BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 5 2014, pages 712–718
doi:10.1093/bioinformatics/btt602

Data and text mining

Advance Access publication October 21, 2013

Exploring high dimensional data with Butterfly: a novel classification algorithm based on discrete dynamical systems

Joseph Geraci[1,2,*], Moyez Dharsee[3], Paulo Nuin[1,3], Alexandria Haslehurst[1], Madhuri Koti[4], Harriet E. Feilotter[1] and Ken Evans[3]

[1]Department of Psychiatry, University Health Network, Toronto, [2]Department of Pathology and Molecular Medicine, Queen's University, Kingston, [3]Ontario Cancer Biomarker Network, Toronto and [4]Department of Biomedical and Molecular Sciences, Queen's University, Kingston, Ontario, Canada

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: We introduce a novel method for visualizing high dimensional data via a discrete dynamical system. This method provides a 2D representation of the relationship between subjects according to a set of variables without geometric projections, transformed axes or principal components. The algorithm exploits a memory-type mechanism inherent in a certain class of discrete dynamical systems collectively referred to as the chaos game that are closely related to iterative function systems. The goal of the algorithm was to create a human readable representation of high dimensional patient data that was capable of detecting unrevealed subclusters of patients from within anticipated classifications. This provides a mechanism to further pursue a more personalized exploration of pathology when used with medical data. For clustering and classification protocols, the dynamical system portion of the algorithm is designed to come after some feature selection filter and before some model evaluation (e.g. clustering accuracy) protocol. In the version given here, a univariate features selection step is performed (in practice more complex feature selection methods are used), a discrete dynamical system is driven by this reduced set of variables (which results in a set of 2D cluster models), these models are evaluated for their accuracy (according to a user-defined binary classification) and finally a visual representation of the top classification models are returned. Thus, in addition to the visualization component, this methodology can be used for both supervised and unsupervised machine learning as the top performing models are returned in the protocol we describe here.

Results: Butterfly, the algorithm we introduce and provide working code for, uses a discrete dynamical system to classify high dimensional data and provide a 2D representation of the relationship between subjects. We report results on three datasets (two in the article; one in the appendix) including a public lung cancer dataset that comes along with the included Butterfly R package. In the included R script, a univariate feature selection method is used for the dimension reduction step, but in the future we wish to use a more powerful multivariate feature reduction method based on neural networks (Kriesel, 2007).

Availability and implementation: A script written in R (designed to run on R studio) accompanies this article that implements this algorithm and is available at http://butterflygeraci.codeplex.com/. For details on the R package or for help installing the software refer to the accompanying document, Supporting Material and Appendix.

Contact: geraci.joseph@gmail.com

*To whom correspondence should be addressed.

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on March 19, 2013; revised on October 2, 2013; accepted on October 16, 2013

1 INTRODUCTION

There is an n-dimensional geometry associated with any given dataset consisting of any number of subjects and n variables. Depending on the choice of which variables to include or emphasize, the geometry changes. For example, consider a dataset consisting of two clusters, e.g. responders and non-responders to some treatment. Under optimal circumstances, a small set of variables (e.g. a subset of gene expression values) can be used to separate the groups. These selected variables determine the relationship between patients in a geometrical way: via the distance between patients. How can one explore this space in an efficient, accurate and reproducible way so that one can study different scenarios, i.e. different sets of variables and the corresponding relationships between subjects in two-dimensions?

Butterfly's most powerful feature is its ability to reveal subclusters of data points even within apparently homogeneous main clusters. Thus, Butterfly provides a tool for the bottom up exploration of data in addition to a procedure that can be used for standard training and testing protocols. Real-life datasets are invariably heterogeneous and many powerful non-linear data exploration tools overfit the data due to the presence of noise. Mainly, this is because conventional tools require the establishment of complex frontiers to separate classes. In the case of Butterfly, the data points are transformed into a 2D representative space, which is the attractor space of the dynamical system, and clusters are separated by straight lines. Thus, non-linear correlations are identified between features, whereas the relationship in space between subjects (e.g. patients) remains simple.

The main purpose of this work is to introduce Butterfly and its efficacy in handling molecular profiling datasets. We present it as a data exploration and machine learning classification tool that can be applied to high dimensional data following any feature reduction step, which brings down the dimensionality to around 50 or below. We do not discuss the technical nuances behind the algorithm nor compare it with other methods in this article. We introduce the algorithm and highlight its ability to accurately classify data. Future research will focus on Butterfly's ability to

conserve the relationships between high dimensional data points even after they are projected to a 2D plane. The main use of Butterfly in the research setting is the ability to view the relationship between subjects, e.g. patients, from the vantage of a set of ($\geq$5) interesting variables, with the specific goal of identifying homogeneous subgroups. Further, the identification of the main variables that lead to a particular classification is returned to the user.

Many of the modern non-linear clustering/dimensional reduction methods rely on the mathematics that drives principle component analysis (Abdi and Williams, 2010) or k-means (Larraaga *et al.*, 2006). Because the mathematics that drives Butterfly is fundamentally different from these methods, we have found that exploratory analyses based on Butterfly and one of these other approaches provides comprehensive analyses and that Butterfly complements these established approaches effectively.

2 BACKGROUND

The desire for accurate and fast classifiers has been inspired by the recent influx of high dimensional datasets from the medical sciences and other fields. Bioinformatics makes regular use of machine learning algorithms including decision trees, support vector machines and regression methods (Larraaga *et al.*, 2006), as they supply researchers with a powerful armament for building classifiers. An emerging challenge is the need to integrate different types of data including genomic [messenger RNA (mRNA), microRNA (miRNA) expression, epigenetic changes, copy number variants, single nucleotide polymorphism], proteomic (mass spectrometry) and clinical data. Platforms that could simultaneously consider heterogeneous variables in an unbiased way are going to be required to effectively use data produced from multiple high-throughput technologies such as microarrays, array comparative hybridization, high-throughput sequencing and mass spectrometry for the same individual. Each of these technologies is a different window on the genome, and a method that can effectively merge data from each of these sources will provide more information about our molecular machinery than any method that uses information from only a single source (Hamid *et al.*, 2009). Further, it is certain that this trend will continue as additional types of data are generated in studies, e.g. metabolomic. Thus, a method that is capable of efficiently working with high dimensional data regardless of data type is a desirable commodity.

2.1 Approach and technical preliminaries

Formally, a dynamical system is the triple (I, M, ϕ), where I is a subset of $\mathbb{R}$ (in the continuous case) or $\mathbb{Z}^+$ (in the discrete case) and M the space over which the mapping ϕ is defined, i.e. $\phi : I \times M \to M$. The interval I is to be thought of as time and this, as mentioned, can either be discrete or continuous, and M is the space where the variables are defined. Thus in general, one can envision ϕ as describing the motion of a point in the space M as a function of time. As an example one can consider the expression levels of a set of genes as time varies. In this case one could define the system by $\phi : (0, T) \times \mathbb{R}^n \to \mathbb{R}^n$. The n-dimensions real space, $\mathbb{R}^n$, represents the fact that one starts with initial expression values for the n genes as a real-valued vector $(i_1, i_2, \ldots, i_n)$, which evolves over time to the expression levels $(f_1, f_2, \ldots, f_n)$ at time T. ϕ contains the information regarding how these particular genes will be expressed from time 0 to time T.

The particular type of dynamical system that our classification algorithm is based on is referred to as an iterated function system (IFS). This is a type of discrete dynamical system in that the time evolution of the system occurs in discrete steps. Given a starting point (or points in general) in the space M, the IFS consists of a finite set of mappings that transforms the point. Each 'time step' corresponds to the selection of a function and its application to the last mapping of the point. The best way to elucidate the idea is to give a famous example that results in the Barnsley fern (Fig. 1). Here the mapping referred to as ϕ actually consists of the application of the following four transformations:

$$T_1(x, y) = (0.85x + 0.04y, -0.04x + 0.85y + 1.6)$$
$$T_2(x, y) = (0.2x + 0.26y, 0.23x + 0.22y + 1.6)$$
$$T_3(x, y) = (-0.15x + 0.28y, 0.26x + 0.24y + 0.44)$$
$$T_4(x, y) = (0, 0.16y)$$

Each of the T_i is a mapping from $\mathbb{R}^2 \to \mathbb{R}^2$ and at each time step one chooses one of these mappings and applies it to the point that results from the previous application of a chosen transformation. The passage of discrete time corresponds to iterating this process over and over starting from an initial point. To create the Barnsley fern, one starts with a point in the plane $\mathbb{R}^2$, say the origin (0,0), and then at each time step one of the T_i is chosen with a given probability. T_1 is chosen with a probability of 85%, T_2 and T_3 are chosen with a probability of 7% each and T_4 is chosen 1% of the time. One iterates the process and ends up with a chain of transformations starting with some mapping applied to (0,0) and then another applied to the result of that mapping and so on. Mathematically, after five iterations the chain could look like $T_2(T_1(T_1(T_4(T_1(0,0)))))$ that is another point in $\mathbb{R}^2$. To get the image in Figure 1, one keeps all the resulting points for 30 000 iterations.

This process of randomly iterating over functions is commonly referred to as the chaos game (Barnsley, 2006), and the resulting image of such a process is known as an attracting set in mathematics. Another interesting example of a chaos game that can be introduced non-technically is the following process that we shall refer to as the dice chaos game. Take a triangle and label one of the vertices 1 and 2 (12), the other vertex 3 and 4 (34) and the final one 5 and 6 (56). Now beginning on any vertex, say vertex (12), roll an unbiased die and observe the result. Say it is either a 3 or a 4. Find the midpoint between vertex (12) and (34)

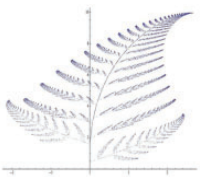

Fig. 1. The Barnsley fern created by iterating the transformation T_1–T_4 as described, 30 000 times

and mark that point and call it P1. Now repeat. Say you roll a 1 or 2. Find the midpoint between P1 and (12) and place a point there and call it P2 and continue. The attractor set for this process is known as the Sierpinski triangle (Fig. 2) and there is also an IFS to generate it.

2.2 Previous work

The algorithm we present is based on a dynamical system (Martelli, 1999) that is a novel instance of the chaos game. This particular algorithm was developed following the realization that one could drive data through a variety of dynamical systems and observe the resulting behavior. Some dynamical systems have the ability to capture patterns inherent in the data through a memory-type mechanism. An example of such a phenomenon will become clear after we present the details behind the algorithm.

There are several algorithms that are closely related to Butterfly. As a related aside, we first mention a class of algorithms that are close in spirit, in that they may be viewed as belonging to the class of data-driven dynamical system approaches. We believe that this approach has never been formalized, but there do exists a set of algorithms designed for data analysis based on cellular automata (Wolfram, 2002). These are discrete dynamical systems different from the ones used in Butterfly, however, we include some references for the interested reader (Fawcett, 2008; Ultsch, 2000, 2002).

The construction of Butterfly came about through an attempt to study protein and DNA sequences, far before we turned our sights to general data analysis. During the study of DNA sequences, several researchers have had the idea to replace the usual chaos game (described previously as the dice chaos game), with a square whose vertices are labeled A,G,C and T. One would then 'play' the chaos game with the DNA sequence of interest. The sequence would *drive* the dynamical system instead of a die. This is precisely what the author in Jeffrey (1990) presented many years before our efforts. This line of research led to the construction of a novel metric for protein sequences by the first author (unpublished), which eventually led to Butterfly.

Several publications exist dealing with the analysis of biological sequences such as Basu *et al*. (1997), Dutta and Das (1992) and Joseph and Sasikumar (2006) and recently Almeida *et al*. (2012) and Vinga *et al*. (2012). These articles capture the essential nature behind Butterfly: they all use the chaos game methodology and drive sequence data through it, via various innovations, to detect patterns. Our idea was to extend this beyond the analysis of sequence data to the analysis of general datasets in an essentially agnostic way. The main advancement we present is a construction that allows one to move from a 4our feature space (A,G,C,T) or 20 feature space (amino acids) to the analysis of general data consisting of many variables.

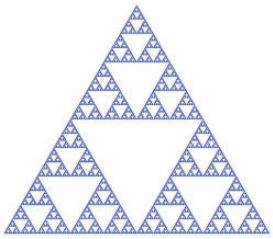

Fig. 2. The Sierpinski triangle is the attracting set of the triangle dice chaos game

Our algorithm is also distantly related to methods developed from the perspective of functional data analysis (FDA) (Chen *et al*., 2011; Wu and Mller, 2010), which was originally created to statistically analyze continuous data. These two articles represent recent advances that take advantage of the machinery inherent in the field of FDA (Ramsay and Silverman, 2005), by converting high dimensional data to functional data (imagine curves in some space that the data are embedded in). Even though these methods are not based on dynamical systems, they do use functional analysis. Like Butterfly, these methods use the mathematics of functions to analyze high dimensional data. The main difference, however, is that Butterfly uses a discrete dynamical system that is essentially an IFS to process discretized data, and these FDA-based methods translate data into functions and then use the methods established in FDA, e.g. functional principal component analysis (Shang, 2013).

3 METHODS

We report results on three datasets: a synthetic dataset that we created in Mathematica (shown and described in the Supporting Material and Appendices), a publicly available gene expression lung cancer dataset and an integrated ovarian cancer dataset that our group at Queen's University generated and analyzed.

The lung cancer dataset (gene expression microarray data) we used is publicly available at GSE10245 (2009) and GSE18842 (2010). From GSE10245 (2009), we extracted 40 lung cancer patient samples with histological subtype adenocarcinoma (AC) and 18 with histological subtype squamous cell carcinoma (SCC). From GSE18842 (2010), we extracted 14 ACs and 32 SCCs for a total of 54 ACs and 50 SCCs.

The ovarian dataset was generated by our group. Ethics approval was obtained from Queen's University and the Ottawa Health Research Institute Research Ethics Boards. Tumor samples were obtained from the Division of Gynecologic Oncology Ovarian Tissue Bank and the Ontario Tumor Bank. All tumors were chemonaive at the time of collection. Post debulking surgery, patients received combination chemotherapy with carboplatin and paclitaxel. Histological classification of the tumors was performed following World Health Organization criteria, and disease stage was determined according to the International Federation of Gynecology and Obstetrics guidelines. Fourteen tumours were classified as chemosensitive (progression-free interval of >18 months), and 11 tumours were classified as chemoresistant (progression-free interval of <8 months). Histopathological examination of tumor sections identified >70% tumor in all samples.

For the gene expression profiling, total RNA was isolated from tumour samples using a combination of Trizol (Invitrogen, CA, USA) and Qiagen RNA isolation kit (Qiagen Inc., Mississauga, CA, USA) as per manufacturer's instructions. RNA integrity was assessed using RNA 6000 nano chips on a Agilent 2100 Bioanalyzer (Agilent Technologies, USA) and concentration was determined using a NanoDrop ND-100 spectrophotometer (NanoDrop Technologies, USA). All downstream microarray analysis was completed using Affymetrix Human Genome U 133 Plus 2.0 arrays (Affymetrix Inc., USA) as per manufacturer's instructions, at the Centre for Applied Genomics (The Hospital for Sick Children, Toronto, ON, Canada). Our miRNA data were derived from the Human miRNA Microarray Kit (V3) and the methylation data from Human Promoter 1.0R Array.

3.1 The algorithm

A dataset is assumed to be an $n \times N$ matrix where there are N subjects (patients for example) and n features (genes for example). Each entry of the matrix will have a number indicating a level (continuous or discrete, i.e. low, medium, etc) or the presence/absence of a feature (single nucleotide polymorphisms for example). For this description of Butterfly, we are assuming we have continuous data.

We provide an outline of the algorithm below. When reviewing the outline, recall the previously described *dice chaos game* on the triangle. The chaos game dynamical system, whose attracting set is the Sierpinski triangle, has an alphabet consisting of $\{1, 2, 3, 4, 5, 6\}$, which correspond to the six faces of a die. Instead of this alphabet, we will be interested in a way to capture the nature of general datasets. In the simple triangle chaos game, each vertex of the triangle is assigned two numbers from the alphabet $\{1, 2, 3, 4, 5, 6\}$, e.g. $\{1, 2\} \rightarrow$ vertex1. The previously cited articles that used the chaos game either dealt with the alphabet $\{A, G, C, T\}$ or the 20 amino acids. The mappings to a vertex of some polygon is straightforward in these cases, e.g. each nucleic acid can be mapped to a vertex of a square in a one to one way and in the cases of the amino acids, one could map several amino acids to the same vertex or use an icosagon for example.

In our case, the first issue is to understand how we are to map our data to some space so that data can drive the chaos game. We chose to divide our data into eight quantiles and label these from lowest to highest $\{0, 1, 2, \ldots, 7\}$. Each feature (dimension) will need to be considered and each can be in one of the eight states. Consider f_i (feature i) for example, and note that, for a particular subject, it will be in one of eight quantiles. Thus, the geometry that the system will run this on will need to have enough space for an alphabet of size $n \times 8$. Choosing a polygon with that many vertices turns out not to be effective. The main issue with choosing an appropriate geometry to accommodate all these features is that after the algorithm projects the data to a 2D surface, one requires these projected points to cluster in sections that are far enough from each other so that they are distinguishable. Mathematically, the geometry chosen determines the characteristics of the 2D space to which the high dimensional points (subjects) are mapped. Thus, we need a way to ensure that points will cluster near other similar points and far enough away from dissimilar points so that they are classified correctly. In other words, we need to ensure that the clustering is discriminating.

3.1.1 The underlying geometry The geometry we use is given in Figure 3 and is to be placed on $\mathbb{R}^2$ so that each node is given a 2D coordinate. This shape captures the following information: the large nodes labeled from L (low) to H (high) represent the level of the features (the eight quantiles) and around each of these there are smaller star shaped 'satellite' nodes, which are the actual features, e.g. genes, proteins, financial predictors and so forth. The blue path in three shows a hypothetical dynamical path that some data impose on the system. It begins at some feature that has a medium low (L1) magnitude (relative to that feature's level across all subjects) then moves to some other feature that has a high medium (M2) magnitude and continues. We ensure that adjacent quantiles are not nearest neighbors in this geometry as this ensures that differences are captured by the dynamical system. Butterfly does not use the usual divide by two rule of the classical chaos game, but uses a rule that changes as the data are processed. For a given fixed permutation of the features, the system proceeds and can be terminated after any number of steps. Butterfly terminates the search after every feature, evaluates the clustering and continues. Our in house version returns the top clustering results. An example will make this clearer.

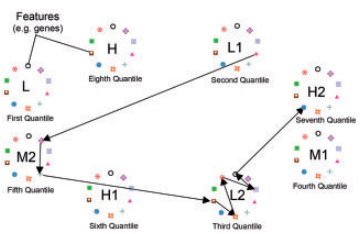

Fig. 3. This represents the geometry that the chaos game is played on in Butterfly

Suppose we have a system that proceeds as indicated in Figure 3 for a particular subject. The string that Butterfly processes in this case is given by *f*1*L*1.*f*2*M*2.*f*3*M*2.*f*4*L*2.*f*5*L*2.*f*6*L*2.*f*7*L*2.*f*8*L*2.*f*9*H*2, where dots are used to separate the steps (or characters that comprise the word). *f*3*M*2, for example, just represents feature three in the fifth quantile. This string of variables, coupled with the quantile that maps from its level (or category), informs the dynamical system's next move. Each character causes the dynamical system to move toward the corresponding labels in the geometry illustrated in Figure 3, according to the system's mathematical rules. Imagine that each character (e.g. *f*3*M*2) tells a virtual ball (corresponding to one patient or subject) to move toward the node labeled as feature *f* in quantile *M*2. This is repeated for every character (i.e. variable or feature). A snapshot of this virtual ball's position in this geometry is recorded for every step/character/feature, until features run out. This is performed for all subjects and the result for every termination point (e.g. the snapshot at a feature) will be a 2D clustering of all subjects.

The algorithm assesses how well the subjects cluster beginning with the third feature as it requires the first two features to initiate. Thus, in the given example, it processes and terminates at *f*3*M*2, evaluate how well the subjects cluster and then continue to process *f*4*L*2, as this is the next step in the string. We shall refer to this cutoff as the *CO-pt (Cut-Off point)*. Again it will evaluate how well the subjects cluster and continue until the string terminates. The number of elements in the string is determined by how much preliminary feature selection Butterfly is asked to perform. In the lung cancer example presented later, 77 genes remained after the initial feature selection step. Butterfly found a model including 13 of these genes which, provided an excellent clustering. Thus, a string of length 77 was input and Butterfly returned 11 genes after it found a clustering with an overall accuracy of 90%.

The actual choice of the geometry given in Figure 3 was constructed in such a way as to allow for clusters to form far enough from each other so that a cluster discriminating algorithm would be able to easily score different models. The underlying geometry defines the space that the dynamical system drives the data through and therefore where clusters will form. The algorithm moves subjects through this space according to the corresponding set of variables, and subjects with similar patterns of variables will arrive at similar areas in the defined geometry. The particular configuration given in Figure 3 is one of many possible configurations. We encourage others to experiment with different configurations and supply this one as a proof of principle to the overall technique and as one that has been highly effective in practice.

A vital issue is with consistency. As long as the same geometry is used for all subjects, then an evaluation of the clusters is possible. The relationship between different underlying geometries and the kinds of clustering that will occur requires further exploration. Our particular construction was arrived at by a desire to handle many variables while maintaining the memory-type mechanism inherent in the chaos game. What are recorded in this particular manifestation of Butterfly are the final points of the subject's dynamics. Owing to the memory mechanism of this dynamical system, the final point corresponds to the particular values of the variables: this is precisely where the dimension reduction occurs.

Note that many more genes can be included, but the feature space will be larger and will take longer to explore before acceptable clusters are discovered and a set of discriminating features extracted. Different permutations of the order that the features are input into Butterfly allow one

to explore different combinations of genes. It is clear that it would take an impossible amount of time to input every possible permutation in the order of a set of features. However, it appears that this is not necessary with Butterfly to extract some meaningful information from data. This has to be studied further and it is beyond the scope of this article, however, we shall make an effort to clarify this matter. Using a filter feature selection method, such as our univariate approach, is common practice and one can use this simple method or a more sophisticated procedure for selecting sets of features. The main point is that the permutation approach would be inordinately expensive for a large number of variables, and thus this particular method works well when the feature selection filter returns between 30–50 variables. Of course the algorithm will not attempt to visit all 10^{32} permutations of a list of 30 variables. With just 50 different random permutations, the more successful models can be extracted and the feature sets examined, compared and reduced. What makes this possible is the aforementioned 'memory' that this particular dynamical system exhibits: the string of variables can be looked at as existing in chunks of length d, and if a variable appears in one of these chunks [or regions of influence (ROIs), see Fig. 4] it influences the configuration of the dynamical system in accordance with how close it is to the CO-pt. This provides a massive reduction in the number of permutations required to find a set of features that differentiates the given groups. In practice, the models that do best are extracted and their features truncated to within a few ROIs of the CO-pt, and the process repeated. This process is iterated to extract a set of features that together can differentiate the groups of interest.

An overview of the Butterfly Algorithm

- Run a feature selection method on the data and reduce the number of features to F, which is selected by the user.
- Create a new dataset with the reduced number of features. Note that this may be a dataset constructed from different sources, e.g. mRNA, methylation, miRNA, neuroimaging and so forth.
- Discretize the data into quantiles, i.e. each feature gets ranked across all subjects. We have found using eight quantiles is effective in practice.
- Map each subject to a string that captures the feature and quantile information as in our example earlier mentioned in the text:

$$f1L1.f2M2.f3M2.f4L2.f5L2.f6L2.f7L2.f8L2.f9H2$$

- Each word segment or character, e.g. $f6L2$, is given a 2D coordinate in $\mathbb{R}^2$ on the space given in Figure 3.
- Run the aforementioned dynamical system. Each subject value drives the path of a curve through the geometry given in Figure 3.
- An array, *Models*, records different final values achieved at different feature randomizations and different features to 'cut-off' the dynamics or progression of the path. More clearly, the process is halted at some CO-pt for all patients. That is, a feature is chosen to terminate the process. The closest preceding features before the CO-pt are the most significant, as they have had the most influence.
- The final points for each subject, which corresponds to the (x, y) coordinate of the dynamical system achieved at the CO-pt, is recorded. Thus, the aforementioned array *Models* are constructed and holds all models that correspond to feature randomization and different CO-pts.
- An additional method is used to evaluate the models via how well the data clusters. A linear support vector machine (SVM) (Larraaga *et al.*, 2006) is performed on all models in *Models* and evaluated via cross-validation (other methods have been used as well and an alternate version of Butterfly will be made available). The top models, according to the training set, are returned to the user, along with the corresponding top features, i.e. the features closest to the particular CO-pt for each model.

The complexity of this algorithm assuming that a dataset has N subjects and n variables is $O((N \cdot n)^2)$ and is amenable to parallelization. Please refer to the Supporting Materials and Appendices for more details on the complexity and notes regarding using the accompanying R package, instructions on how to run it on RStudio and other technical issues.

4 RESULTS ON DATA

For a description and visualization of the synthetic dataset, please see the Supporting Materials and Appendices Section.

4.1 Lung cancer data

After running Butterfly on the aforementioned lung cancer dataset, we arrived at a model illustrated in Figure 5. For the purpose of this article, we first ran a univariate feature selection algorithm [a Matlab version is freely available at (http://matlabdatamining.blogspot.ca/2006/12/feature- selection- phase-1- eliminate.html)], used the top 77 discriminating features, then used Butterfly to identify a model using the following genes: GBP6, CLCA2, ATP11A, COL4A6, KRT5, KRT6B, XXYLT1 (C3orf21), TRIM29 and SOX2 where the importance of influence increases as you proceed through the list. Each of the genes on this list has been associated with cancer (e.g. Lu, 2010; Zhou, 2012) providing face validity for this simplified version, despite the fact that it relies heavily on finding discriminating univariate features. Butterfly has the ability to quickly project high dimensional data onto 2D data while still preserving the relationships between the patients. This means that in its simplest form, one searches through many models (of low dimension) and extract those that score highly on a training set, and thus extract features on the basis of which features are nearest to the particular CO-pt, mentioned in Section 3.1. Note that the separation between groups is achieved by a straight line. Even though the relationship between patients and variables are non-linear, the resulting 2D model is simple and thus more likely to generalize.

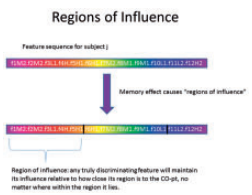

Fig. 4. Sets of features become chunked into ROIs

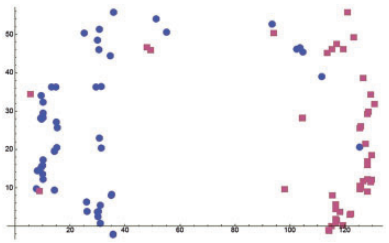

Fig. 5. The result of running Butterfly on a 77 gene lung cancer dataset

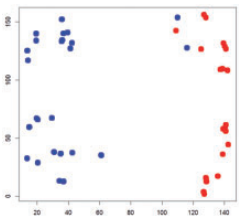

Fig. 6. A model produced on a 48 patient lung cancer dataset. Blue indicates AC cases, and red indicates SCC cases

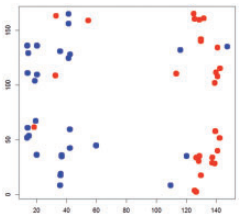

Fig. 7. The model shown in Figure 6 applied to a test set consisting of 28 AC cases and 28 SCC cases

To demonstrate training and test set reproducibility, we include two figures (Figs. 6 and 7) arrived at by splitting the above dataset into a training and test set. The training set consists of 26 AC cases and 22 SCC cases, and the test set consists of 28 cases of each type. This model had an overall accuracy of over 95% on the training set, as it misclassified two AC cases. The model performed with an accuracy of 85% on the test set, as it misclassified four cases in each type. One can download butterflyTest.R from http://butterflygeraci.codeplex.com/SourceControl/latest to apply a particular model to a test set found from running ButteRfly.R on a training set.

4.2 Ovarian cancer

Butterfly is effective with integrated datasets consisting of different types of data. Our goal was to extract relevant biological information with respect to how well 25 ovarian cancer patients responded to cisplatin treatment. Our dataset consists of just over 3300 methylation probes (after a reduction), ~900 miRNAs and ~19 000 genes. Using various univariate feature selection tools, we assembled a dataset of 356 features coming from each modality, i.e. 124 mRNA features, 82 miRNA features and 150 methylation probe features. Various models were constructed and their features extracted. Here we report on the model that is given in Fig. 8 where it correctly separated 9 of 11 responders and 12 of 13 non-responders. The model was constructed by inputing this 356 feature dataset into our algorithm and visually exploring the top models that were output.

The feature set we extracted from the model given in Figure 8 was as follows:

- miR-142 (micro-RNA)
- CEP192 (gene)
- DLG2 (gene)
- CENPQ (gene)

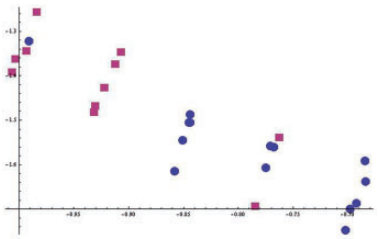

Fig. 8. Integrating mRNA, miRNA and methylation data for an ovarian cancer project. The red points indicate 11 women who responded favorably to cisplatin treatment, and the blue represents 14 women who did not

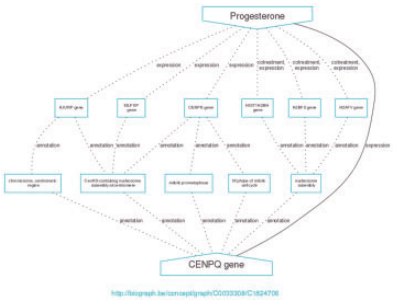

Fig. 9. Diagram of CENPQ's involvement in various biological processes related to cancer. *Reference: http://biograph.be/concept/graph/C0033308/C182470*

in addition to four methylation probes that are still being studied. These features have been implicated in the literature and even have relevance for chemoresistance (Andreopoulos, 2012; Gomez-Ferreria *et al.*, 2007; Sahab *et al.*, 2010). If one refers to Figure 9, one can see the various ways CENPQ can affect cancer progression, including through the mitotic stages.

5 CONCLUSION

In this article we presented an algorithm for the visualization, clustering and classification of data that is based on a simple discrete dynamical system. The novel aspect of the classification protocol is the dynamical system portion sandwiched between a feature selection filter and model evaluation algorithm. The iterative dynamical system transforms a multidimensional dataset into a 2D representation without using typical geometric and algebraic approaches found in other methods. Thus, Butterfly is ultimately a data visualization tool that allows researchers to probe their data by reviewing a set of high ranking models and observing the relationships between subjects in a 2D space. Each model is based on giving priority to a set of features and providing a human readable representation of the resulting relationships between subjects. However, our methodology allows for both supervised and unsupervised machine learning. The main benefits of using Butterfly to build predictive models is that only a simple linear split in a 2D space is required to separate groups, and this minimizes the chances of over fitting noise in data and its ability to provide a human readable representation of the structure of the data including subclusters within expected

clusters. The benefit over support vector machines (Larraaga *et al*., 2006) is that with Butterfly this is always done in a 2D space, where as the support vector machine has to establish the separating contour in a space whose dimension is equal to the number of variables considered. Butterfly has a lot in common superficially with principal components analysis (Abdi and Williams, 2010), but they are sufficiently different mathematically to justify using both as data exploration tools. In fact, principal components analysis can be used as a variable reduction tool before using Butterfly to examine subgroups. Butterfly's ability as an effective multivariate feature selection tool will be the subject of a future publication.

ACKNOWLEDGEMENTS

Joseph Geraci thanks Dr Igor Jurisica at the Ontario Cancer Institute who helped him to initiate the project that eventually transformed into Butterfly. His support and advice were invaluable. He also thanks Dr Sidney Kennedy and Dr Jonathan Downar who were supportive of this endeavor during his time as a fellow in the Department of Psychiatry at the University Health Network. He also thanks Dr Jeremy Squire for allowing him access to the integrated ovarian cancer dataset. The authors would also like to thank the ovarian cancer patients who have donated tumour to the Division of Gynecologic Oncology Ovarian Tissue Bank at the Ottawa Hospital Research Institute.

Funding: NSERC IRDF 2010-2012, Eli Lilly Canada Fellowship 2012 - 2014, Ontario Brain Institute and The Buchan Family Foundation. The Ontario Institute for Cancer Research through funding provided by the Government of Ontario.

Conflict of Interest: none declared.

REFERENCES

Abdi,H. and Williams,L. (2010) Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat*., **2**, 433–459.

Almeida,J. *et al*. (2012) Fractal mapreduce decomposition of sequence alignment. *Algorithms Mol. Biol*, **7**, 12.

Andreopoulos,B.A.D. (2012) Integrated analysis reveals hsa-mir-142 as a representative of a lymphocyte-specific gene expression and methylation signature. *Cancer Inform*., **11**, 61–75.

Barnsley,M. (2006) *Super Fractals*. Cambridge University Press, New York.

Basu,S. *et al*. (1997) Chaos game representation of proteins. *J. Mol. Graph Model*., **15**, 279–289.

Chen,K. *et al*. (2011) Stringing high-dimensional data for functional analysis. *J. Am. Stat. Assoc*., **106**, 275–284.

Dutta,C. and Das,J. (1992) Mathematical characterization of chaos game representation. new algorithms for nucleotide sequence analysis. *J. Mol. Biol*., **228**, 715–719.

Fawcett,T. (2008) Data mining with cellular automata. *ACM SIGKDD Explor. Newslett*, **10**, 32–39.

Gomez-Ferreria,M. *et al*. (2007) Human cep192 is required for mitotic centrosome and spindle assembly. *Curr. Biol*., **17**, 1960–1966.

GSE10245. (2009) Geo lung cancer data set - gse10245, http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE10245 (18 March 2013, date last accessed).

GSE18842. (2010) Geo lung cancer data set - gse18842, http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE18842 (18 March 2013, date last accessed).

Hamid,J. *et al*. (2009) Data integration in genetics and genomics: methods and challenges. *Hum. Genomics Proteomics*, **2009**, doi:10.4061/2009/869093.

Jeffrey,H. (1990) Chaos game representation of gene structure. *Nucleic Acids Res*., **18**, 2163–2170.

Joseph,J. and Sasikumar,R. (2006) Chaos game representation for comparison of whole genomes. *BMC Bioinformatics*, **7**, 243.

Kriesel,D. (2007) *A Brief Introduction to Neural Networks*. http://www.dkriesel.com (17 April 2013, date last accessed).

Larraaga,P. *et al*. (2006) Machine learning in bioinformatics. *Brief Bioinform*, **7**, 86–112.

Lu,Y. *et al*. (2010) Evidence that sox2 overexpression is oncogenic in the lung. *PLoS One*, **5**, e11022.

Martelli,M. (1999) *Introduction to Discrete Dynamical Systems and Chaos*. Wiley-Interscience, Hoboken, New Jersey, U.S.A.

Ramsay,J. and Silverman,B. (2005) *Functional Data Analysis*. 2nd edn. Springer, New York.

Sahab,Z. *et al*. (2010) Tumor suppressor rarres1 regulates dlg2, pp2a, vcp, eb1, and ankrd26. *J. Cancer*, **1**, 14–22.

Shang,H. (2013) *A survey of functional principal component analysis*. AStA Advances in Statistical Analysis, Springer.

Ultsch,A. (2000) An artificial life approach to data mining. In: *Proceedings of European Meeting of Cybernetics and Systems Research*, Wein.

Ultsch,A. (2002) Data mining as an application for artificial life. In: *Proceedings of Fifth German Workshop on Artificial Life*. pp. 191–197.

Vinga,S. *et al*. (2012) Pattern matching through chaos game representation: bridging numerical and discrete data structures for biological sequence analysis. *Algorithms Mol. Biol*., **7**, 10.

Wolfram,S. (2002) *A New Kind of Science*. Wolfram Media Inc., Champaign, IL.

Wu,P. and Mller,H. (2010) Functional embedding for the classification of gene expression profiles. *Bioinformatics*, **26**, 509–517.

Zhou,Z.Y. *et al*. (2012) Significance of trim29 and β-catenin expression in non-small-cell lung cancer. *J. Chin. Med. Assoc*., **75**, 296–74.