.

Although there are several datasets of substructures that conform different kinds of fingerprints, not all of them are publicly available. For instance, DataWarrior (Sander *et al.,* 2015) is a free software that lets the user calculate the similarity between two molecules depending on their Smile Strings; however, the fingerprint of each molecule remains hidden for the user and only the similarity is displayed on the program. Another example is the MACC fingerprints, also known as the MDL keys (Durant *et al.*, 2002), for which only 166 out of 960 substructures were made public.

The additional types of fingerprints used in this project were MACC and Topological fingerprints (obtained with *RDKit*); and Daylight, Estate, Klekota-Roth and PubChem fingerprints (obtained with the module *PyFingerprint*, available at *https://github.com/hcji/PyFingerprint. Accessed on 16th July 2019*).

Finally, regarding the Pfam-counts input vectors used, some of the domains are completely different while others are varieties of the same domain with almost identical biological function. Different methods for summarising the information of similar domains were attempted as a means of simplifying the differences between BGCs that produce the same compounds.

4. Results

4.1. Suitability of CDK fingerprints

It is essential that fingerprints, apart from being predictable, are significantly different depending on the product category to which they belong. That is, fingerprints related to products of the same category should be clustered together when computing dimensionality reduction.

In the next figure it is shown the '*t-distributed Stochastic Neighbour Embedding*' plot (t-SNE) of CDK fingerprints in a 2-dimensional space where the points are coloured depending on the category of the product:

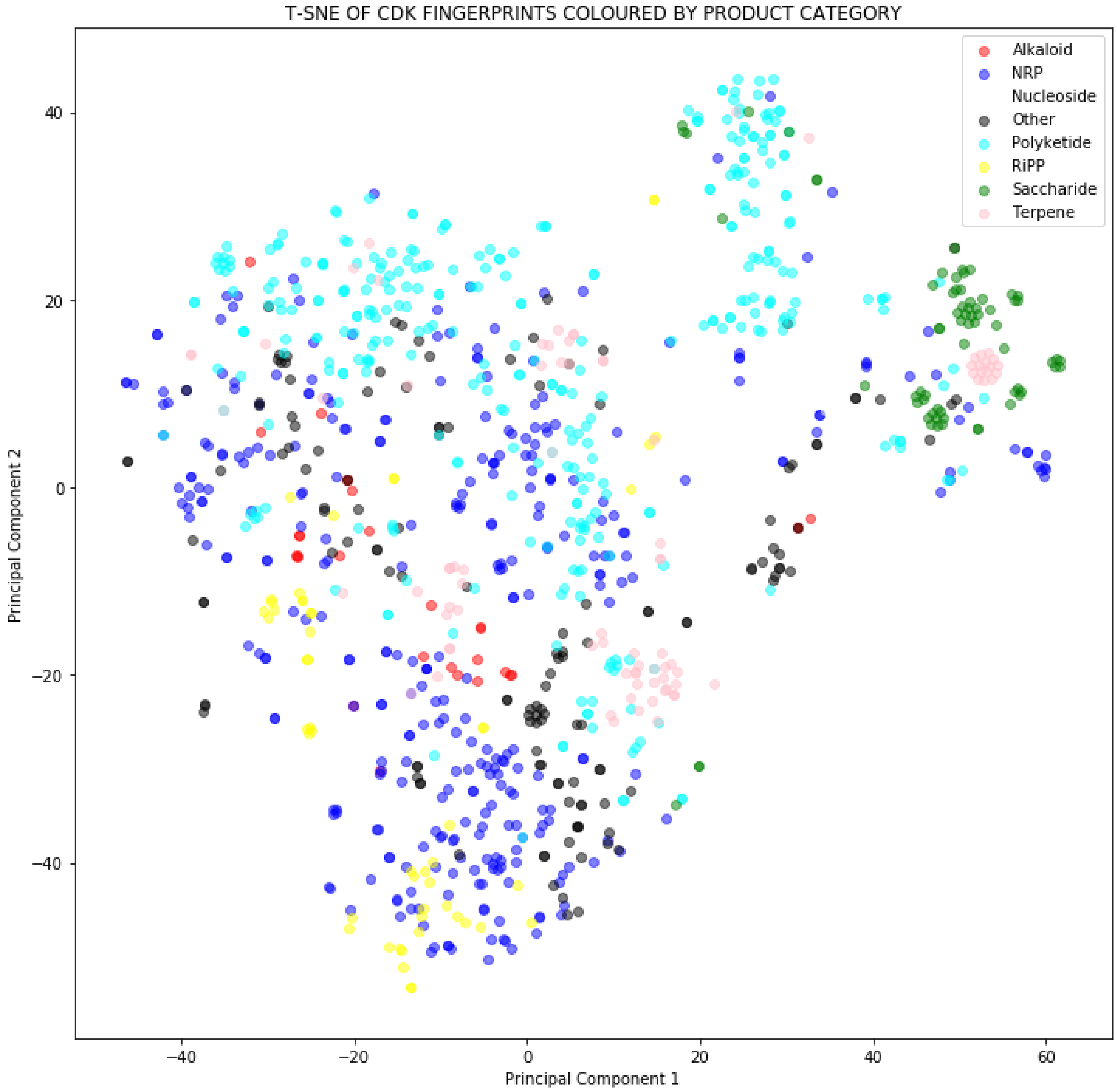


Figure 3. Dimensionality reduction of CDK fingerprints coloured by category. The t-SNE plot shows fingerprints of the same colour being close to each other in most categories, especially saccharides (green).

Even though the resolution is not likely to be high when separating seven different categories in a single 2D plot, interestingly enough, saccharides' fingerprints are much closer to each other when compared against other categories. Arguably, categories separated with better resolution might have specific substructures (markers) which might help predict their products with higher accuracy.

Afterwards, each individual fingerprint substructure was analysed with regards to the proportion of zeros and ones within each category. If a certain substructure is a marker for a certain group, it will be present in those fingerprints with higher frequency than in the rest of the categories.

With this purpose, it was calculated the difference between the two categories with the highest percentage for each substructure. Then, it was found that if a 20% difference cut-off was applied, in 12 out of 38 cases, Saccharides were the Top 1 category. If the cut-off was 30%, then Saccharides were the Top 1 in 11 out of 15 cases. Furthermore, when the cut-off is 40%, 10 out of 12 cases are markers for Saccharides. This might be a sign that saccharides will be predicted with higher accuracy: CDK fingerprints seemingly include reliable saccharides markers which might help discriminate these products from the rest:

Table 1. Substructures with significant abundance difference between the first and the second category. The cut-off applied was 40%. 10 out of 12 substructures are markers for Saccharides.

Substructure		**Difference between**	
position (0-based)	**Name**	**1st and 2nd (%)**	**Top 1 category**
3	Quaternary carbon	46.37	Terpenes
12	Primary alcohol	44.62	Saccharides
13	Secondary alcohol	42.25	Saccharides
40	1,2-Diol	52.22	Saccharides
55	Acetal	60.97	Saccharides
62	Acetal like	58.14	Saccharides
279	Bridged rings	42.75	RiPPs
280	Sugar pattern 1	55.29	Saccharides
281	Sugar pattern 2	64.41	Saccharides
282	Sugar pattern combi	58.52	Saccharides
284	Sugar pattern 2 alpha	64.41	Saccharides
285	Sugar pattern 2 beta	64.41	Saccharides

4.2. Baseline results

All the results were compared against the baseline of expected scores obtained by using random input vectors. In that case, the 206 fingerprints would be ranked randomly. It was computed the frequency with which the expected fingerprint would be set to be within the Top 1, Top3, Top 5, Top 10 and Top 25. The expected output vector would be ranked first once every 206 times, ranked second all those times it was not ranked first and divided by 205 possible output vectors for this iteration. Sequentially, the expected output vector would be ranked third all those times it was not ranked first nor second divided by 204 possible vectors that might be chosen in that position. In order to know the frequency of the expected vector being within the Top 3, the probabilities of it being ranked first, second and third would be added:

Table 2. Expected results for random input vectors without any biological meaning.

	Frequency (%)
TOP 1	**0.49**
TOP 3	**1.46**
TOP 5	**2.43**
TOP 10	**4.85**
TOP 25	**12.14**

4.3. IOKR algorithm using k-mers

The results obtained by using k-mers as the input for the IOKR algorithm were only slightly better than the baseline. It seems like greater lengths result in lower scores. The results are shown in tables 3 and 4:

Table 3. Results for DNA k-mers using IOKR (%). The results are the % of times the expected output vector is ranked within the specified Top out of all the vectors in the test set (206). Each column is the average of the crossvalidation of 5 subsets . The percentages obtained are slightly higher than the baseline.

Length	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Top 1	3.98	3.39	2.62	1.17	1.26	0.87	0.78
Top 3	7.47	6.79	5.24	3.78	3.10	2.52	2.13
Top 5	9.51	8.53	7.57	5.53	4.36	3.49	2.81
Top 10	15.71	12.22	12.42	10.09	7.86	7.27	5.82
Top 25	28.77	24.63	23.57	19.69	18.82	16.78	15.03

Table 4. Amino acid k-mers using IOKR (%). They performed worse than low-length DNA k-mers.

Length	**2**	**3**	**4**
Top 1	1.94	1.07	0.97
Top 3	4.27	2.72	1.94
Top 5	6.21	4.85	2.91
Top 10	11.45	8.44	7.28
Top 25	22.99	18.04	16.99

4.4. IOKR algorithm using Pfam Counts

The process of optimising three parameters: λ (in the IOKR formula), Υ_x (in the Gaussian kernel for Pfam vectors) and Υ_y (in the Gaussian kernel for fingerprint vectors) involved running several experiments in which two parameters would stay the same and the other one would change in order to find the best configuration. In the beginning, λ was changed:

Table 5. IOKR results (%) when changing the value of λ. The best scores in each row are highlighted in yellow. The best value of λ seems to be 0.01 (highlighted in green).

Υ_x	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Υ_y	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
λ	0.001	0.003	0.01	0.03	0.1	0.3	1	3
Top 1	32.20	31.03	33.85	31.91	32.4	26.87	20.27	11.06
Top 3	40.45	40.35	42.10	42.28	40.35	37.54	28.90	18.72
Top 5	46.07	45.20	47.43	47.33	45.98	42.58	35.11	22.99
Top 10	55.19	54.61	55.77	55.09	55.58	51.02	45.00	32.30
Top 25	69.35	70.32	71.77	66.90	71.20	68.19	60.04	49.51

Then, Υ_x was changed to identify its optimal value:

Table 6. IOKR results (%) when changing the value of Υ_x. When it is 0.003, the scores are best overall than with 0.01.

Υ_x	0.001	0.003	0.01	0.03	0.1	0.3	1	3
Υ_y	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
λ	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Top 1	31.62	33.75	33.85	29.68	28.23	29.87	31.04	31.23
Top 3	41.42	44.03	42.10	36.95	34.43	38.70	38.12	37.73
Top 5	47.72	48.69	47.43	41.42	38.02	41.70	41.99	41.42
Top 10	56.64	57.80	55.77	50.53	44.23	47.62	48.01	47.33
Top 25	71.89	78.64	71.77	65.18	57.03	58.58	58.19	57.33

Finally, λ and Υ_x were fixed and Υ_y was changed:

Table 7. IOKR results (%) when changing the value of Υ_y. The results seem promising when $\Upsilon_y = 0.1$ since it gets quite a higher score for the Top 1 than in the rest of the experiments.

ϒ x	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003
ϒ y	0.001	0.003	0.01	0.03	0.1	0.3	1	3
λ	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Top 1	31.62	31.24	33.75	33.17	36.27	33.85	28.42	27.74
Top 3	41.42	41.13	44.03	42.77	45.39	41.51	31.91	29.39
Top 5	48.01	45.2	48.69	46.85	49.86	46.17	34.24	29.68
Top 10	57.71	55.09	57.80	56.26	57.71	51.60	37.05	30.45
Top 25	71.39	70.32	78.64	68.96	72.16	62.76	42.97	34.14

Although this configuration of parameters (Υ_x=0.003 Υ_y=0.1 and λ=0.01) seems to be the best one, the variance depending on the random initial order of the clusters in each experiment might be bigger than that springing from the change in the value of the parameters. Therefore, three experiments trying the best configurations of parameters were carried out maintaining for each experiment the same order of clusters for every setting:

Table 8. Results of experiment 1 (%). The best configurations of parameters are tested with the same random order of clusters to avoid intrinsic variance.

ϒ x	0.01	0.01	0.01	0.003	0.003	0.003	0.003	3
ϒ y	0.01	0.01	0.01	0.01	0.03	0.1	0.3	3
λ	0.001	0.01	0.1	0.01	0.01	0.01	0.01	0.01
Top 1	31.33	31.72	30.16	31.72	33.08	35.01	34.14	32.50
Top 3	40.64	41.32	40.54	41.42	43.07	43.94	41.03	38.61
Top 5	46.07	47.23	46.85	48.50	49.08	49.95	45.2	42.00
Top 10	54.03	55.77	55.09	56.84	57.13	58.78	51.40	48.59
Top 25	68.28	69.74	70.03	72.36	72.70	72.94	63.53	58.20

Table 9. Results of experiment 2 (%). The clusters' order is shuffled again, and all the configurations are tested with the new order. The best set up is the same one than in experiment 1.

ϒ x	0.01	0.01	0.01	0.003	0.003	0.003	0.003	3
ϒ y	0.01	0.01	0.01	0.01	0.03	0.1	0.3	3
λ	0.001	0.01	0.1	0.01	0.01	0.01	0.01	0.01
Top 1	31.33	31.81	31.03	31.52	32.98	35.31	34.24	32.39
Top 3	40.83	41.13	40.06	41.71	42.78	44.23	40.16	39.67
Top 5	45.40	46.56	45.59	47.53	48.50	48.79	44.72	43.07
Top 10	53.93	54.71	54.51	55.49	56.94	57.62	51.51	48.01
Top 25	68.48	70.75	70.32	71.39	72.21	72.58	62.67	58.39

Table 10. Results of experiment 3 (%). It seems that $\Upsilon_x=0.003$ $\Upsilon_y=0.1$ and $\lambda=0.01$ systematically produce the best results regardless of the original order of clusters.

ϒ x	0.01	0.01	0.01	0.003	0.003	0.003	0.003	3
ϒ y	**0.01**	**0.01**	**0.01**	**0.01**	**0.03**	**0.1**	**0.3**	**3**
λ	**0.001**	**0.01**	**0.1**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Top 1	32.59	33.17	32.20	32.88	34.24	36.28	34.34	32.40
Top 3	40.93	41.81	40.45	43.36	44.13	45.30	40.84	39.77
Top 5	46.36	47.24	45.59	48.11	49.08	49.57	45.78	43.36
Top 10	55.38	55.29	54.22	56.07	56.06	56.94	51.99	48.69
Top 25	67.12	67.9	69.36	69.35	69.93	70.71	62.37	59.07

Finally, the results of the three experiments were averaged in order to compare them with the results obtained afterwards with the Keras neural network:

Table 11. Average of the three experiments (%). $\Upsilon_x=0.003$ $\Upsilon_y=0.1$ and $\lambda=0.01$ conform the best set up for the IOKR algorithm with Pfam counts as the input.

ϒ x	0.01	0.01	0.01	0.003	0.003	0.003	0.003	3
ϒ y	**0.01**	**0.01**	**0.01**	**0.01**	**0.03**	**0.1**	**0.3**	**3**
λ	**0.001**	**0.01**	**0.1**	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
Top 1	31.75	32.23	31.13	32.04	33.43	35.53	34.24	32.43
Top 3	40.8	41.42	40.35	42.16	43.33	44.49	40.68	39.35
Top 5	45.94	47.01	46.01	48.05	48.89	49.44	45.23	42.81
Top 10	54.45	55.26	54.61	56.13	56.71	57.78	51.63	48.43
Top 25	67.96	69.46	69.90	71.03	71.61	72.08	62.86	58.55

Additionally, it was also interesting to find out if every product category was predicted with the same accuracy or if, on the contrary, there were some types of BGC's products that were easier to predict. With this purpose, two experiments with five cross validation replicates each were carried out. The results of the 10 batches were added together to calculate the percentage of the rankings for each category:

Table 12. IOKR results broken down into product categories using Pfam counts as the input. Saccharides are predicted with higher accuracy than the rest of the categories.

	TOP 1	TOP 3	TOP 5
ALKALOID	**55.17%** (32 out of 58)	**62.07%** (36 out of 58)	**63.79%** (37 out of 58)
NRP	**29.27%** (192 out of 656)	**38.11%** (250 out of 656)	**42.07%** (276 out of 656)
NUCLEOSIDE	**0%** (0 out of 2)	**0%** (0 out of 2)	**0%** (0 out of 2)
OTHERS	**38.41%** (106 out of 276)	**44.93%** (124 out of 276)	**48.55%** (134 out of 276)
POLYKETIDE	**36.78%** (242 out of 658)	**45.14%** (297 out of 658)	**50.91%** (335 out of 658)
RIPP	**39.22%** (40 out of 102)	**45.10%** (46 out of 102)	**50.00%** (51 out of 102)
SACCHARIDE	**64.29%** (90 out of 140)	**70.00%** (98 out of 140)	**74.29%** (104 out of 140)
TERPENE	**39.41%** (67 out of 170)	**51.18%** (87 out of 170)	**62.35%** (106 out of 170)

	TOP 10	TOP 25
ALKALOID	**63.79%** (37 out of 58)	**72.41%** (42 out of 58)
NRP	**49.54%** (325 out of 656)	**66.62%** (437 out of 656)
NUCLEOSIDE	**0%** (0 out of 2)	**0%** (0 out of 2)
OTHERS	**52.90%** (146 out of 276)	**65.58%** (181 out of 276)
POLYKETIDE	**61.85%** (407 out of 658)	**76.90%** (506 out of 658)
RIPP	**58.82%** (60 out of 102)	**66.67%** (68 out of 102)
SACCHARIDE	**78.57%** (110 out of 140)	**90.71%** (127 out of 140)
TERPENE	**74.12%** (126 out of 170)	**85.29%** (145 out of 170)

These results support the previous analysis on CDK fingerprints where saccharides were clustered together in the t-SNE plot. It seems like the presence of specific markers help predict saccharides with higher accuracy.

4.5. Neural network using k-mers

The scores obtained using both kinds of k-mers (DNA and amino acids) are quite similar to those achieved with the IOKR algorithm. The performance is worse than with Pfam counts. This supports the fact that they do not convey much biological meaning since they just summarise repetitions of words of length “k” in the sequence, which is not that directly related with the product of a gene cluster. The results are shown in tables 13 and 14:

Table 13. Results of DNA k-mers using a Keras neural network (%). The scores obtained are rather low and close to those expected to be achieved with random input vectors.

Length	2	3	4	5	6
Top 1	1.46	1.94	0.49	2.91	1.46
Top 3	3.40	4.37	3.88	6.31	1.46
Top 5	4.85	6.80	3.88	7.77	6.31
Top 10	8.74	8.25	7.77	14.08	6.31
Top 25	19.90	19.42	14.08	20.87	15.05

Table 14. Results of amino acid k-mers using a Keras neural network (%). The results are similar to those obtained with DNA k-mers.

Length	2	3	4
Top 1	6.31	0.49	0.97
Top 3	12.62	3.88	2.43
Top 5	15.53	3.88	5.34
Top 10	25.24	7.77	9.71
Top 25	36.89	16.50	19.42

4.6. Neural network using Pfam Counts

Many architectures and parameters were tried in this section of the project. Getting a neural network to optimally work with high dimensionality in both the input and the output vectors is usually a laborious process. Among all the possibilities that can be adjusted, it is crucial to optimise the number of hidden layers, the number of neurons per layer, the optimiser, the learning rate, the loss function and the number of epochs.

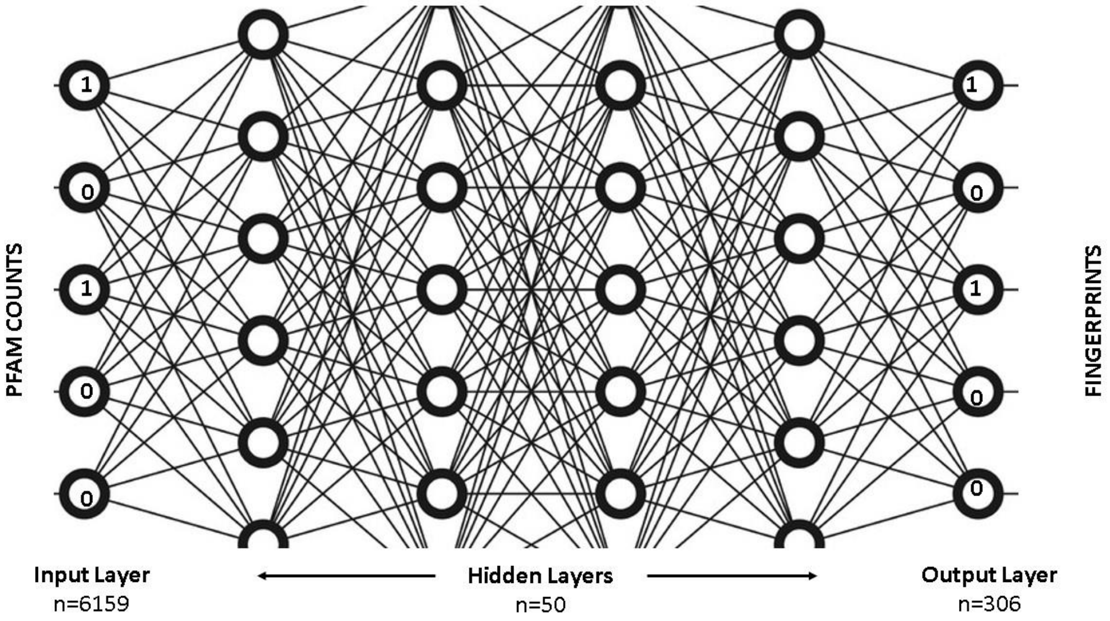


Figure 4. Neural network architecture for which the best scores were obtained. A binary input vector of 6159 bits (Pfam counts) is passed to the input layer and then the output layer gives a 306-bits output vector as a result (fingerprint). There are four hidden layers of 50 neurons each. The activation function is ReLU for all the layers but for the last one (sigmoid). The output numbers are a probability between 0 and 1, for instance, 0.6 means that the probability of 1 is 60% and the probability of 0 is 40%. The numbers were rounded up and down to 1 and 0 before calculating the kernel values with the fingerprints from the test set. Modified from *https://i.udemycdn.com/course/750x422/1795952_e23e_2.jpg (Accessed on 10th August 2019)*.

Despite systematically getting higher scores throughout several experiments, the results are still far away from those obtained with the IOKR algorithm. The configuration that obtained the best scores was comprised of 4 hidden layers of 50 neurons each. Regarding the learning rate, it was 0.01 during the first 25 epochs, then 0.003 for 35 epochs and 0.001 during the last 20 epochs. At this point, even if the learning rate is lowered more to try to increase the accuracy, the binary crossentropy loss function is usually stuck in a local optimum which cannot be solved using a smaller learning rate. Nonetheless, the results are much better than those achieved with k-mers. In the following tables it is shown the three final experiments and the average obtained with this architecture:

Table 15. Results of the first experiment of the neural network with the best configuration found for Pfam counts (%).

	SUBSET 1	SUBSET 2	SUBSET 3	SUBSET 4	SUBSET 5	AVERAGE
TOP 1	19.90	21.84	22.82	22.33	23.79	**22.14**
TOP 3	33.01	31.55	31.07	30.10	33.01	**31.75**
TOP 5	35.44	36.41	36.41	34.95	38.35	**36.31**
TOP 10	45.15	45.63	47.09	44.66	48.06	**46.12**
TOP 25	64.56	58.74	62.14	62.62	61.17	**61.85**

Table 16. Second experiment. Even though the order of the BGCs was randomised again, the results (%) seem to be consistent with those of the first experiment.

	SUBSET 1	SUBSET 2	SUBSET 3	SUBSET 4	SUBSET 5	AVERAGE
TOP 1	26.21	24.76	16.02	20.39	26.70	**22.82**
TOP 3	37.86	30.58	24.76	30.10	35.92	**31.84**
TOP 5	45.63	35.44	33.50	37.38	42.23	**38.84**
TOP 10	57.77	47.09	46.12	47.57	53.88	**50.49**
TOP 25	72.82	59.71	60.68	63.11	67.96	**64.86**

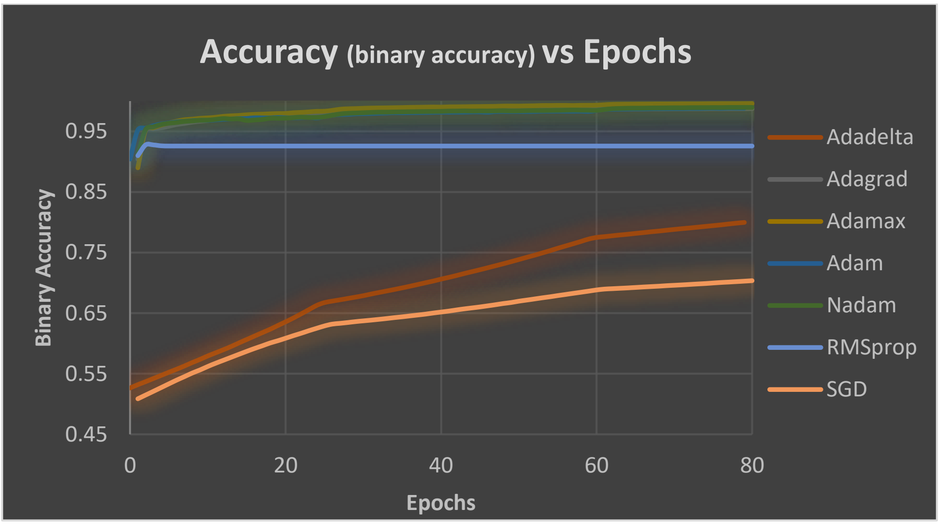
Table 17. Third experiment of the neural network with Pfam counts. There is a considerable difference in the scores (%) among the five subsets depending on which clusters are randomly put in each crossvalidation subset. The average results are consistent with those obtained in experiments 1 and 2.

	SUBSET 1	SUBSET 2	SUBSET 3	SUBSET 4	SUBSET 5	AVERAGE
TOP 1	35.92	18.45	16.02	22.82	21.84	**23.01**
TOP 3	45.63	28.16	27.67	32.04	31.07	**32.91**
TOP 5	50.49	33.98	35.92	38.83	37.38	**39.32**
TOP 10	57.28	43.20	45.63	49.03	51.46	**49.32**
TOP 25	67.48	57.28	59.22	66.50	63.59	**62.81**

Table 18. Final results (%) of the Neural Network using Pfam Counts as the input. The scores obtained for the Top 1 are 10-15% worse than those obtained with IOKR.

	FINAL AVERAGE
TOP 1	22.66
TOP 3	32.17
TOP 5	38.16
TOP 10	48.64
TOP 25	63.17

These experiments were run using the Keras optimiser "Adam". However, there were several optimisers which obtained similar results. In the following graph it is shown the loss function and the accuracy progress of the model throughout the epochs for each optimiser. RMSprop, Adadelta and SGD performed worse than the other four optimisers, whose results were similar:



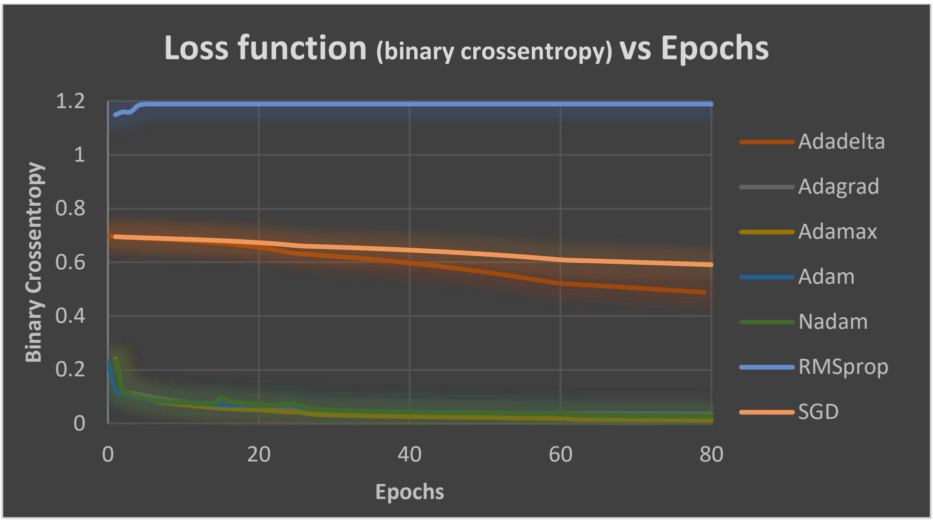


Figure 5. Evolution of the Accuracy and the Loss function with the number of epochs during which the Keras Neural Network has been trained. Several optimisers were compared. RMSprop, SGD and Adadelta proved to be unsuitable whereas the others performed similarly. Adam was chosen for the rest of the project.

4.7. Future work, where next?

In table 19 and table 20 it is shown the average of the scores obtained for five subsets with different types of fingerprints. The same random order of clusters was used in all the experiments so as to compare their potential against the results obtained with CDK fingerprints, with which both algorithms had been optimised initially:

Table 19. Comparation of the results (%) of other fingerprints against the CDK ones using the IOKR algorithm. MACC fingerprints (they only include the 166 public substructures) obtain slightly better scores.

	CDK	DAYLIGHT	ESTATE	KLEKOTA-ROTH	MACC	PUBCHEM	TOPOLOGICAL
TOP 1	**35.63**	28.83	31.46	30.68	36.60	33.79	25.05
TOP 3	**43.88**	33.79	38.25	35.73	46.12	40.10	27.47
TOP 5	**49.42**	36.11	41.84	38.64	51.65	43.11	27.77
TOP 10	**57.48**	39.13	47.09	43.4	59.61	49.51	28.74
TOP 25	**70.78**	46.99	60.78	52.62	73.59	59.90	31.26

Table 20. Results (%) for different types of fingerprints with the same random order of clusters using the neural network architecture previously optimised for CDK fingerprints. These scores might be increased individually by searching the best architecture for each set of fingerprints since the length of the output vector is different in each case. Nevertheless, these results should be representative of their future potential.

	CDK	DAYLIGHT	ESTATE	KLEKOTA-ROTH	MACC	PUBCHEM	TOPOLOGICAL
TOP 1	**23.49**	15.83	25.15	21.36	21.17	20.29	21.55
TOP 3	**33.01**	23.69	33.30	30.87	31.36	32.04	33.01
TOP 5	**39.61**	29.03	38.74	36.02	37.57	37.28	38.84
TOP 10	**50.29**	37.28	48.84	45.82	47.57	48.16	48.83
TOP 25	**65.15**	51.36	61.94	60.19	64.85	64.46	64.27

The IOKR results showed that the potential of other types of fingerprints is not far away from the scores obtained with CDK fingerprints. Furthermore, MACC fingerprints got better scores using this algorithm even if only the 166 public substructures were used (the size of the total set is around 1000 substructures). Some of these fingerprints' substructures might fill the gaps identified for CDK fingerprints, since most of the markers belong to the Saccharides. Consequently, a mixture of markers coming from different types of fingerprints might be combined to get better predictions for all the categories.

Regarding the neural network results, they are only representative of their suitability for predicting BGCs' products. To get conclusive results, each of these fingerprints should undertake several experiments trying different optimisers, learning rates, number of hidden layers, number of neurons per layers, etc. Nevertheless, many of these fingerprints performed in a similar way to CDK fingerprints and showed their future potential.

Finally, several ways of summarising the information of the Pfam-counts input vectors were tried but they were not successful. The name of the Pfam domains usually consists of a name plus "_" followed by a number; however, this is only sometimes the case and it is difficult to find an algorithm that clusters correctly each Pfam name with its variations.

For each Pfam domain, it was tried to remove a certain number of characters from the end of the name depending on the size of the domain name. The length of the input vector was reduced significantly to around 4000 types of Pfam domains; however, the scores obtained were slightly worse than those achieved with the original Pfam-counts vectors (possibly, the clustering of similar domains was not accurate enough). Future work will include the summarization of close Pfam domains into common categories for obtaining more accurate and meaningful input vectors.

5. Discussion

Biosynthetic gene clusters (BGCs) are groups of genes located close to each other which ultimately produce secondary metabolites (Chavali and Rhee, 2018). These BGCs can be found in different organisms such as bacteria, fungi and plants (Schläpfer *et al*., 2017; Lind *et al*., 2017). Genes in a certain BGC code for enzymes which coordinate a set of consecutive reactions in order to produce as a result a compound with relevant properties such as an antibiotic, an insecticide, etc (Chavali and Rhee, 2018).

In the last few years, many new BGCs have been discovered due to recent technological advances in computational power and machine learning approaches altogether with new bioinformatics tools. However, most of the BGCs are yet not annotated and it is still to be discovered which compounds are produced by them.

During this project, two different algorithms (neural network and IOKR) were used in order to discover new structural information about BGCs' products. As predicting a compound is rather difficult, fingerprints (a combination of 1s and 0s which indicate the presence or absence of certain substructures) were used as the output instead.

CDK fingerprints were initially analysed to find out if they were a good candidate for machine learning. These fingerprints were subjected to dimensionality reduction, colouring them by product category. It was found that even though the segregation of the categories is not perfect, most categories' fingerprints are close to each other in the t-SNE plot, especially the saccharides.

Additionally, the substructures included in CDK fingerprints were individually analysed to find those which are found in some categories with higher frequency than in others. It was found that when selecting only those substructures for which there is a statistical difference between the first and the second category with more abundance of a certain substructure, most of the times they turn out to be saccharide markers. If the cut-off is 30% difference between the Top 2 categories, 11 out of 15 are saccharide markers whereas when the cut-off is raised to 40% difference, 10 out of 12 are markers for the saccharide category. Arguably, the presence of specific substructures for a certain category might result in their products being predicted with higher accuracy.

With regard to the input vectors, it is essential that they summarise the genetic information of BGCs. Both DNA and amino acid k-mers were compared against a more biologically-consistent approach, Pfam counts. The presence or absence of certain Pfam domains in the genetic sequence of a BGC is related with the type of enzymes encoded by the BGC. Consequently, the type of compound produced is significantly associated with the Pfam domains that might be found in the BGC's sequence.

In the first place, both DNA and amino acid k-mers were used as the input of the IOKR algorithm. Regardless of the tuning of the three parameters (λ, Υ_x and Υ_y), the results were only slightly better than the outcome expected to be obtained by using random vectors as the input, for which the expected output should be ranked first around 0.49% of the times.

Afterwards, IOKR was systematically optimised using Pfam counts as the input. The best settings of parameters were progressively found until a final experiment was carried out with the same random order of clusters for all settings in order to reduce the variance arising from the order established at the start of each experiment.

The parameters that consistently obtained the best scores were λ=0.01, Υ_x=0.003 and Υ_y=0.1. Three different experiments' scores were averaged, and as a result, the expected fingerprint was ranked Top 1 more than 35% of the times, Top 5 almost 50% of the times and Top 25 almost 75% of the times. The results achieved by IOKR seemed quite promising since they were much better than the baseline; however, it was also interesting to know if certain product categories were predicted with higher accuracy than others.

Consequently, two different experiments with five crossvalidation subsets each were averaged to break down the results depending on the product category. The saccharides proved to be highly predictable, being the expected fingerprint ranked Top 1 64% of the times, Top 5 in almost 75% of the times and Top 25 90% of the times. It was also found that alkaloids are ranked Top 1 more than 55% of the times and terpenes' expected output vector is within the Top 25 with a frequency of 85%. Saccharides' scores support the initial analysis on CDK fingerprints where several markers for this category had been found.

Afterwards, the neural network was run with different configurations for both DNA and amino acid k-mers. The scores once again were marginally different from those expected by using random inputs. These results support the absence of biological meaning in using only repetitions of words of length “k” to encapsulate the type of product encoded by a BGC.

The Keras neural network turned out to be significantly more effective by using Pfam counts as the input. However, despite having carried out a considerable number of experiments, its results are still distant from those obtained with the IOKR algorithm. The best scores were obtained by using 4 hidden layers of 50 neurons each and the Adam optimiser (other optimisers were shown to perform similarly nevertheless). The learning rate was set to 0.01 for 25 epochs, then to 0.003 during 35 epochs and finally to 0.001 for the last 20 epochs. Using this algorithm, the expected fingerprint is ranked Top 1 around 23% of the times, Top 10 almost 50% of the times and Top 25 around 63% of the times. Even though they are not as good as the IOKR ones, they are still significantly better than the baseline.

Finally, future work includes using other types of fingerprints which might improve the encouraging results already obtained with CDK fingerprints. It has been shown that they contain many markers for saccharides while other categories cannot be predicted with the same accuracy due to the lack of substructures specific to them. The MACC keys are the most promising ones as they got the best scores for IOKR whereas their neural network results were quite close to the CDK ones, considering that the neural network was not optimised for working with 166-bit output vectors (MACC) but with 306 ones (CDK). The MACC keys potential is even greater when estimating how the scores would increase if the 960 substructures from MDL were made public. However, it is not necessary to replace one type of fingerprints for another one. Arguably, it would be helpful to find different markers for each category in different sets of fingerprints and join all of them together into a single fingerprint.

Another area of improvement would be implementing a compact way of summarising the Pfam domains found in a BGC's sequence. There are several variations of many domains which likely carry out the same function even if the current algorithm is considering them different. It is yet to be found a systematic algorithm able to encapsulate all those similar Pfam domains under the same name to reduce the current 6159-bits-long input vector into a condensed one which would convey the same information but in a more consistent way.

Ultimately, this project might be combined with others which are aimed at predicting fingerprints from MS/MS spectra (Brouard *et al.*, 2016). A certain spectrum must share the same fingerprint which will be predicted from a BGC. Consequently, if both fields of research are further developed, in the future, it might be possible to predict which BGC is responsible for a certain spectrum obtained by analysing a sample with tandem mass spectrometry.

6. Conclusions

1st – Two genomic representations and two machine learning algorithms were used to check if it was possible to predict structural fingerprints from genomic data.

2nd – The best performance was obtained by using Pfam counts with the IOKR algorithm. The expected output is ranked Top 1 around 35% of the times. This score is much higher considering only saccharides (64%), due to specific markers present in CDK fingerprints.

3rd – Further work includes combining this project with related research in predicting fingerprints from MS/MS spectra.

7. Bibliography

Adamek, M., Spohn, M., Stegmann, E. and Ziemert, N. (2016). Mining Bacterial Genomes for Secondary Metabolite Gene Clusters. Methods in Molecular Biology, 1520 pp.23–47.

Avsec, Ž., Kreuzhuber, R., Israeli, J., Xu, N., Cheng, J., Shrikumar, A., Banerjee, A., Kim, D.S., Beier, T., Urban, L., Kundaje, A., Stegle, O. and Gagneur, J. (2019). The Kipoi repository accelerates community exchange and reuse of predictive models for genomics. Nature biotechnology, 37(6), pp.592–600.

Basile, A.O. and Ritchie, M.D. (2018). Informatics and machine learning to define the phenotype. Expert review of molecular diagnostics, 18(3), pp.219–226.

Brouard, C., Shen, H., Dührkop, K., d'Alché-Buc, F., Böcker, S. and Rousu, J. (2016). Fast metabolite identification with Input Output Kernel Regression. Bioinformatics, 32(12), pp. i28–i36.

Celesti, F., Celesti, A., Wan, J. and Villari, M. (2018). Why Deep Learning Is Changing the Way to Approach NGS Data Processing: A Review. IEEE reviews in biomedical engineering, 11, pp.68–76.

Cereto-Massagué, A., Ojeda, M.J., Valls, C., Mulero, M., Garcia-Vallvé, S. and Pujadas, G. (2015). Molecular fingerprint similarity search in virtual screening. Methods (San Diego, Calif.), 71, pp.58–63.

Chavali, A.K. and Rhee, S.Y. (2018). Bioinformatics tools for the identification of gene clusters that biosynthesize specialized metabolites. Briefings in bioinformatics, 19(5), pp.1022–1034.

Chen, K., Wei, Z., Zhang, Q., Wu, X., Rong, R., Lu, Z., Su, J., de Magalhães, J.P., Rigden, D.J. and Meng, J. (2019). WHISTLE: a high-accuracy map of the human N6-methyladenosine (m6A) epitranscriptome predicted using a machine learning approach. Nucleic Acids Research, 47(7), pp. e41–e41.

Chen, W., Feng, P. and Lin, H. (2012). Prediction of replication origins by calculating DNA structural properties. FEBS letters, 586(6), pp.934–8.

Dawit N. and Henkel, W. (2017). Prediction of Essential Genes based on Machine Learning and Information Theoretic Features. Conference: BIOINFORMATICS 2017 8th International Conference on Bioinformatics Models, Methods and Algorithms

Durant, J.L., Leland, B.A., Henry, D.R. and Nourse, J.G. (2002). Reoptimization of MDL Keys for Use in Drug Discovery. Journal of Chemical Information and Computer Sciences, 42(6), pp.1273–1280.

Eddy, S.R. (2009). A new generation of homology search tools based on probabilistic inference. Genome Informatics 23:205-211 2009.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. Nature, 521(7553), pp.436–44.

Lind, A.L., Wisecaver, J.H., Lameiras, C., Wiemann, P., Palmer, J.M., Keller, N.P., Rodrigues, F., Goldman, G.H. and Rokas, A. (2017). Drivers of genetic diversity in secondary metabolic gene clusters within a fungal species. PLOS Biology, 15(11), p.e2003583.

Macesic, N., Polubriaginof, F. and Tatonetti, N.P. (2017). Machine learning: novel bioinformatics approaches for combating antimicrobial resistance. Current opinion in infectious diseases, 30(6), pp.511–517.

Medema, M.H., Kottmann, R., Yilmaz, P., Cummings, M., Biggins, J.B., Blin, K., de Bruijn, I., Chooi, Y.H., Claesen, J., Coates, R.C., Cruz-Morales, P., Duddela, S., Düsterhus, S., Edwards, D.J., Fewer, D.P., Garg, N., Geiger, C., Gomez-Escribano, J.P., Greule, A., Hadjithomas, M., Haines, A.S., Helfrich, E.J.N., Hillwig, M.L., Ishida, K., Jones, A.C., Jones, C.S., Jungmann, K., Kegler, C., Kim, H.U., Kötter, P., Krug, D., Masschelein, J., Melnik, A.V., Mantovani, S.M., Monroe, E.A., Moore, M., Moss, N., Nützmann, H.-W., Pan, G., Pati, A., Petras, D., Reen, F.J., Rosconi, F., Rui, Z., Tian, Z., Tobias, N.J., Tsunematsu, Y., Wiemann, P., Wyckoff, E., Yan, X., Yim, G., Yu, F., Xie, Y., Aigle, B., Apel, A.K., Balibar, C.J., Balskus, E.P., Barona-Gómez, F., Bechthold, A., Bode, H.B., Borriss, R., Brady, S.F., Brakhage, A.A., Caffrey, P., Cheng, Y.-Q., Clardy, J., Cox, R.J., De Mot, R., Donadio, S., Donia, M.S., van der Donk, W.A., Dorrestein, P.C., Doyle, S., Driessen, A.J.M., Ehling-Schulz, M., Entian, K.-D., Fischbach, M.A., Gerwick, L., Gerwick, W.H., Gross, H., Gust, B., Hertweck, C., Höfte, M., Jensen, S.E., Ju, J., Katz, L., Kaysser, L., Klassen, J.L., Keller, N.P., Kormanec, J., Kuipers, O.P., Kuzuyama, T., Kyrpides, N.C., Kwon, H.-J., Lautru, S., Lavigne, R., Lee, C.Y., Linquan, B., Liu, X., Liu, W., Luzhetskyy, A., Mahmud, T., Mast, Y., Méndez, C., Metsä-Ketelä, M., Micklefield, J., Mitchell, D.A., Moore, B.S., Moreira, L.M., Müller, R., Neilan, B.A., Nett, M., Nielsen, J., O'Gara, F., Oikawa, H., Osbourn, A., Osburne, M.S., Ostash, B., Payne, S.M., Pernodet, J.-L., Petricek, M., Piel, J., Ploux, O., Raaijmakers, J.M., Salas, J.A., Schmitt, E.K., Scott, B., Seipke, R.F., Shen, B., Sherman, D.H., Sivonen, K., Smanski, M.J., Sosio, M., Stegmann, E., Süssmuth, R.D., Tahlan, K., Thomas, C.M., Tang, Y., Truman, A.W., Viaud, M., Walton, J.D., Walsh, C.T., Weber, T., van Wezel, G.P., Wilkinson, B., Willey, J.M., Wohlleben, W., Wright, G.D., Ziemert, N., Zhang, C., Zotchev, S.B., Breitling, R., Takano, E. and Glöckner, F.O. (2015). Minimum Information about a Biosynthetic Gene cluster. Nature Chemical Biology, 11(9), pp.625–631.

Min, S., Lee, B. and Yoon, S. (2017). Deep learning in bioinformatics. Briefings in bioinformatics,18(5), pp.851–869.

Miotto, R., Wang, F., Wang, S., Jiang, X. and Dudley, J.T. (2018). Deep learning for healthcare: review, opportunities and challenges. Briefings in bioinformatics, 19(6), pp.1236–1246.

Montazeri, M., Montazeri, M., Montazeri, M. and Beigzadeh, A. (2016). Machine learning models in breast cancer survival prediction. Technology and Health Care, 24(1), pp.31–42.

Muegge, I. and Mukherjee, P. (2016). An overview of molecular fingerprint similarity search in virtual screening. Expert opinion on drug discovery, 11(2), pp.137–48.

Park, S.H., Do, K.-H., Kim, S., Park, J.H. and Lim, Y.-S. (2019). What should medical students know about artificial intelligence in medicine? Journal of educational evaluation for health professions, 16, p.18.

Sander, T., Freyss, J., von Korff, M. and Rufener, C. (2015). DataWarrior: An Open-Source Program For Chemistry Aware Data Visualization And Analysis. Journal of Chemical Information and Modeling, 55(2), pp.460–473.

Schläpfer, P., Zhang, P., Wang, C., Kim, T., Banf, M., Chae, L., Dreher, K., Chavali, A.K., Nilo-Poyanco, R., Bernard, T., Kahn, D. and Rhee, S.Y. (2017). Genome-Wide Prediction of Metabolic Enzymes, Pathways, and Gene Clusters in Plants. Plant Physiology, 173(4), pp.2041–2059.

Signal, B., Gloss, B.S., Dinger, M.E. and Mercer, T.R. (2017). Machine learning annotation of human branchpoints. Bioinformatics, 34(6), pp.920–927.

Skinnider, M.A., Merwin, N.J., Johnston, C.W. and Magarvey, N.A. (2017). PRISM 3: expanded prediction of natural product chemical structures from microbial genomes. Nucleic Acids Research, 45(W1), pp. W49–W54.

Villebro, R., Shaw, S., Blin, K. and Weber, T. (2019). Sequence-based classification of type II polyketide synthase biosynthetic gene clusters for antiSMASH. Journal of Industrial Microbiology & Biotechnology, 46(3–4), pp.469–475.

Yin, Z., Ai, H., Zhang, L., Ren, G., Wang, Y., Zhao, Q. and Liu, H. (2019). Predicting the cytotoxicity of chemicals using ensemble learning methods and molecular fingerprints. Journal of applied toxicology: JAT, p.10.1002/jat.3785.

Zhang, Z., Pan, Z., Ying, Y., Xie, Z., Adhikari, S., Phillips, J., Carstens, R.P., Black, D.L., Wu, Y. and Xing, Y. (2019). Deep-learning augmented RNA-seq analysis of transcript splicing. Nature Methods, 16(4), pp.307–310.