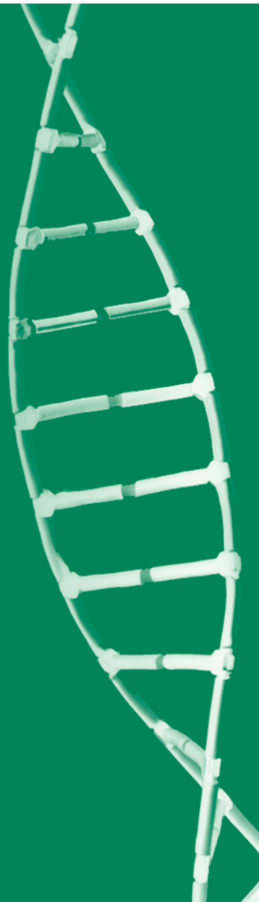
Milan Češka
David Šafránek (Eds.)

LNBI 11095

Computational Methods in Systems Biology

16th International Conference, CMSB 2018
Brno, Czech Republic, September 12–14, 2018
Proceedings



Springer

Lecture Notes in Bioinformatics 11095

Subseries of Lecture Notes in Computer Science

Milan Češka · David Šafránek (Eds.)

Computational Methods in Systems Biology

16th International Conference, CMSB 2018
Brno, Czech Republic, September 12–14, 2018
Proceedings

Springer

Editors
Milan Češka
Brno University of Technology
Brno
Czech Republic

David Šafránek
Masaryk University
Brno
Czech Republic

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Bioinformatics
ISBN 978-3-319-99428-4 ISBN 978-3-319-99429-1 (eBook)
https://doi.org/10.1007/978-3-319-99429-1

Library of Congress Control Number: 2018951890

LNCS Sublibrary: SL8 – Bioinformatics

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Inferring Mechanism of Action of an Unknown Compound from Time Series Omics Data

Akos Vertes[1], Albert-Baskar Arul[1], Peter Avar[1], Andrew R. Korte[1], Hang Li[1], Peter Nemes[1], Lida Parvin[1], Sylwia Stopka[1], Sunil Hwang[1], Ziad J. Sahab[1], Linwen Zhang[1], Deborah I. Bunin[2], Merrill Knapp[2], Andrew Poggio[2], Mark-Oliver Stehr[2], Carolyn L. Talcott[2](✉), Brian M. Davis[3], Sean R. Dinn[3], Christine A. Morton[3], Christopher J. Sevinsky[3], and Maria I. Zavodszky[3]

[1] Department of Chemistry, George Washington University,
Washington DC 20052, USA
[2] SRI International, Menlo Park, CA 94025, USA
clt@csl.sri.com
[3] GE Global Research, Niskayuna, NY 12309, USA

Abstract. Identifying the mechanism of action (MoA) of an unknown, possibly novel, substance (chemical, protein, or pathogen) is a significant challenge. Biologists typically spend years working out the MoA for known compounds. MoA determination is especially challenging if there is no prior knowledge and if there is an urgent need to understand the mechanism for rapid treatment and/or prevention of global health emergencies. In this paper, we describe a data analysis approach using Gaussian processes and machine learning techniques to infer components of the MoA of an unknown agent from time series transcriptomics, proteomics, and metabolomics data.

The work was performed as part of the DARPA Rapid Threat Assessment program, where the challenge was to identify the MoA of a potential threat agent in 30 days or less, using only project generated data, with no recourse to pre-existing databases or published literature.

1 Introduction

Lethal chemical and biological agents could be introduced through deliberate malevolent activity or inadvertent release into the environment. Technologies that enable rapid characterization of the mechanism of action (MoA) of these agents would facilitate the identification of countermeasures, such as a known antidote or antagonist for a specific cellular pathway. Likewise, characterizing the MoA of pathogenicity of a virus or bacterium could uncover host/microbial interactions that could be exploited to control the infection.

Sponsored by the US Army Research Office and the Defense Advanced Research Projects Agency; accomplished under Cooperative Agreement W911NF-14-2-0020.

M. Češka and D. Šafránek (Eds.): CMSB 2018, LNBI 11095, pp. 238–255, 2018.
https://doi.org/10.1007/978-3-319-99429-1_14

A simple representation of this problem is the exposure of immortalized or primary human cells to the agent in question. Although this model system does not possess the complexity of the response by an organ or an entire organism, it still retains much of the challenges associated with trying to infer the MoA from perturbing biochemical pathways in a living system. At the cellular level, there are many potential mechanisms affecting cellular processes, such as regulation of the cell cycle, transcription/translation, metabolism, and signaling. The molecules that modulate these processes are diverse and their interactions are complex. Thus, a systems biology approach, i.e., untargeted data collection from multiple data types (such as comparative analysis of quantitative changes in the transcriptome [8,28,30], proteome [11,14,23], and metabolome [4,5,22] is critical for elucidating the MoA [20]. Furthermore, mechanistic events induced by exposure to the agent may vary over a wide range of timescales following the challenge; some events occur in seconds to minutes, others take hours to days to manifest. Thus, it becomes important to gather timeseries data in order to observe the unfolding of complex molecular changes in the cells.

These considerations motivate a multi-omics timeseries approach to gather experimental data at the molecular level. Following data acquisition, robust analytical and network building methods [2,3,10] are used to identify components of the MoA and their relationships. A primary goal is to identify biomolecules with significantly altered abundances, as these may represent potential participants in the MoA. Another goal is to distinguish the agent induced MoA from normal metabolic activity in a population of cells. There are two main approaches to arrive at MoA candidates from comparative multi-omics data. Knowledge-based methods, e.g., Ingenuity® Pathway Analysis (Qiagen) and PANTHER, rely on combining extensive curated information extracted from biomedical literature with enrichment analysis of the transcriptomic, proteomic and metabolomic profile changes as a result of the experimental perturbation [13,17]. The algorithms operating on the knowledge base are used to infer the causal networks underlying the user data, potential upstream regulators, and potential downstream effects. Although these methods show strong promise for the identification of the MoA, they are less useful when the agent is novel because of the scarcity or absence of curated information. In the absence of reliable knowledge base content for the studied perturbing agent, de novo approaches can be followed. These can rely on the analysis of static interaction networks, such as the DeMAND method [29], or on cross-correlation analysis between the time courses of the differentially regulated compounds identified in the omics data, e.g., ProTINA [18]. After an array of potential MoAs is identified, downselection to the potential mechanism is performed through confirmatory experiments to quantify effects on specific pathways and/or cellular responses to the agent.

In this paper, we describe data analysis algorithms and workflows developed to address the need to identify MoA components based purely on comparative multi-omics time course data. We applied a broad untargeted approach quantifying changes in the transcriptome, proteome and metabolome, using challenge agents with well-characterized MoAs, to develop and verify the analytic tools and approaches that would enable rapid MoA discovery. We developed robust techniques for high throughput, high coverage multi-omics data generation and data centric analysis. The data analyses were carried out with team members blinded to agent identity during multiple testing periods. A description of experimental methods and early results have been reported in prior conference proceedings [24–27]. Here, we detail the approaches used for data analysis, specifically those approaches based on generating Gaussian process models from time series data. Multiple approaches for identifying MoA components allowed us to cover MoAs with diverse characteristics by looking at the data in multiple ways.

Plan. In Sect. 2 we describe the Gaussian process model and analysis workflows based on this model. Section 3 briefly describes the data to be analyzed. Section 4 illustrates application of the analysis workflow to identify candidate MoA participants. Section 5 summarizes, compares to related work, and discusses future directions.

2 Gaussian Process-Based Analysis Workflow

In the following we present some core parts of our data analysis workflow. The overall workflow has been represented and automated using *JupyterFlow*, an in-house tool developed at SRI International that is based on Petri nets (generalized dataflow graphs) [7] and integrates features of interactive computing (Jupyter/Python Notebooks) [12] with data-driven distributed and parallel execution on heterogenous clusters, typically with GPUs (Graphical Processing Units) on a subset of the nodes. JupyterFlow integrates with Tensor Flow [1], which we use both for estimating Gaussian Process Models [6] and for training Neural Networks (such as the Autoencoders discussed below).

JupyterFlow supports multiple modes of operations: the parallel execution of workflows on powerful GPU servers, the distributed execution on heterogeneous clusters and the cloud (where GPUs may be only available on some machines), and a mode of disconnected operation, where laptops may occasionally synchronize with the main workflow, but otherwise can work independently (without continuous network connectivity). Generally, the workflow executes automatically as a collection of computational notebooks with automatically inferred dataflow dependencies and resource constraints

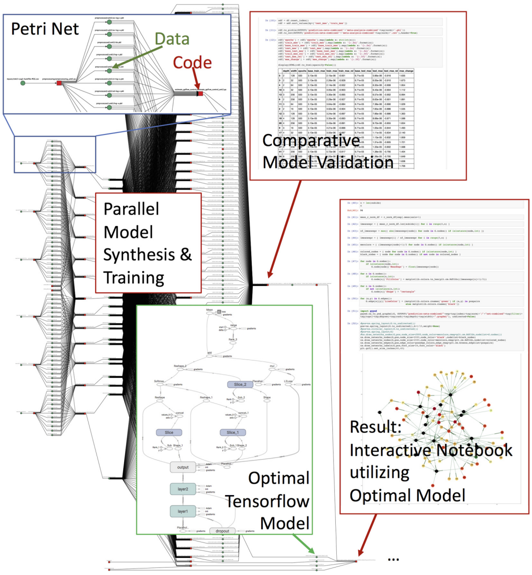

Petri Net
Data
Code
Comparative Model Validation
Parallel Model Synthesis & Training
Result: Interactive Notebook utilizing Optimal Model
Optimal Tensorflow Model
...

Fig. 1. Small excerpt of the RTA workflow in JupyterFlow. Each transition in the Petri net is defined by a Python/Jupyter notebook.

(e.g., memory, number of GPUs), which are essential to optimize the location of computations in the cluster. In addition, each notebook can also be executed interactively (e.g., for diagnosis and experimentation). Just as TensorFlow utilizes a form of dataflow graphs in the small, JupyterFlow utilizes dataflow graphs in the large. In the case of RTA, our current workflow (see Fig. 1 for a small excerpt) consists of ≈10,000 notebooks utilizing Python and TensorFlow that give rise to ≈270,000 metalevel dataflow dependencies (this is not counting dependencies at the TensorFlow level). We are currently executing this workflow on SRI's private cluster (with up to 4 GPUs per node) and in the Google Cloud (using large instances with up to 48 CPUs and 8 GPUs).

2.1 Gaussian Processes and What They Accomplish

A Gaussian process [21] represents a *probability distribution* over a *parameterized class of continuous functions*. The projection at each time point is Gaussian, i.e., normally distributed, with a *mean and variance that is explicitly represented* in the Gaussian process model. A time series of observations is defined by sampling at a finite number of arbitrary time points.

Gaussian process modeling allows us to address several challenges when facing high-dimensional, very noisy biological time series data. First, observations are only available at a small number of irregularly spaced time points. Also the number of observations at each time point may vary. By exploiting the continuity of the underlying functions, Gaussian processes allow us to benefit from the local structure to estimate the mean and to provide an explicit estimate of variance at each time point. The second challenge arises in the integration of multiple data sources, in our case we have transcriptomic, proteomic, and metabolomic data, each with different sampling time points, and different amounts of measurement noise. Again, the Gaussian processes allow us to integrate all data into a uniform continuous time scale and allow us to build specialized models for each data source to reflect the *biological variation* and the *measurement error*.

Specifically, we use *GPflow* [6], a library based on Tensor Flow, to estimate Gaussian processes separately for each type of data, for each compound (transcript, protein, or metabolite), and for treated and control condition. All observed quantities are uniformly represented in log2 scale. Furthermore, we use a log-based time scale to reflect the approximate spacing of observations with more observations closer to the beginning of the experiment (defined as Time 0). The class of functions is generated by a kernel that combines squared exponentials (reflecting biological variation) in the log-based time scale with additive white noise (to capture measurement error). The squared exponential time scale parameter is fixed at 2 to reflect the density of available observations. The squared exponential variance is fixed for each type of data by using an average of the optimal values over all compounds. The white noise variance is computed by Maximum Likelihood Estimation (MLE) for each specific compound.

The output of the Gaussian process workflow is a uniform time series for each compound with 100 time points for control and treated sample means and the corresponding standard deviations. In reality, the learned Gaussian process model is continuous. The 100 time points sample time dimension at sufficiently many discrete points to preserve the information. The *log-ratio time series* for each compound, which is the primary basis for subsequent stages of our workflow, is then computed by point-wise difference between treated and control. Figure 2 shows example plots.

In addition the confidence in the change (as measured by the log-ratio) is scored by computing the area between the standard deviation bands for control and treated samples. We perform computation for 1 and 2 standard deviations, the former to account for the higher degree of noise in the proteomics and metabolomics data. This provides a *ranking of the measured compounds* according to the area between the bands generated by the standard deviations in the visualization of Gaussian processes. It generalizes the biologist criteria that for a significant change the standard deviations of the control and treated samples should not overlap. This ranking can be used as a first filter to focus attention on compounds in which we have some confidence in the measured change.

2.2 PCA and the Resulting Ranking

Standard Principal Component Analysis (PCA) is carried out on the log-ratio timeseries generated by the Gaussian process model to determine the main sources of variance.[1] In the following we refer to the application of PCA along the time dimension, which yields principal components that are vectors over the 100 time points, assigning a weight to each timepoint corresponding to its contribution to the overall variance.[2]

Our results show that across all experiments, the top three components allow substantial *reduction in dimension* and hence in complexity of computations based on the PCA representation. As few as three components allow us to capture at least 90% of the variance (typically more than 95%). The output of this stage is a linear transformation of the time series data into a three dimensional vector space. As a by-product, the contribution of a compound to each PCA component provides another ranking of its response to a treatment.

2.3 Many Kinds of Clusterings

Clustering is a form of *abstraction*, which is essential for the biologist to cope with the overwhelming amount of observational data. However, since no single abstraction technique is superior in all applications, the clustering algorithm should be seen as an open-ended parameter in our approach. From the large number of clustering algorithms, we identified three general algorithm types that are sufficiently scalable to deal with the large number of observed compounds. Each type focuses on different features and they are all parameterizable to influence the degree of abstraction, giving the biologist a tunable magnifying glass.

[1] We use the machine learning framework [19] for PCA and basic clustering.

[2] Alternatively, PCA can be applied along the gene/compound dimension which we have done in another part of our RTA workflow.

Our current workflow includes well-known algorithms such as k-means clustering and two kinds of standard hierarchical clustering. It also includes a more recently proposed density-based hierarchical clustering algorithm (hdbscan) [16] that is more scalable than the standard ones, and supports partial clustering, meaning that not all compounds have to be assigned to the generated clusters.

Specifically, our k-means clustering workflow is based on the vector space arising from the PCA dimensionality-reduction (applied to the log-ratio time series data) equipped with the Euclidian metric. For tunability we allow the number of clusters k to range over $\{16, 32, 64, 128, 256, 512\}$. The other clustering algorithms are used with Pearson correlation ρ directly derived from the log-ratio time series data, using $1 - \rho$ as an affinity measure.[3] Standard hierarchical clustering is performed with two linkage methods (average distance and Ward's minimum variance method) that are very familiar for biologists working with gene expression data. A number of distance cutoffs are then used to generate clusterings of different granularity from the hierarchical clustering ($\{0.25, 0.5, 1, 2, 3, 4, 5, 10, 20, 30, 40, 50\}$ for average and $\{2, 3, 4, 5, 10, 20, 30, 40, 50, 100, 200, 300, 400, 500\}$ for Ward). Density-based clustering is performed with minimum cluster sizes $\{2, 3, 4, 5\}$ and an α-parameter ranging over $\{0.01, 0.1, 0.25, 0.5, 0.75, 0.9, 0.99\}$ to control the granularity.

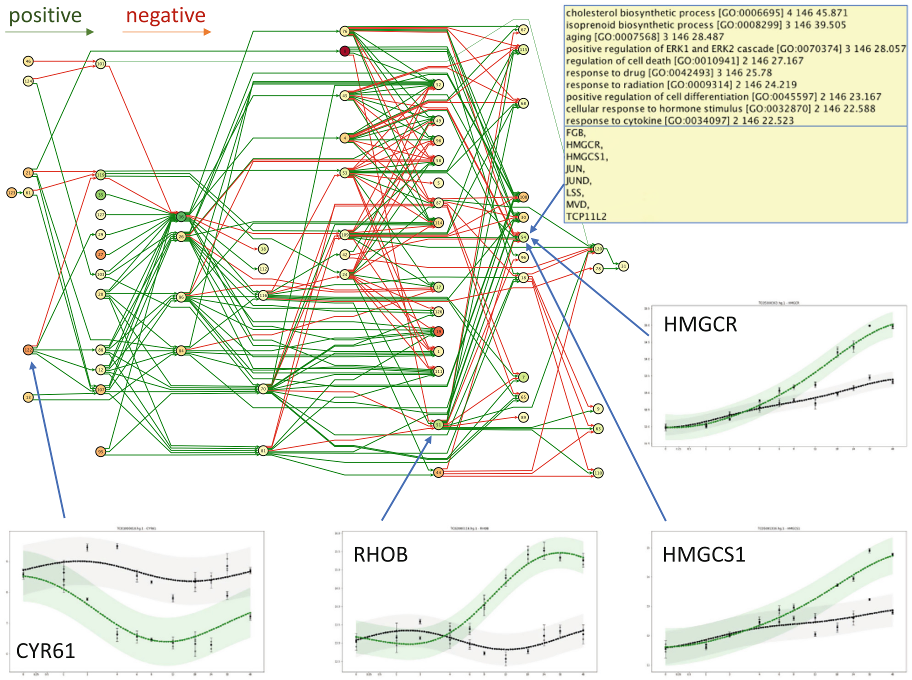
2.4 Clustered Cross-Correlation Graphs

Our workflow includes several *graph synthesis methods* that are intended to support the biologist in the identification of the MoA. For example, Pearson correlation ρ can be naturally visualized as a family of *undirected* graphs parameterized by a correlation threshold. These graphs have compounds (or suitable subsets) as nodes and edges denote positive or negative correlation $\|\rho\|$ beyond the threshold, and they often result in scale-free networks characterized by a power law node degree distribution [3].

Slightly more generally, we can define cross-correlation graphs by replacing the symmetric notion of correlation by cross-correlation, which is defined as correlation between a time series and its shifted version and associated with a time direction and a time lag. Hence, the resulting graphs are *directed*. In our current workflow, we consider discrete time lags in the range 1–20 based on the Gaussian process representation with 100 time points. We use green (red) edges to indicate the existence of a positive (negative) cross-correlation with a time lag in the range, and always perform some pruning of the graph to eliminate orphan nodes, i.e., isolated compounds. A heuristic is used to keep the size of the graph manageable and amenable to automatic layout algorithms and to facilitate interpretability by the biologist.

In spite of these efforts, the resulting graphs are often too low-level or detailed. Hence, we developed a more abstract notion of *clustered cross-correlation graphs* (Fig. 2). Here, cross-correlation is lifted from pairs of compounds to pairs of clusters, as they arise from one of the algorithms in our

[3] Another affinity measure we have explored is based on correlated changes (using time series derivatives) but beyond the scope of this paper.



cholesterol biosynthetic process [GO:0006695] 4 146 45.871
isoprenoid biosynthetic process [GO:0008299] 3 146 39.505
aging [GO:0007568] 3 146 28.487
positive regulation of ERK1 and ERK2 cascade [GO:0070374] 3 146 28.057
regulation of cell death [GO:0010941] 2 146 27.167
response to drug [GO:0042493] 3 146 25.78
response to radiation [GO:0009314] 2 146 24.219
positive regulation of cell differentiation [GO:0045597] 2 146 23.167
cellular response to hormone stimulus [GO:0032870] 2 146 22.588
response to cytokine [GO:0034097] 2 146 22.523

FGB,
HMGCR,
HMGCS1,
JUN,
JUND,
LSS,
MVD,
TCP11L2

Fig. 2. Sample clustered cross-correlation graph for Unk5 Genes (see Sect. 4.3) using PCA-based clustering. Each node represents to a cluster of genes modeled as Gaussian processes with the color indicating the average log2-ratio (fold change) over time and over the genes in the cluster (green = increased, red = suppressed). For one cluster we show the full list of genes and ranked GO-process annotations. The Gaussian models for sample transcripts are shown in the timeseries plots: green is treated, black is control (surrounded by 2 standard deviation bands), dots with whiskers show the original data. (Color figure online)

clustering workflow. The lifting is done through averaging, i.e., cluster cross-correlation for a fixed lag from a source to a target cluster is defined as the average cross-correlation for the same fixed lag over all pairs with the first component in the source and the second component in the target cluster. Using this notion, clustered cross-correlation graphs have clusters of compounds as nodes and defined in complete analogy to the basic cross-correlation graphs. Similarly, orphan clusters are removed from the graph so that only non-trivial connected components remain.

The advantage of using cross-correlation as a basis for graph synthesis is that it is a well-studied concept from signal processing that is visually intuitive for the biologist. The mathematical simplicity of the concept can however be a disadvantage, since it will miss potential causal interactions of higher complexity that do not manifest themselves as (homogeneous) time-lagged correlation. Hence, our workflow contains other complementary techniques for graph synthesis (beyond the scope of this paper) based on neural networks which try to learn and exploit characteristic features in the time profiles.

2.5 Differential Anomaly Identification Using Autoencoders

It is expected that the observed time series data contains many anomalies due to measurement errors or other forms of noise. However, given that Gaussian processes allow us to filter most of the noise and smoothen the data to some extent, any *remaining anomalies* suggest a potential role in the MoA, and can give valuable clues in conjunction with the previous results and graphs.

To detect such anomalies we use a variation of *convolutional autoencoders* [15]. An autoencoder [9] is a neural network that is a sequential composition of an encoder and a decoder that together are trained so that the output reconstructs the original input as closely as possible. The encoder transforms the input (time series with 100 data points in our case) into a low-dimensional feature space (we found that as few as 4 dimensions are sufficient to reconstruct most inputs). In our specific application it is a convolutional neural network (using a window of size 41) without weight-sharing (reflecting heterogeneity in time) but with L1-regularization (favoring sparsity), followed by max-pooling, and exponential-linear activation functions. The decoder is a dense neural network with L2-regularization and linear activation. In order to prevent certain compounds from dominating the data set, all input time series are normalized. We use mean square error as the loss function for training, which uses the Adam-optimizer with a fixed number of 40000 epochs and a batch size of 1000 (500 for proteomics). A random subset of 10% of the compounds is used for validation. Autoencoder models are synthesized separately for transcriptomics and proteomics data sets as well as for the combination of transcriptomics, proteomics, and metabolomics data. When building models involving proteomics data we found that adding a dropout layer between encoder and decoder (with a low dropout rate of 0.01 and 0.0001 for the combined data set) reduced the risk of overfitting.

One of our anomaly-detection workflows aims at detecting what we call *differential anomalies*, that is behavior in the treated condition that is hard to predict by a model that only learns the behavior under the control condition. To detect differential anomalies, we use an autoencoder model trained on the control time series data and apply it to the treated time series data for all compounds to quantify the reconstruction error (mean-square error between original and predicted time series). We then generate a ranked list of compounds sorted by this reconstruction error (highest error first).[4]

2.6 Heuristic Annotation of Clusters

Our workflow implements a *generic heuristic method* to compute likely classifications of a cluster from classifications of its elements. It is used to annotate

[4] We explored some other metrics based on the original (non-normalized) time series data that we omit for brevity.

each cluster with a ranked lists of classifications based on GO terms (process or function) or HGNC families. This is the *only use of prior knowledge* in our approach, and serves mostly to interpret proposed MoA candidates.

The generic method assumes a set of classification concepts and a known relationship between compounds and concepts, which does not have to be one-to-one (i.e., overlapping concepts or hierarchies are allowed, as they occur for example in the GO hierarchy). We denote by p_c the probability that the concept c is associated with a random compound (assuming uniform distribution) covered by the relationship. By $f_c(S)$ we denote the number of compounds in a cluster S that are associated with the concept c, and by $f(S)$ we denote the corresponding sum over all concepts. For each cluster S we use the score $s_c = -f_c(S)\log_2(p_c)$, which is the logarithm of $1 - p_c^{f_c(S)}$, to rank all concepts for the given cluster.

In our workflow, we use GO processes, GO functions, and HGNC families as sources of classification concepts. In each case, we use the top 10 concepts c to label each cluster S with $(c, f_c(S), f(S), s_c)$, or a few more if there is a tie in the score. The label makes it explicit that the classification c is based on $f_c(S)$ out of $f(S)$ concepts. Intuitively, our scoring function reduces the impact of frequently occurring concepts (low information content) but amplifies the impact of co-occurring concepts (a form of confirmation) in the same cluster. It should be noted that the scoring function does not rely on expression levels of compounds and is not intended for comparisons between clusters.

3 The Omics Data

The data to be analysed was generated as follows. HepG2/C3A hepatocyte were exposed to five different challenge agents: forskolin, nocodazole, bendamustine, nexturastat A, and atorvastatin. Molecular responses were measured from three biological replicates at eight to ten time-points between 10 s and 48 h post-exposure by microarray-based transcriptomics, multiplexed high-resolution LC-MS shotgun proteomics, and novel untargeted metabolomics to quantify >67,000 transcripts, >5,000 proteins and >200 metabolites. Immuno-assays and qRT-PCR were performed to validate the proteomic and transcriptomic data, respectively. Biological assays were performed to down select and confirm the MoA candidates inferred from the various data analyses.

The challenge agents spanned a wide range of mechanisms: forskolin is a diterpene that activates the enzyme adenylyl cyclase and the cAMP signaling pathway; nocodazole is a microtubule-targeting agent that disrupts microtubule polymerization and induces cell cycle arrest in the prometaphase; bendamustine is an alkylating agent with a nitrogen mustard group which causes the formation of crosslinks between DNA bases; nexturastat A is an inhibitor of HMG-CoA reductase which is a key step in cholesterol biosynthesis.

4 Choosing MoA Candidates

In this section we illustrate the use of the Gaussian process based analyses of time series omics data to identify candidate participants in the MoA in response to exposure to a challenge agent. We begin with an overview of the process we developed and then illustrate the ideas using data from two of the challenge compounds studied.

4.1 Overview of Selection Process

In addition to the Gaussian process (GP) based analysis we carry out some standard time series analysis to provide a cross check. For each entity, the log2 fold change was computed at each of the measured time points as the difference of the log2 of the means of the control and treated replicates. We refer to these change vs time plots as *basic* time profiles. Time points for which the set of control measurements are either all above or all below the set of treated measurements are annotated as significant.[5]

The Shape of Response. Before looking for specific candidate MoA participants we look at the shape of the overall response in two ways: the level of up/down regulation at each timepoint; and the population level cell cycle state over time. For the levels of up/down regulation we make *udby* (Up/Down By) charts. These are bar charts plotting the relative number of entities that are first up/down regulated by a given threshold amount at each measured time point. This gives a visual representation of the overall shape of the response (when are events happening). The given threshold magnitude is typically between .5 and 1 log2 fold change, adjusted to ensure at least 3–5% of the measured data is included. Maps are also computed associating to each entity the maximum and minimun log2 fold change along with the times the extrema are reached. Figure 3 on the left shows the *udby* chart for Unk3 transcripts (based on the basic time profile). From the chart we observe activity at 4–6 h and 18–24 h, but there is a lull at 8 h.

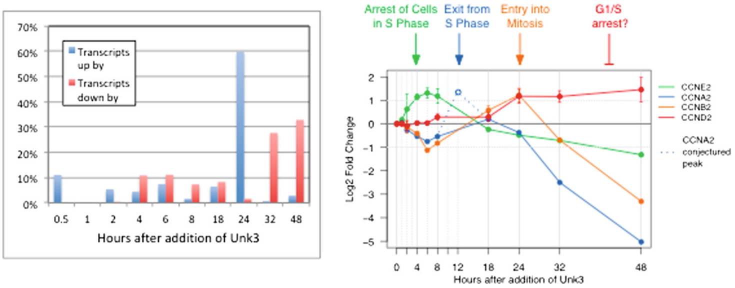


Fig. 3. Up/Dn By Chart and Cyclin gene plots for Unk3

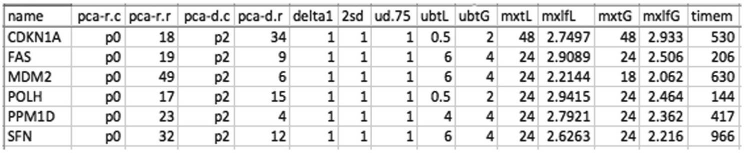
Information about population cell cycle state is visualized by plotting the time series profiles of four key Cyclins. The shape of these plots contains information about whether or not the perturbing agent disrupts the cell cycle and if so at

[5] Currently we restrict analysis of transcriptomics data to protein coding genes.

which points. Figure 3 on the right shows the basic time profiles of Cyclin mRNA expression for Unk3 transcripts. From the plot we see a pattern of Cyclin mRNA expression that reflects a transient S phase arrest followed by a synchronized transition through the cell cycle and an apparent arrest at G1/S. The reasoning and the reason it is interesting to look at the cyclins is as follows. Cyclins are proteins whose protein and mRNA concentrations vary at different stages in the cell cycle. As long as the cells are growing exponentially, the proportion of cells in each cell cycle stage stays constant and the expression of each cyclin does not change over time. An increase in the expression of cyclin E2 (CCNE2) indicates that the proportion of cells in S-phase is increasing. If the arrest were permanent, the level of CCNE2 would stay high for the rest of the time course. The decrease in CCNE2 expression from 6–18 h indicates that the arrest is transient.

Cross Correlation and Visualization. To provide a visual summary of the potential relations among responding elements, we use the cross-correlation graphs on clusters (described in Sect. 2.4). Graph nodes are annotated by results from analyses including: classification, cluster members, average change, GO terms and family classifications (see Fig. 2). One can visually identify potentially interesting nodes by criteria such as being input, output, or highly connected, i.e., with high betweenness centrality, and use the annotations to drill down.

Filtering and Ranking. We consider several criteria for selecting or ranking transcripts, proteins and metabolites (aka entities). The *1sd* and *2sd* separation properties identify entities with statistically significant change by measuring the space between one or two standard deviation bands around the control and treated GP time profiles as explained in Sect. 2.1. The *udby.th* property holds for entities that have a significant log2 fold change of at least *th* at some measured time point, where *th* is typically between .5 and 1. This is computed using the basic time profile, using the significance annotation at each time point. The *delta.th* property holds of an entity if the difference between the maximum and minimum log2 fold changes (in the GP time profile) is greater than *th*. This is relevant because the GP profile may be shifted up or down, or over smoothed (likely due to sparse or noisy data) so that the GP time profile is essentially flat. Figure 4 shows properties for transcripts from the Unk3 data. All of these transcripts satisfy the *2sd*, *udby.75*, *delta.1* properties. The table also includes the *upby* .75 time, and the time and magnitude of the maximum fold change (*mxt*,*mxlf*) computed from both the basic and GP profiles (*L*,*G*).



name	pca-r.c	pca-r.r	pca-d.c	pca-d.r	delta1	2sd	ud.75	ubtL	ubtG	mxtL	mxlfL	mxtG	mxlfG	timem
CDKN1A	p0	18	p2	34	1	1	1	0.5	2	48	2.7497	48	2.933	530
FAS	p0	19	p2	9	1	1	1	6	4	24	2.9089	24	2.506	206
MDM2	p0	49	p2	6	1	1	1	6	4	24	2.2144	18	2.062	630
POLH	p0	17	p2	15	1	1	1	0.5	2	24	2.9415	24	2.464	144
PPM1D	p0	23	p2	4	1	1	1	4	4	24	2.7921	24	2.362	417
SFN	p0	32	p2	12	1	1	1	6	4	24	2.6263	24	2.216	966

Fig. 4. Examples of properties for Unk3 transcripts.

The *pca.r* and *pca.d* ranking functions assign rank n (smaller is better) to an entity if it appears as the n^{th} element in the ranked list of one of the PCA components (positive or negative), where the PCA analysis is based on the GP ratio time profile or its derivative, respectively. From Fig. 4 we see that the example transcripts all have rank below 20 for at least one PCA analysis.

The *timem*, *time*, *diffm*, and *diff* ranking functions assign rank n to an entity if it is the n^{th} element in the ranked list according to the corresponding anomaly calculation as described in Sect. 2.5. *timem* and *time* anomaly rankings are based on the GP ratio time profile where for *timem* the mean square error is computed after transforming the model back to the original data scale, while for *time* it is computed on the normalized model. Thus *timem* is biased towards larger changes. *diffm* and *diff* anomaly rankings are based on predicting the treated time series from the control time series. In the case of the example transcripts of Fig. 4, none have an anomaly rank below 100. Only *timem* is shown. Given these properties and ranking functions we combine them by conjunction and disjunction to select subsets of entities for further examination.

Feature Summary Spreadsheets. As an aid to exploring the candidate subsets we summarize the above properties/ranks and other information in a spreadsheet with rows corresponding to entities in some subset of interest. The columns record satisfaction of properties, ranks, and results from other computations. (Figure 4 is a selection from one such spread sheet.) The entities can then be grouped by sorting on different combinations of columns to hopefully reveal key players. The columns include the ten properties and rankings discussed above. There are also columns labeled *cl.r* and *cl.d* that record the number of the cluster containing the entity, using the PCA K-means clustering (with 128 clusters for transcriptomics and proteomics). *cl.r* similarity is based on the GP ratio time profile and *cl.d* similarity is based on the derivative of the GP ratio time profile. We also include columns *maxlf*, *minlf*, *maxt*, and *mint* recording the log2 fold change maximum, minimum, and corresponding times.

Picking Winners. With all the analyses, properties, and ranking functions in hand, the challenge is to identify the most significant responders to the treatment. The main criteria of success is whether the data somehow reveals the 'canonical' MoA. But there are generally many possibly important things going on that are not included in the canonical pathway. Perhaps the data will reveal something new.

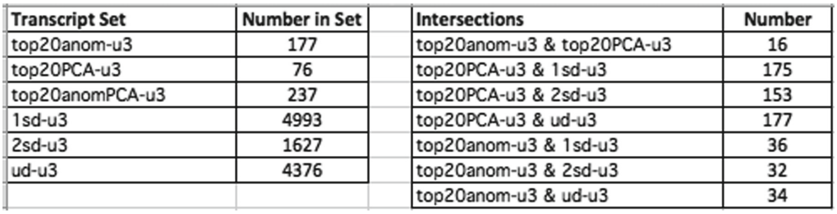
We start by investigating two subsets. One subset is the entities that satisfy a separation property (*1sd* or *2sd*) or an udby property (*udby.th*). The choice of property within the two classes is determined by initially wanting to limit the number of candidates to ~1000 or less. The other subset contains the entities that rank in the top n of one of the six ranking functions. Generally, taking n to be 20 gives us a reasonable set to consider.

We make feature summary spreadsheets for each of the two subsets and look for groups that stand out according to some collection of features. When sorting by one of the cluster columns we can get additional information by looking at the annotations (GO term or family classifications) associated with clusters that have a relatively large number of elements from the subset under consideration. Of particular interest are the most specific annotations. GO terms such as signal transduction, cell cycle, or metabolic process don't tell you much although they can support conjectures derived from clusters annotated with more specific terms.

4.2 Challenge Compound Unk3

For Unknown 3 (henceforth Unk3) the transcriptomics data supported basic time profiles for 18208 transcripts and GP-based time profiles for 18834 transcripts. There were 4396/2803 transcripts that had a log2 fold change magnitude of at least 0.75 at some time point according to the basic/GP timeprofile. Of the 2803 transcripts identified by GP-based analysis, 1579 satisfied the *delta.75* property (defined in Sect. 4.1). The difference between basic and GP-based results has to do with the Gaussian process smoothing and integrating all the data, while the basic analysis computes changes point-wise. There were 4993 transcripts that passed the one standard deviation separation filter (satisfied *1sd*), while 1627 transcripts passed the two standard deviation separation filter (satisfied *2sd*).

The *udby* chart for Unk3 transcripts and the time profiles of Cyclin mRNA expression have already been presented in Sect. 4.1. Now we look at the top ranked transcripts according to PCA ranking and anomaly ranking. Thus we defined `top20PCA-u3` to be the set of transcripts ranked in the top 20 of one of the three PCA components (positive or negative), using PCA based on the GP time profile or its derivative. We defined `top20anom-u3` to be the set of transcripts ranked in the top 20 of one of four anomaly measures (timem, time, diffm, diff). We let `top20anomPCA-u3` denote the union of `top20PCA-u3` and `top20anom-u3`. To get a first impression of how the filters based on statistical significance relate to filters based on features we compared the PCA and anomaly ranked sets to those passing the one and two standard deviation separation filters (*1sd*, *2sd* respectively) and to the filter selecting transcripts with log2 fold change magnitude at least 0.75 at some time point (*ud*). The numbers in each set and their intersections are shown in Fig. 5.



Transcript Set	Number in Set	Intersections	Number
top20anom-u3	177	top20anom-u3 & top20PCA-u3	16
top20PCA-u3	76	top20PCA-u3 & 1sd-u3	175
top20anomPCA-u3	237	top20PCA-u3 & 2sd-u3	153
1sd-u3	4993	top20PCA-u3 & ud-u3	177
2sd-u3	1627	top20anom-u3 & 1sd-u3	36
ud-u3	4376	top20anom-u3 & 2sd-u3	32
		top20anom-u3 & ud-u3	34

Fig. 5. Overview of rank-based selections for Unk3.

We see that the ranking according to PCA vs anomaly is quite different, only 16 transcripts pass both filters. Also, the statistical filters overlap substantially with the PCA based filters, but much less with the anomaly based filters.

For the next step we chose to focus on the `top20anomPCA-u3` and produced a sortable feature table with one row for each transcript in `top20anomPCA-u3`. Sorting by feature (PCA or anomaly) doesn't give any obvious insights. So, we sort by cluster (ratio-based or derivative based). We start by looking at clusters that contain at least six elements from `top20anomPCA-u3` and search the GO annotations associated with these clusters for the most specific annotations. The second annotation of Cluster 16 is "DNA damage response, signal transduction by p53 class mediator resulting in cell cycle arrest [GO:0006977]". This cluster contains CDKN1A, FAS, MDM2, PPM1D, SFN, POLH, (P53 responsive and DNA repair genes). With this clue we look further and find two more clusters with this annotation and several additional clusters with DNA damage/DNA repair annotations. The transcripts in these clusters are up-regulated early (by 2–6 h) and reach a maximum around 24 h. Consistent with the observation based on the plot of the time profiles of the Cyclins (Fig. 3), four of the larger clusters are annotated with GO terms related to G1-S transition. Two of these clusters are down regulated by 18 h and reach minimum at 48 h, while two have periods of up and down regulation. The highly ranked POLH, upregulated between 4 and 24 h, is a DNA polymerase involved in DNA repair.

We conclude that the treatment likely resulted in DNA damage leading to a P53 response in addition to the temporary cell cycle arrest. After the measurements and data analysis, Unk3 was revealed to be bendamustine, known to crosslink DNA strands and interfere with DNA damage repair mechanisms. This confirmed the results of the analyses.

4.3 Challenge Compound Unk5

For Unknown 5 (henceforth Unk5) the transcriptomics data supported basic time profiles for 15685 transcripts and GP-based time profiles for 17347 transcripts. There were 570/407 transcripts that had a log2 fold change magnitude of at least .75 at some time point according to the basic/GP time profile.

Of the 407 transcripts identified by GP-based analysis, 272 satisfy *delta.75*. There were 4442 transcripts that passed the one standard deviation separation filter, while 1287 transcripts passed the two standard deviation separation filter.

Figure 6 shows the *udby* chart for Unk5 transcripts and the time profiles of Cyclin mRNA expression. From the chart we see that the bulk of the transcriptomics activity happens at 18 h or later. From the Cyclin plot we see a pattern of Cyclin mRNA expression that reflects a permanent arrest in G1, namely the steady increase in Cyclin D2 (CCND2) after 24 h.

As for Unk3, we next look at the top ranked transcripts according to PCA ranking and anomaly ranking. Thus we defined `top20PCA-u5` to be the set of transcripts t ranked in the top 20 according to PCA ranking, that is $pca.r(t) < 20$ or $pca.d(t) < 20$. We defined `top20anom-u5` to be the set of transcripts t ranked in the top 20 according to one of the four anomaly measures, i.e., $timem(t) < 20$

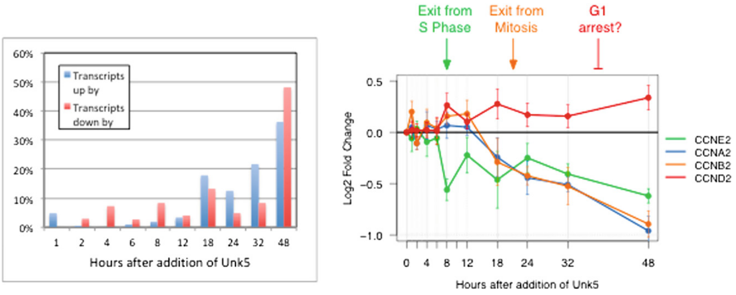


Fig. 6. Up/Dn By Chart and Cyclin plots for Unk5

or $\mathit{time}(t) < 20$ or $\mathit{diffm}(t) < 20$ or $\mathit{diff}(t) < 20$. We let `top20anomPCA-u5` denote the union of `top20PCA-u5` and `top20anom-u5`. As for Unk3, the ranking according to PCA vs. anomaly is quite different, only 24 transcripts pass both filters. We note that `top20anom-u5` and `top20PCA-u5` contain eight metallothioneins: MT1A, MT1B, MT1E, MT1F, MT1G, MT1H, MT1M, and MT1X. These transcripts appear in four different clusters, all are upregulated by 1 h and reach a maximum between 2 h and 8 h.

For the next step we chose to focus on the `top20anomPCA-u5` and produced a sortable feature table with one row for each transcript in `top20anomPCA-u5` and columns as described in Sect. 4.1. Sorting by $cl.r$ (ratio based clustering) we find three relatively large clusters annotated with Cholesterol Biosynthetic process GO terms. Two more clusters are annotated with metabolic process GO terms. Cluster 54 contains HMGCS1, HMGCR, MVD, LSS, FGB, JUN, JUND, and TCP11L2, the first four are enzymes in the canonical Cholesterol biosynthesis pathway (see Figs. 2, 7). All of the entities in the cluster are up-regulated midway and reach a maximum late. The Cholesterol enzymes pass the .75 threshold at 18 h and reach a maximum at 48h. Five clusters are annotated with metal ion and zinc binding GO terms. Sorting by $cl.d$ (derivative based clustering) we find two clusters annotated with Cholesterol Biosynthetic process GO terms and one with Cholesterol import. There are three clusters annotated with metal ion and zinc binding GO terms.

Although the proteomics data for this pathway is too sparse to be highly ranked by the GP analysis, once we have the clue, we plot the raw data for transcripts and proteins in the pathway and see a general trend for slow steady up-regulation and a remarkable agreement between transcripts and the proteins they code for. The metabolomic data plots for Cholesterol and Chenodeoxycholic indicate possible down regulation sometime after 8 h, adding evidence to a conjecture that the unknown is inhibiting something in that pathway and the cells are responding by trying to make more. This is illustrated in Fig. 7.

After the omics experiments and data analysis, Unk5 was revealed to be atorvastatin, an inhibitor of HMG-CoA reductase (HMGCR), as proposed by the analysis. The strong early response by the metallothioneins remains a mystery to be further investigated.

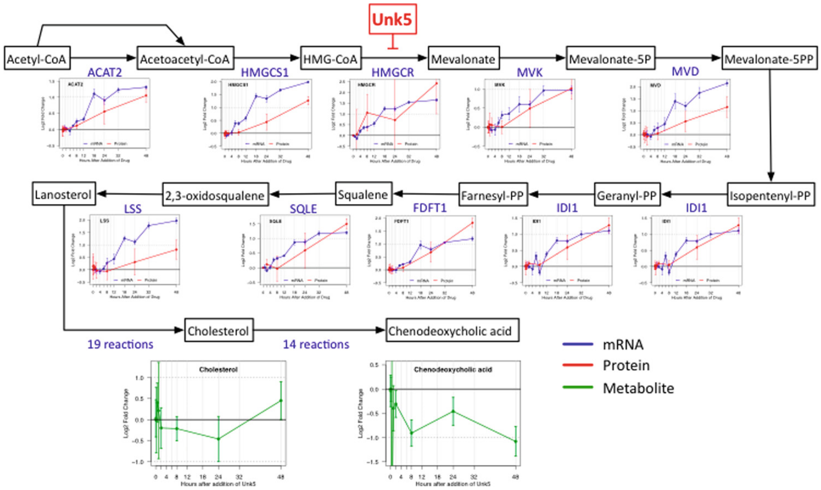


Fig. 7. Canonical Cholesterol pathway annotated with Unk5 data

5 Conclusions and Future Directions

We have presented a collection of algorithms and reproducible workflow for analysis of comparative multi-omics time series data. The goal is to be able to identify candidate participants in the MoA of chemical or biological agents using data analysis, without recourse to prior knowledge such as interaction and pathway databases. The algorithms are based on Gaussian process models that leverage a large number of data points to derive useful information from data that may be noisy or have missing data points. The Gaussian process models support integration of timeseries based on different sets of time points and different types of data. The workflows produce several functions for ranking candidates according to different features, functions for grouping entities according to different similarity criteria, and graph generation algorithms for visual representation of correlations and potential causality relations among entities or groups of entities. Each function and relation brings different insights into the cellular response to a perturbation. We illustrated the application of this computational analysis suite to identify key elements of the MoA of two unknowns, eventually revealed to be bendamustine and atorvastatin.

As mentioned in Sect. 1, existing approaches to determining the MoA of a cellular perturbation rely on existing knowledge, and may work with only single timepoints or a small number of timepoints (see [18] for an extensive overview). Two recent works [18,29] based on network analysis are perhaps the closest to our approach. DeMAND [29] scores candidate MoA proteins based on an assessment of the global dysregulation of their molecular interactions following perturbation, based on an input gene-protein interaction network. Multiple time points can be used, but the temporal information is not used. ProTina [18] creates a cell type specific protein-gene regulatory network (from existing knowledge bases) based on a dynamic model of the gene transcription. Protein targets are scored based on the enhancement/attenuation of protein-gene regulations. The dynamic model allows inclusion of temporal information in the scoring process. Note that in contrast, our approach does not require a regulation network as input.

Future Directions. We note that in contrast to usual work where a single approach is developed and validated against synthetic and experimental data, our workflows result in multiple approaches modeling diverse features that might be observed as part of an MoA, giving a flexible tool set for analyzing data from a truly unknown perturbing agent. One important future direction is validating the approach on a wide class of cellular perturbations and to generate a validated characterization of each of the approaches. The current tool suite is used by experts in computational modeling. Increasing the level of automation for identifying MoA candidates by combining the different approaches and integrating the visualization capabilities would make the capabilities accessible to more researchers and hopefully help to advance research in MoAs at the cellular response level.[6]

References

1. Abadi, M., et. al.: TensorFlow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, pp. 265–283. USENIX Association (2016)
2. Barabasi, A.L., Gulbahce, N., Loscalzo, J.: Network medicine: a network-based approach to human disease. Nat. Rev. Genet. **12**(1), 56–68 (2011)
3. Barabasi, A.L., Oltvai, Z.N.: Network biology: understanding the cell's functional organization. Nat. Rev. Genet. **5**(2), 101–113 (2004)
4. Cajka, T., Fiehn, O.: Toward merging untargeted and targeted methods in mass spectrometry-based metabolomics and lipidomics. Anal. Chem. **88**(1), 524–545 (2016)
5. Dettmer, K., Aronov, P.A., Hammock, B.D.: Mass spectrometry-based metabolomics. Mass Spectrom. Rev. **26**(1), 51–78 (2007)
6. de Matthews, G., et al.: GPflow: a gaussian process library using tensorflow. J. Mach. Learn. Res. **18**, 40:1–40:6 (2017)
7. Girault, C., Valk, R.: Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-662-05324-9
8. Goodwin, S., McPherson, J.D., McCombie, W.R.: Coming of age: ten years of next-generation sequencing technologies. Nat. Rev. Genet. **17**(6), 333–351 (2016)

[6] Disclaimer. Research was sponsored by the U.S. Army Research Office and the Defense Advanced Research Projects Agency and was accomplished under Cooperative Agreement Number W911NF-14-2-0020. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office, DARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

9. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
10. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., Barabasi, A.L.: The large-scale organization of metabolic networks. Nature **407**(6804), 651–654 (2000)
11. Kim, M.S., et al.: A draft map of the human proteome. Nature **509**(7502), 575–581 (2014)
12. Kluyver, T., et. al.: Jupyter notebooks - a publishing format for reproducible computational workflows. In: Loizides, F., Schmidt, B. (eds.) Positioning and Power in Academic Publishing: Players, Agents and Agendas, pp. 87–90. IOS Press (2016)
13. Kramer, A., Green, J., Pollard, J., Tugendreich, S.: Causal analysis approaches in ingenuity pathway analysis. Bioinformatics **30**(4), 523–530 (2014)
14. Mann, M., Kulak, N.A., Nagaraj, N., Cox, J.: The coming age of complete, accurate, and ubiquitous proteomes. Mol. Cell **49**(4), 583–590 (2013)
15. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional autoencoders for hierarchical feature extraction. In: Honkela, T., Duch, W., Girolami, M., Kaski, S. (eds.) ICANN 2011 Part I. LNCS, vol. 6791, pp. 52–59. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21735-7_7
16. McInnes, L., Healy, J., Astels, S.: HDBSCAN: hierarchical density based clustering. J. Open Sour. Softw. **2**(11) (2017)
17. Mi, H., et al.: Panther version 11: expanded annotation data from gene ontology and reactome pathways, and data analysis tool enhancements. Nucleic Acids Res. **D45**(1), D183–D189 (2017)
18. Noh, H., Shoemaker, J.E., Gunawan, R.: Network perturbation analysis of gene transcriptional profiles reveals protein targets and mechanism of action of drugs and influenza a viral infection. Nucleic Acids Res. **46**(6), e34 (2018)
19. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
20. Pujol, A., Mosca, R., Farres, J., Aloy, P.: Unveiling the role of network and systems biology in drug discovery. Trends Pharmacol. Sci. **31**(3), 115–123 (2010)
21. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (2005)
22. Tautenhahn, R., et al.: An accelerated workflow for untargeted metabolomics using the METLIN database. Nat. Biotechnol. **30**(9), 826–828 (2012)
23. Uhlen, M., et al.: Tissue-based map of the human proteome. Science **347**(6220), 4 (2015)
24. Vertes, A., et. al.: Time-dependent metabolomics in systems biology context for mechanism of action studies. In: US HUPO Conference - Proteomics: From Genes to Function, San Diego, CA (2017)
25. Vertes, A., et. al.: Mechanism of action identification of threat agents within 30 days. In: Society of Toxicology 57th Annual Meeting, San Antonio, TX (2018)
26. Vertes, A., et. al.: Novel high-throughput metabolomic techniques and mainstream tools for the discovery of drug mechanism of action. In: US HUPO 14th Annual Conference - Technology Accelerating Discovery, Minneapolis, MN (2018)
27. Vertes, A., et. al.: Systems biology approach for mechanism of action identification in 30 days. In: ASMS Conference, San Diego, CA (2018)
28. Wang, Z., Gerstein, M., Snyder, M.: RNA-Seq: a revolutionary tool for transcriptomics. Nat. Rev. Genet. **10**(1), 57–63 (2009)
29. Woo, J.H., et al.: Elucidating compound mechanism of action by network Perturbation analysis. Cell **162**(2), 441–451 (2015)
30. Xu, W.H., et al.: Human transcriptome array for high-throughput clinical studies. Proc. Natl. Acad. Sci. U.S.A. **108**(9), 3707–3712 (2011)