OUKA
The University of Osaka
Institutional Knowledge Archive

Title	Stratification of triple-negative breast cancers using patient-specific modeling
Author(s)	井元, 宏明
Citation	大阪大学, 2023, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/92181
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Stratification of triple-negative breast cancers using patient-specific modeling

患者固有モデリングによるトリプルネガティブ乳がんの層別化

Hiroaki Imoto

井元　宏明

Department of Biological Sciences

Graduate School of Science

Osaka University

大阪大学大学院理学研究科生物科学専攻

A Thesis Submitted for the Degree of

Doctor of Philosophy

博士学位論文

January 2023

令和 5 年 1 月

Abstract

Cancer treatment and drug development are impeded by patient heterogeneity at the levels of mutations and gene expression. Hence, methods for identifying prognostic markers for individual treatment are urgently required. As a growing number of studies indicate, cells may transmit information by altering the temporal dynamics of their signaling molecules, I hypothesized that the signaling dynamics of patients could be utilized as biomarkers. However, most available data obtained from patients represents static snapshots taken at a single point in time, not time-resolved dynamics. To overcome this problem, I developed Pasmopy (Patient-Specific Modeling in Python), a computational framework for the stratification of patients based on intracellular signaling dynamics generated from personalized kinetic models. Using this framework, I built a model of the ErbB receptor signaling network, trained parameters against time-series data obtained from four breast cancer cell lines, and performed *in silico* simulations of 377 breast cancer patients using The Cancer Genome Atlas (TCGA) transcriptome datasets. The temporal activation dynamics of Akt, extracellular signal-regulated kinase (ERK), and c-Myc in each patient were able to accurately predict the differences in prognosis and sensitivity to kinase inhibitors in triple-negative breast cancer. The framework presented in this study applicable to any type of signaling network and allows for the identification of key molecular mechanisms in individual patients, which would be difficult if only correlations between individual gene signatures and clinical outcomes were examined.

Acknowledgments

I would like to thank and show my gratitude to my supervisor, Dr. Mariko Okada, for providing me with unsurpassed academic support. I am very thankful for the opportunity you have given me.

Special thanks to Ms. Sawa Yamashiro who contributed equally to the patient-specific modeling study. I could not have finished this work without your help.

I am also grateful to the current and former members of the Laboratory for Cell Systems, particularly Dr. Ken Murakami, for updating the code to be compatible with the newest GDC Data Portal; Dr. Suxiang Zhang, for performing the western blot experiments with breast cancer cell lines; Mr. Fabian Ormersbach, for his contribution to the BioMASS modeling framework; Mr. Hiroki Michida for his tremendous effort in revising the NF-κB work; and Mr. Kyoichi Ebata, for our fruitful discussions.

I would like to express my sincere gratitude to Dr. Boris Kholodenko and Dr. Oleksii Rukhlenko, for the thought-provoking discussions during my stay at Systems Biology Ireland at University College Dublin (UCD). You have taught me so much. I am so happy and proud to have done research with you.

I would also like to thank Dr. Cormac Taylor, Dr. Naonori Kodate, and Mr. Darragh Ryan-Cahill for supporting me in more ways than I can express during my stay at UCD.

I would like to thank Dr. Masahiro Ueda and Dr. Koichi Fujimoto for critically reading this thesis and for their valuable discussions.

Many thanks to my family, in particular my mum, my dad, my brother, and my grandparents, for their love and constant support. You have encouraged and pushed me.

Thank you to all my friends for all the time we have spent together.

Last but by no means least, I am grateful to the Japan Society for the Promotion of Science (Grant-in-Aid for JSPS Fellows Grant No. 20J20192; Overseas Challenge Program for Young Researchers) for the wonderful support throughout my doctoral study.

Table of Contents

Abstract ..... **i**

Acknowledgments ..... **ii**

Originality ..... **vii**

1 Introduction **1**

1.1 Research topic 1

1.2 Computational systems biology approaches in cancer research 3

1.3 Contribution of this thesis 4

2 Materials and methods **7**

2.1 Model development 9

2.2 Training datasets 10

2.3 Parameter estimation 10

2.4 Individualization of the mechanistic model 11

2.5 Data processing in transcriptomic data integration 12

2.6 Clustering breast cancer patients with gene expression level 15

2.7 Gene mutation analysis 16

2.8 Extraction of response characteristics 16

2.9 Sensitivity analysis 17

2.10 Drug response data analysis 18

3 Results 19

3.1 Development of BioMASS: a computational framework for modeling and analysis of signaling systems 19

3.2 A comprehensive model of c-Fos regulation triggered by the ErbB receptor activation 21

3.3 Inferring intracellular signaling dynamics from individual gene expression profiles 24

3.4 Development of Pasmopy: a scalable computational toolkit for patient-specific modeling 28

3.5 Using transcriptomic data to personalize the mechanistic model 34

3.6 Stratification of TNBC patients based on signaling dynamics 38

3.7 Identification of the mechanisms affecting patient prognosis 45

3.8 Applying model-based stratification to other types of cancer 52

4 Discussion 55

4.1 Summary and Conclusion 55

4.2 Future perspectives 60

5 Appendix 64

5.1 Modeling biological systems using *BioMASS* 64

5.1.1 Code availability 64

5.2 Model preparation 65

5.2.1 Text-to-model conversion 65

5.2.2 Set the input of the model 73

5.2.3 Normalize simulation results 75

5.2.4 Choose an ODE solver to use 75

5.2.5 Set experimental data for parameterization of the model 76

5.2.6 Set lower/upper bounds of parameters to be estimated 79

5.2.7 Create a new model 82

5.3 Parameter estimation 82

5.3.1 Using `optimize()` function 82

5.3.2 Data export and visualization 84

5.3.3 Visualization of simulation results 85

5.4 Sensitivity analysis 86

Bibliography 89

Originality

Papers

This thesis contains material from the following papers (organized chronologically):

1) **Imoto, H.**, Zhang, S. & Okada, M. A Computational Framework for Prediction and Analysis of Cancer Signaling Dynamics from RNA Sequencing Data—Application to the ErbB Receptor Signaling Pathway. *Cancers* **12**, 2878 (2020). https://doi.org/10.3390/cancers12102878
2) **Imoto, H.**, Yamashiro, S. & Okada, M. A text-based computational framework for patient -specific modeling for classification of cancers. *iScience* **25**, 103944 (2022). https://doi.org/10.1016/j.isci.2022.103944
3) **Imoto, H.**, Yamashiro, S., Murakami, K. & Okada, M. Protocol for stratification of triple-negative breast cancer patients using *in silico* signaling dynamics. *STAR Protocols* **3**, 101619 (2022). https://doi.org/10.1016/j.xpro.2022.101619

Open-source software

A substantial component of my Ph.D. has been the democratization of patient-specific modeling via open-source software development. In particular:

BioMASS: A Python Framework for Modeling and Analysis of Signaling Systems

https://github.com/biomass-dev/biomass

Pasmopy: Patient-Specific Modeling in Python

https://github.com/pasmopy/pasmopy

Other

My Ph.D. work has included several other papers, but as they cover other topics – bifurcation analysis of a model of signal-dependent cell cycle entry, and mathematical modeling of transcriptional switches – they do not form a part of this thesis.

- **Imoto, H.** & Okada, M. Signal-dependent regulation of early-response genes and cell cycle: a quantitative view. *Current Opinion in Systems Biology* **15**, 100-108 (2019). https://doi.org/10.1016/j.coisb.2019.04.003
- Michida, H., **Imoto, H.**, Shinohara, H., Yumoto, N., Seki, M., Umeda, M., Hayashi, T., Nikaido, I., Kasukawa, T., Suzuki, Y. & Okada-Hatakeyama, M. The Number of Transcription Factors at an Enhancer Determines Switch-like Gene Expression. *Cell Reports* **31**, 107724 (2020). https://doi.org/10.1016/j.celrep.2020.107724

Once again, these are not included in this thesis.

1 Introduction

1.1 Research topic

Triple-negative breast cancer

Breast cancer is the most commonly diagnosed cancer in women. The estrogen receptor (ER), progesterone receptor (PR), and human epidermal growth factor receptor 2 (HER2) expression levels are used in the conventional molecular categorization of breast cancer (HER2). These receptors are both oncogenic drivers and relevant drug targets. Triple-negative breast cancers (TNBCs) lack expression of ER, PR and HER2 and are considered the most aggressive breast cancer subtype. Although TNBC patients respond to chemotherapy initially, they have a poorer prognosis than the other subtypes[1]. As a result, further stratification of TNBCs is required to choose the best therapeutic strategy for each patient.

Because the majority of basal-like cancers are also triple-negative breast cancers[2], it has been claimed that the triple-negative and basal-like phenotypes are effectively synonymous. Therefore, for this thesis, I focused on the basal-like subtype available from the Cancer Genome Atlas Database (TCGA)[3] to better stratify TNBCs.

ErbB signaling network

The ErbB receptor signaling pathway is a prototypical signaling pathway involved in a variety of physiological processes, and its dysregulation is strongly linked to the

development and progression of many types of human cancer[4]. Systems-level analyses of ErbB receptor signaling pathways have revealed ligand-specific cellular responses arising from this network and pathway regulatory mechanisms[5–11]. Nonetheless, it is unclear why the same stimulus (such as a growth factor) elicits such disparate responses in various cell types.

The ErbB receptor tyrosine kinase family is composed of four members, ErbB1 (also called the epidermal growth factor (EGF) receptor), ErbB2, ErbB3, and ErbB4. Several ligands known to bind to these receptors. When a ligand binds to the receptor, the receptors form dimers, activating their intrinsic tyrosine kinases, and thereby stimulating multiple signaling pathways[12]. Cells transmit information from the extracellular environment to the nucleus via signaling molecules and temporal activation patterns in this sequence of events[13]. EGF and another member of the EGF family, heregulin (HRG), for example, induce transient and sustained extracellular signal-regulated kinase (ERK) activity in cell lines such as MCF-7 that express low and high levels of ErbB1 and ErbB3 receptors, respectively[14,15]. Finally, changes in ERK activation dynamics affect the expression patterns of immediate-early genes such as c-Fos and c-Myc transcription factors[16], which are important regulators of the cell cycle[17,18], differentiation[14,19], and metabolism[20]. Therefore, a system-level understanding of the common mechanisms that regulate ErbB dynamics across cell types is important for uncovering a fundamental mechanism in individual cancer progression.

1.2 Computational systems biology approaches in cancer research

Cancer is a heterogeneous disease in terms of mutation signatures, gene expression profiles, and responses to drug treatments[21]. Innovations in sequencing, genome-wide mutation measurements, and transcriptomics profiles have increased awareness of inter-patient heterogeneity[22,23]. As a result, various data-driven algorithms, such as machine learning methods[24,25], have been developed to identify correlations between these gene signatures and clinical outcomes. Despite these efforts, the molecular mechanisms by which different genomic and transcriptomic profiles predict distinct patient-specific prognostic outcomes remain unknown.

Ordinary differential equations (ODEs)-based mechanistic models of the biochemical network are regarded as one of the most promising approaches for identifying regulatory mechanisms in biological systems[13,26–28]. Several attempts have focused on pan-cancer signaling networks to explore the mechanisms underlying heterogeneous responses[10,29], by integrating mechanistic modeling with transcriptome profiles acquired from cancer cell lines[30]. In these studies, experimental data on signaling activities, cell growth, and drug responses from more than 100 cell lines were used for model prediction. Training the model with the datasets allowed it to accurately predict cell-specific drug responses, as opposed to the untrained model[10,29]. Accordingly, these cell line profile studies suggest the potential of “patient-specific models” that can predict personalized prognoses and drug responses based on patients’ signaling and transcriptome profiles[31–34]. However, there are several challenges to overcome to achieve

this. Despite the availability of clinical transcriptome data from public databases, collecting signaling activity from each patient is not feasible due to the difficulties of culturing cells from cancer tissues. Additionally, pharmacological responses predicted by patient-specific models cannot be evaluated in living patients. Furthermore, as another fundamental issue, mathematical modeling typically necessitates specific mathematical expertise from the users. To be able to apply mathematical modeling to patient data analysis, we need a simpler, more readable format that many biologists can use for cancer classification.

1.3 Contribution of this thesis

The overall aim of this thesis is to improve patient-specific modeling and develop a novel method that overcomes the challenges mentioned above.

In this study, I developed a computational framework called Pasmopy (Patient-Specific Modeling in Python) (Figure 1.1). Pasmopy enables the conversion of text describing biochemical reactions (such as association, phosphorylation, and degradation) in signaling networks into ODE models without mathematical knowledge from the users. It also offers several biologist-friendly functions, such as the parameterization of patient models against the learning datasets obtained from cultured cell lines, the prediction of patient prognosis based on simulation outputs, the ability to investigate the molecular mechanisms underlying patient outcomes, and the ability to identify potential drug targets for individual patients.

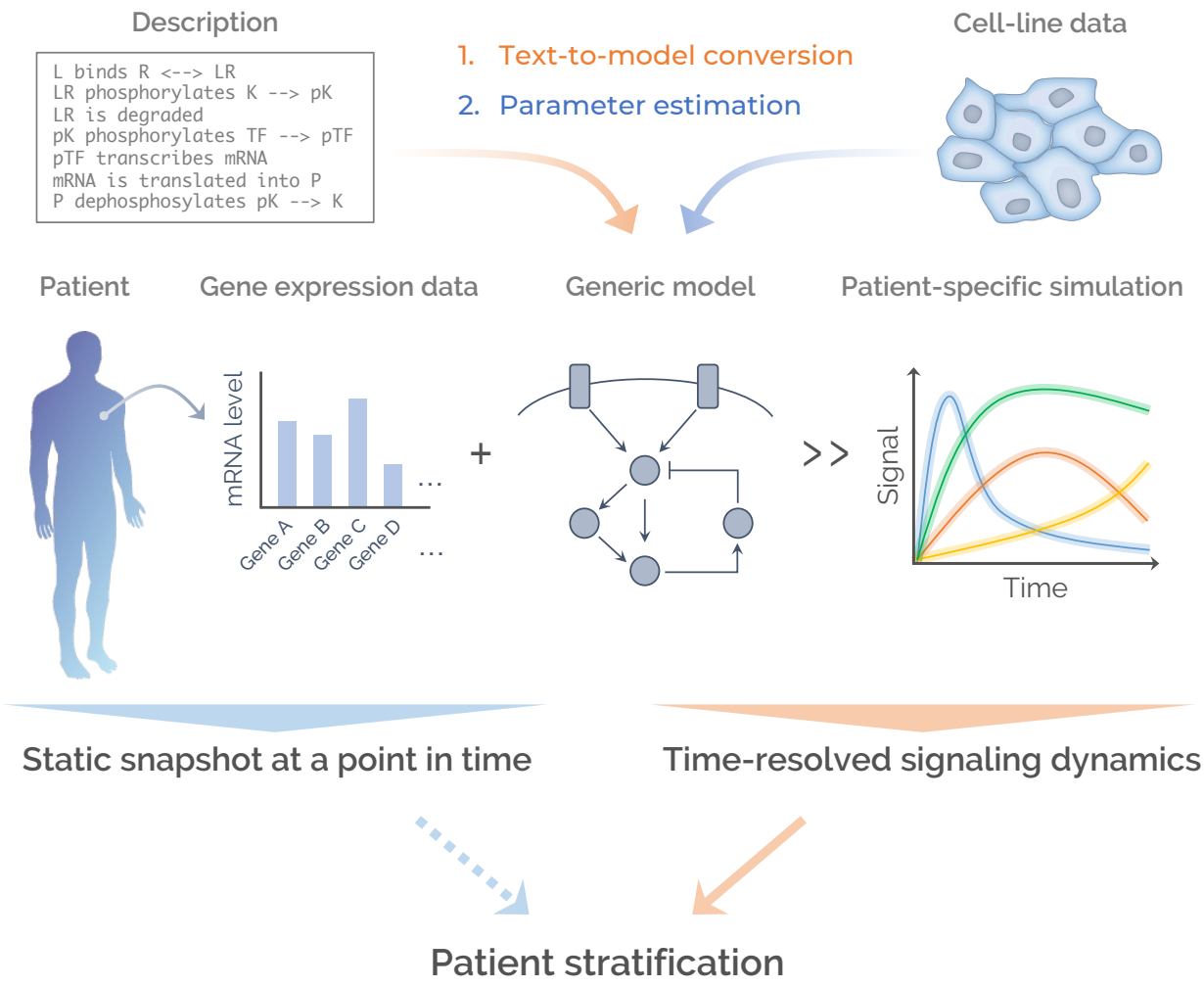


Figure 1.1: Pasmopy – a scalable framework for patient-specific modeling

Pasmopy is compatible with my previously developed modeling framework BioMASS. It provides useful tools for text-to-model conversion, parameter estimation, personalization of the model using transcriptome data, and patient stratification based on *in silico* signaling dynamics.

Using this tool, I developed a personalized kinetic model of the ErbB receptor signaling network. The model includes a series of biochemical reactions involved in ErbB receptor activation and c-Myc induction[35,36]. By combining individual patient transcriptome datasets obtained from TCGA[3] with personalized models, I succeeded in classifying TNBC patients into poor and good prognosis groups, based solely on the *in silico* Akt, ERK, and c-Myc dynamics of each patient. My models suggested that these subclusters can be classified by a simple metric: the epidermal growth factor receptor (EGFR) (ErbB1) expression ratio to other ErbB receptor families. Further analysis of the models implied that patients with poorer prognoses were more resistant to treatments targeting the EGFR. I also confirmed that the same model could stratify colon cancer patients[37] based on predicted signaling dynamics, indicating that these two cancers may share common regulatory mechanisms in the signaling network determining prognosis.

2 Materials and methods

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Antibodies		
Anti-c-Myc (phospho S62)	abcam	Cat#ab51156 RRID:AB_869189
Anti-rabbit IgG, HRP-linked Antibody	Cell Signaling Technology	Cat#7074S RRID:AB_2099233
Chemicals, peptides, and recombinant proteins		
DMEM (High Glucose)	Nacalai Tesque	Cat#08458-16
Fetal Bovine Serum	Sigma-Aldrich	Cat#F7524
Penicillin-streptomycin	Nacalai Tesque	Cat#09367-34
Deposited data		
CCLE drug-response data	Barretina et al., 2012	https://sites.broadinstitute.org/ccle/
CCLE RNAseq gene expression data (read counts)	Barretina et al., 2012	https://sites.broadinstitute.org/ccle/
TCGA-BRCA gene expression data (Htseq-Counts)	NIH GDC data portal	https://portal.gdc.cancer.gov/
TCGA-COAD gene expression data (Htseq-Counts)	NIH GDC data portal	https://portal.gdc.cancer.gov/
TCGA-BRCA Somatic mutation data – MuTect2	NIH GDC data portal	https://portal.gdc.cancer.gov/
LINCS: Basal profile of receptor tyrosine kinase signaling network measured by ELISA	Niepel et al., 2013	https://lincs.hms.harvard.edu/niepel_scisignal_2013/ HMS Dataset #20137
Experimental models: Cell lines		

Human: MCF7	ATCC	Cat#HTB-22; RRID:CVCL_0031
Human: BT-474	ATCC	Cat#HTB-20 RRID:CVCL_0179
Human: SK-BR-3	ATCC	Cat#HTB-30 RRID:CVCL_0033
Human: MDA-MB-231	ATCC	Cat#CRM-HTB-26 RRID:CVCL_0062
Software and algorithms		
Python 3.7.2	Python Software Foundation	https://www.python.org
pasmopy v0.1.0	This paper	https://github.com/pasmopy/pasmopy
biomass v0.5.2	Imoto et al., 2020	https://github.com/biomass-dev/biomass
numpy v1.19.2	Van Der Walt et al., 2011	https://numpy.org
scipy v1.6.2	Virtanen et al., 2020	https://scipy.org
pandas v1.2.4	pandas – Python Data Analysis Library	https://pandas.pydata.org
seaborn v0.11.2	Waskom, 2021	https://seaborn.pydata.org
Julia 1.6.2	The Julia Programming Language	https://julialang.org
BioMASS.jl v0.5.0	Imoto et al., 2020	https://github.com/biomass-dev/BioMASS.jl
R 4.0.2	The R Foundation	https://www.r-project.org
TCGAbiolinks v2.18.0	Colaprico et al., 2016	https://bioconductor.org/packages/TCGAbiolinks/
sva v3.38.0	Zhang et al., 2020	https://bioconductor.org/packages/sva/

biomaRt v2.46.3	Durinck et al., 2009	https://bioconductor.org/packages/biomaRt/
qvalue v2.22.0	Storey Lab	https://.bioconductor.org/packages/release/bioc/html/qvalue.html

2.1 Model development

The model used in this study is based on a system of ODEs. Most rate equations in the model were derived from the law of mass action. I applied the Michaelis–Menten and Hill equations for reactions describing (de)phosphorylation and transcription, respectively. The upstream signaling network model included ErbB receptor activation, the RAS-RAF-MEK-ERK cascade, and the Akt-PI3K pathway, which was adapted from the model by Birtwistle et al.[38] and integrated the c-Myc regulation process, which was constructed specifically for this study. The resulting model has 319 rate equations, 228 species, and 648 parameters. In the original model, it was assumed that the route of membrane recruitment did not affect the function of proteins, and a “membrane-localized state” was introduced to reduce the complexity of membrane recruitment. I did not use this expression, but I imposed parameter value constraints. In the rate equation, I assumed that the kinetic parameter describing the binding of downstream proteins to an adaptor protein, e.g., Grb2 and Gab1, is identical regardless of how the adaptor protein is recruited to the membrane. To study how the upstream activity of ERK and Akt controls c-Myc induction and activation, I added the process of c-Myc regulation so that ERK and Akt could activate and stabilize c-Myc, respectively[39].

2.2 Training datasets

I used time-series data on phosphorylated Akt, ERK, and c-Myc stimulated with EGF and HRG (eight time-points, up to 120 min) obtained from four breast cancer cell lines (MCF-7, BT-474, SK-BR-3, and MDA-MB-231). The datasets on phosphorylated Akt, ERK, and c-Myc were obtained from previous studies[40,41]. Before treatment with 10 nM EGF or HRG, the cells were synchronized by serum starvation for 16 h. The cells were lysed with BioPlex Lysis buffer (Bio-Rad Laboratories, Hercules, CA, USA), cell lysates were cleared by centrifugation (13,000 rpm, 15 min, 4°C), and the total protein concentration in the supernatants was determined using a protein assay reagent (Bio-Rad Laboratories). I adopted the transfer and normalization methods from previous studies[42,43] to minimize transfer errors and variability between the blots. The experimental data used in this study was obtained by Suxiang Zhang.

2.3 Parameter estimation

Of the 648 parameters, including kinetic constants and weighting factors, 220 were trained against the time series phospho-protein data and cell-line-specific transcriptomic data obtained from the Cancer Cell Line Encyclopedia (CCLE)[30]. During parameterization, gene expression level information was used to individualize the maximal transcription rate and nonzero initial conditions in each cell line (see Methods 2.4). I used BioMASS.jl[40] (version 0.5.0), which provides a Julia interface to the biomass parameter estimation. The parameters were trained using a global parameter estimation method called Differential Evolution (DE)[44] that minimizes the residual sum of squares between experimental

measurements and simulations. The optimization was stopped after the objective function value dropped below the 6.0 threshold, at which point the simulations agreed well with the experimental data in the four breast cancer cell lines. The results for all 30 independent parameter estimation runs were saved in.dat format using BioMASS.jl. For the original biomass framework in Python to recognize and read the optimized parameters, these results were converted into the standard binary file format in NumPy[45,46].

2.4 Individualization of the mechanistic model

The gene expression profiles for each cell line or patient were incorporated into the model through the following methods:

(i) When the initial amount of a species is zero and induced by upstream signals, e.g., *dusp* mRNA and *c-myc* mRNA:

The rate equation for transcription, *v*, is described by the Hill equation:

$$v = \frac{V \cdot [TF]^n}{K^n + [TF]^n}, \tag{1}$$

where *V* is the maximal transcription rate, *K* is the concentration of transcription factor ([TF]) producing a 50 % maximal response, and *n* is the Hill coefficient. In this case, the transcriptomic data was incorporated as the maximal transcription rate, *V*, using the following equation:

$$V = \sum_{i}^{N} \alpha_i \cdot x_i\,, \tag{2}$$

where N = 38 is the number of genes used in the mathematical model (see Table 3.1), x is the transcripts per million (TPM) value (relative log expression (RLE) normalized and post-ComBat), and α is the corresponding weighting factor for a gene to estimate transcription rate.

(ii) When the initial condition is not zero:

The transcriptomic data is used to estimate the initial value of protein, y_0 via the:

$$y_0 = \sum_{i}^{N} \beta_i \cdot x_i \,, \tag{3}$$

where β is the weighting factor to estimate the initial amounts of model protein species.

In both cases, the weighting factors were estimated during model parameterization.

2.5 Data processing in transcriptomic data integration

The criteria for sample selection in the stratification of TCGA-BRCA and TCGA-COAD, i.e., upper age limits, cancer stages, and the range of patients' total read counts, are described in Table 2.1.

Table 2.1: The criteria used for pre-processing TCGA-BRCA/CCLE-BREAST and TCGA-COAD/CCLE-BREAST samples

TCGA/CCLE	TCGA: Upper age limit	TCGA: Stages of cancer	Total read counts
BRCA/BREAST	59	Stage I, Stage IA, Stage IB, Stage II, Stage IIA, Stage IIB	Lower: 40,000,000 Upper: 140,000,000
COAD/BREAST	79	Stage I, Stage IA, Stage IB, Stage II, Stage IIA, Stage IIB	Lower: 10,000,000 Upper: 160,000,000

TCGA-BRCA and TCGA-COAD RNA-seq samples were downloaded using R TCGAbiolinks[47] version 2.18.0. The gene expression tables for HTSeq-based count retrieval were handled by TCGAbiolinks: `GDCquery` command. This repository contained 1222 RNA-seq samples that have been uniformly processed from row data. Specifically, samples donated from patients over the upper age limit, entering late cancer stages, or with noncancerous solid tissue, and duplicated samples were filtered from the data table.

The CCLE RNA-seq gene count matrix was downloaded from https://data.broadinstitute.org/ccle/CCLE_RNAseq_genes_counts_20180929.gct.gz, and samples other than those obtained from breast cancer cell lines were excluded from the data table. After merging these two matrices by ensemble gene ID, we executed the R ComBat-seq program[48] to adjust batch effects between the two datasets[49] (Figure 2.1). Since total read counts varied widely among samples, samples among the adjusted datasets with a total reads value lower than the lower bound (TCGA-BRCA/CCLE-BREAST: 40,000,000, TCGA-COAD/CCLE-BREAST: 10,000,000) or greater than the upper bound (TCGA-BRCA/CCLE-BREAST: 140,000,000, TCGA-COAD/CCLE-BREAST: 160,000,000) were excluded. After normalization of the library size using the RLE method[50], the count matrix was normalized using the TPM method. The gene lengths used in the TPM calculation[51] were determined as the differences between the start and end positions of each gene on the chromosomes retrieved from the R biomaRt package[52] version 2.46.3.

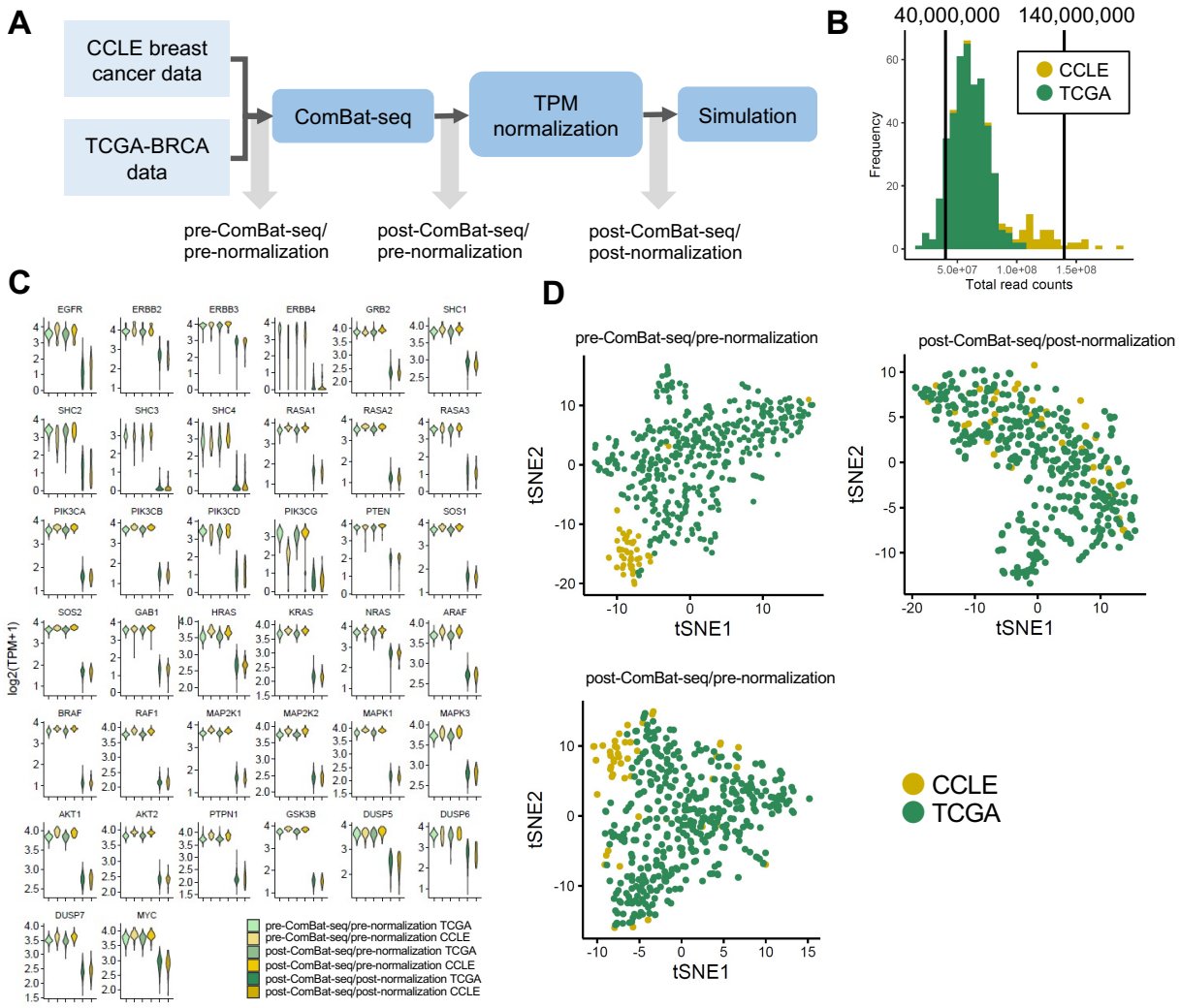


Figure 2.1: Pre-processing of the TCGA-BRCA and CCLE-BREAST datasets

(**A**) Procedure for the integration of transcriptomic datasets obtained from the TCGA and CCLE. (**B**) The cut-off boundaries used to exclude the variability of total read counts and the distribution of values of total read counts for TCGA-BRCA and CCLE-BREAST datasets. (**C**) Distribution of expression levels of two datasets at each step of the procedure. (**D**) tSNE visualization of RNA-seq data for each sample. tSNE coordinates were calculated using Rtsne (version 0.15) while using all available genes in each matrix.

To predict the ErbB signaling dynamics of the TCGA-COAD samples, I reused the parameter sets optimized in the breast cancer model (TCGA-BRCA). Based on previous works[15,38,39], I picked up 38 genes in the ErbB-c-Myc pathway, and their mRNA levels were incorporated into the mechanistic model (see Table 3.1). Since the normalized TPM values of the 38 genes in the four cell lines (MCF-7, BT-474, SK-BR-3, and MDA-MB-231) in the TCGA-BRCA: CCLE-BREAST matrix, *X*, are different from those in the TCGA-COAD: CCLE-BREAST matrix, *Y*, the values were transformed by multiplying a scaling factor, F_i, for the ith gene to reproduce the experimental observations in the four breast cancer cell lines. F_i is calculated using the following equation:

$$F_i = \frac{\max_i X_{i,\{\mathrm{MCF7,BT474,SKBR3,MADMB231}\}}}{\max_i Y_{i,\{\mathrm{MCF7,BT474,SKBR3,MADMB231}\}}}. \quad (4)$$

Note that this stage of data processing was done by Sawa Yamashiro.

2.6 Clustering breast cancer patients with gene expression level

After removing genes with zero expression levels in more than half of the samples from the log2-transformed matrix, the *p*-values between the cluster 1 and cluster 2 samples were calculated using Student's t-test. The q-values (FDR threshold: 0.05) were estimated from the *p*-values with the *qvalue* package (ver 2.22.0), and genes with a value lower than 0.05 were designated as differentially expressed genes (DEGs). For clustering based on gene expression, the expression levels of all genes were log2-transformed and then clustered by the expression level of a given gene (genes in the model, PAM50 genes, DEGs) using the k-medoids method. This analysis was done by Sawa Yamashiro.

2.7 Gene mutation analysis

The Mutation Annotation Format (MAF) files used to store the somatic mutations in each sample were summarized, analyzed, and visualized using the maftools Bioconductor package. The `GDCquery_Maf()` command of TCGAbiolinks[47] was used to obtain the MAF file of TCGA-BRCA detected by the MuTect2 pipeline[53]. After extracting the patient information used for clustering, we used the `oncoplot()` command to plot the mutations of the genes used in the model. This analysis was done by Sawa Yamashiro.

2.8 Extraction of response characteristics

Pasmopy provides the following response characteristics to classify personalized simulations: “max,” “AUC,” and “droprate.” The maximum and time-integrated responses are calculated using the `numpy.max()` and `scipy.integrate.simpson()` functions, respectively. The rate of decline (“droprate”), r, is defined as follows:

$$r = -\frac{(A_{end} - A_{max})}{(T_{end} - T_{max})}, \tag{5}$$

where T_{end} is the simulation end time (120 min), and T_{max} is the time to reach the maximum level. T_{max} was calculated using the `numpy.argmax()` function. A_{end} and A_{max} denote the normalized simulated values at 120 min and T_{max}, respectively.

I applied the maximum value of simulated dynamics for both EGF and HRG stimulations to classify samples. After calculating the standard score (*z*-score) of each feature, patients were classified based on the Euclidean distance of each patient using the *k*-medoids method. The prognostic score of each patient was given as *n*, where *n* is the

score of patients who deceased within *n* - 1 to *n* years and 20 is the score of patients who were alive.

2.9 Sensitivity analysis

Sensitivity analysis can be applied to identify which reactions in complex biochemical systems are important and which are not, such as determining which protein plays a key role in controlling the dynamics of the signaling output[54,55]. The sensitivity coefficients S_y were calculated using the following equation:

$$S_y = \partial \ln M / \partial \ln y_j , \quad (6)$$

where *M* is the signaling metric, i.e., the maximum level of phosphorylated c-Myc with EGF stimulation, and y_j is each nonzero species in the mechanistic model[8]. The sensitivity coefficients were calculated using finite difference approximations with 1 % changes in the initial conditions. A positive coefficient implies that variations in the input increase the value of the output, whereas a negative coefficient implies that variations in the input suppress the value of the output compared to the initial amounts. Large coefficients, whether positive or negative, indicate potentially significant control mechanisms. To calculate sensitivity coefficients, I used the `PatientModelAnalyses` class in Pasmopy version 0.1.0 with the `biomass_kws={"metric": "maximum", "style": "heatmap"}` options.

In this study, the maximum value of EGF-induced c-Myc activation during the observation time from 0 min to 120 min was collected for the following analysis.

2.10 Drug response data analysis

The drug response and gene expression data from the CCLE were downloaded from https://data.broadinstitute.org/ccle_legacy_data/pharmacological_profiling/CCLE_NP24.2009_Drug_data_2015.02.24.csv and https://data.broadinstitute.org/ccle/CCLE_RNAseq_genes_counts_20180929.gct.gz, respectively. First, RNA-seq count datasets from the CCLE were normalized using the TPM method after the library size of all samples had been converted with the RLE method. To study the effect of EGFR inhibitors on cancer cells, I used cell lines with high EGFR expression ratios relative to other ErbB receptor families (ErbB2, ErbB3, and ErbB4) by setting the minimum EGFR expression level to the median of all cell lines. Next, I divided the cell lines into three groups: (i) high (top 30 EGFR expression ratios), (ii) low (bottom 30), and (iii) middle (the other 169 cell lines). The efficacy and potency of a drug were simultaneously quantified by the "activity area"[30], whose values were extracted from the column "ActArea" in the drug response data. I used pandas v1.2.4 for loading data, scipy[56] v1.6.2 for the statistical test, and seaborn[57] v0.11.2 for data visualization.

3 Results

3.1 Development of BioMASS: a computational framework for modeling and analysis of signaling systems

Mechanistic modeling is a powerful method to investigate complex biological systems since it enables the integration of prior knowledge from the field with different datasets generated under a wide range of experimental conditions[58,59]. Although a significant number of modeling studies are dedicated to explaining the dynamic properties of cellular signaling, most biological processes described in these models are limited and do not comprehensively encompass signaling pathways in mammalian cells. Therefore, one of the current challenges in the field is to integrate these independently constructed models and, thus, understand a signaling system at the whole cell level[60–63]. However, this task can only be completed if some obstacles are overcome. For example, parameter estimation to reproduce experimental observations such as dose-response and time series data, becomes increasingly difficult as the model becomes larger[29,64–66].

To overcome this issue, I developed BioMASS (https://github.com/biomass-dev/biomass), a Python framework for Modeling and Analysis of Signaling Systems[40]. BioMASS is a user-friendly simulation tool for experimental biologists that is licensed under the Apache license 2.0. BioMASS supports (i) parameter estimation of ODE models, (ii) sensitivity analysis, and (iii) effective visualization of simulation results (Figure 3.1).

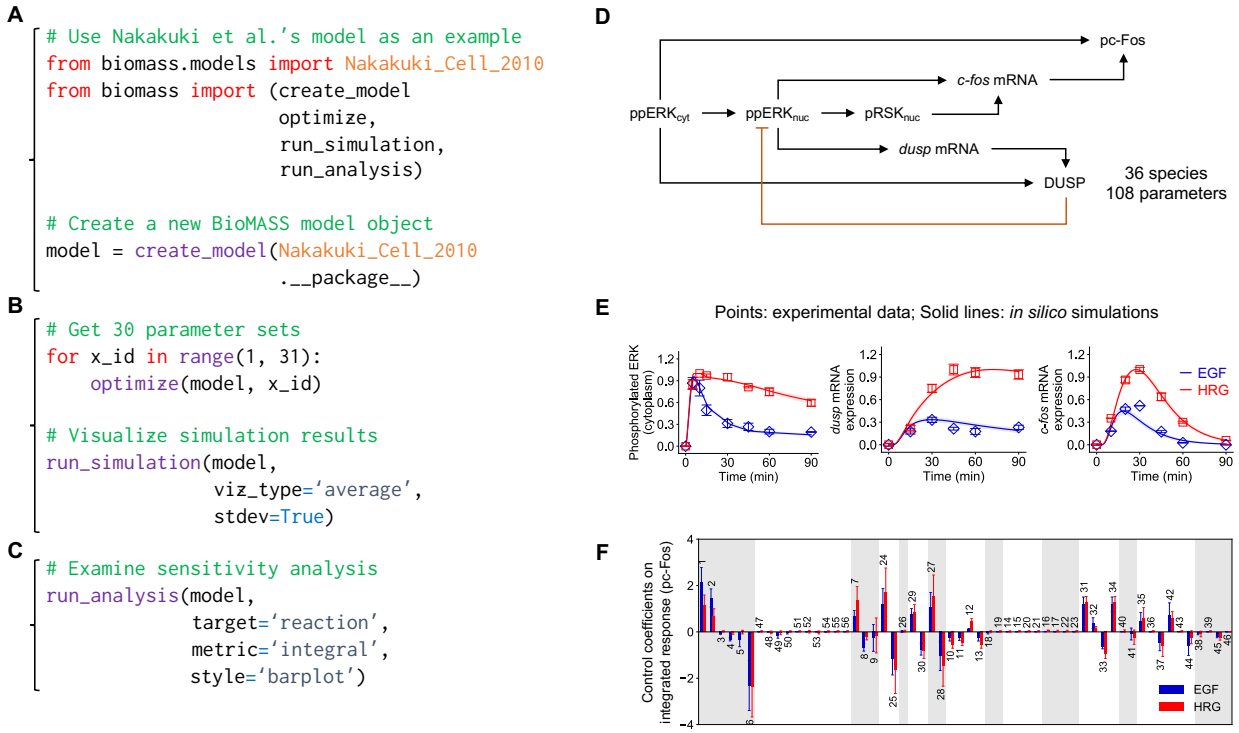


Figure 3.1: Overview of the BioMASS workflow

This workflow implements the model of Nakakuki et al. (**A–C**) Example code for model import (**A**), parameter estimation and visualization of simulation results (**B**) and sensitivity analysis (**C**). (**D**) Core reaction scheme of the Nakakuki model[15]. (**E**) Simulation results using optimized parameter sets. (**F**) Output of the sensitivity analysis.

In the modeling of biological systems, parameter estimation is one of the most critical steps[64,67–69]. Since differential equations describing biological networks are often complex and the solution space is non-convex and non-linear, the ODE optimization problem can contain multiple local optima. In addition, experimental datasets used for parameter estimation are often sparse and not continuous. To solve this, BioMASS enables the integrative evaluation of parameter estimation and sensitivity analysis of the dozens of parameter candidates. Users can perform sensitivity analysis[70] after model fitting to identify critical parameters, species, or regulations in the system of interest. By assessing parameter fitting and sensitivity analysis together, it is possible to identify critical or robust regions in the network or the origin of heterogeneity generated from the network.

3.2 A comprehensive model of c-Fos regulation triggered by the ErbB receptor activation

In a previous study, a mathematical model regarding c-Fos regulation in response to ErbB receptor activation was developed to understand how different types of growth factors give rise to unique gene expression patterns and cell fates[15]. The model provided novel insights into the control mechanism by which cells convert transient and sustained ERK signals into an all-or-none response of phosphorylated c-Fos. However, the input for this model was derived from an interpolation of the phosphorylated MEK signal quantified from western blots of the MCF-7 cell line, which makes assessing the quantitative effect of the input signal from the receptor difficult.

In the first part of this study, I integrated and constructed a comprehensive model of ErbB signal transduction[38] and the immediate-early gene response[15] to investigate the common regulatory mechanisms of ErbB receptor signaling on transcriptional control in multiple breast cancer cell lines. The integrated model contains several important processes in cancer, such as the MAPK pathway, the PI3K-Akt pathway, and c-Fos induction for cell proliferation and survival (Figure 3.2). I considered phosphorylated ERK (pERK), Akt (pAkt), and c-Fos (pc-Fos) to be key determinants of this network and, hence, measured the time course patterns of these proteins in four breast cancer cell lines (MCF-7, BT-474, SK-BR-3, and MDA-MB-231), which represent different breast cancer subtypes (Luminal A, Luminal B, HER2-positive, and triple-negative, respectively)[71].

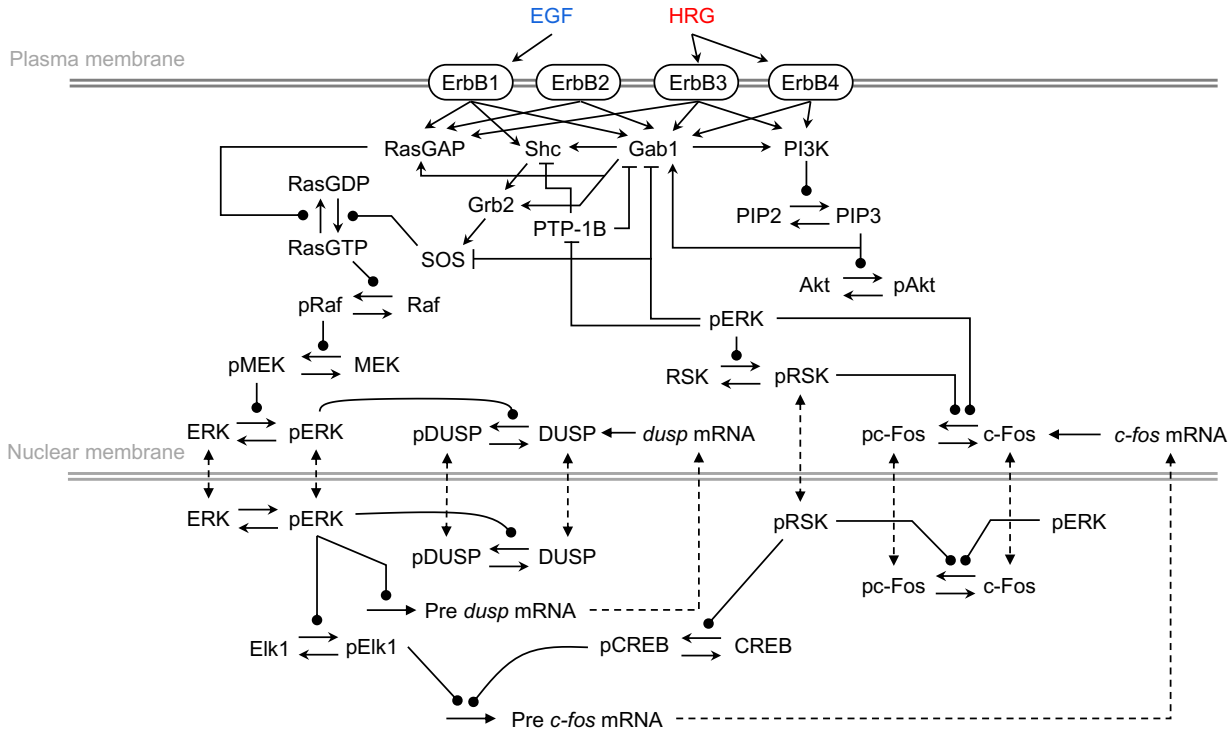


Figure 3.2: Influence diagram of the molecular interactions in the integrated model

3.3 Inferring intracellular signaling dynamics from individual gene expression profiles

For the construction of a cell-specific model for each cell line, I incorporated gene expression data from the CCLE[30] to infer the initial amounts of the 19 gene products described in the model. To make a model individualized to particular cell lines, Fröhlich et al. assumed that kinetic parameters, such as transport, binding, and phosphorylation rates, depend only on the chemical properties of the involved species and are identical across cell lines[29]. Furthermore, genome-wide association studies (GWAS)[23] have revealed that disease-related genomic mutations are significantly associated with gene expression levels in cis or trans-acting genes, and have been named Expression Quantitative Trait Loci (eQTL)[72]. In this model, I assume that differences in rate constants, which may originate from mutations in protein-coding regions, are negligible when compared to the gene expression levels of the cancer network. As a result, to infer the ErbB signaling dynamics, this study introduced the same assumption as Fröhlich et al.

I used time series data of pAkt, pERK, and pc-Fos from three breast cancer cell lines (MCF-7, BT-474, and MDA-MB-231) and autologous CCLE RNA-seq expression data to train the model parameters. To estimate parameters from the training datasets, I minimized the residual sum of squares using a genetic algorithm[73] used to obtain common parameter sets that can reproduce ErbB signaling dynamics in these three cell lines. I ran optimization simultaneously and ultimately obtained 30 good-fitting parameter sets (Figure 3.3A).

To test my hypothesis that gene expression levels are the major factor determining network dynamics, I attempted to predict signaling dynamics in another breast cancer cell line subtype, SK-BR-3, as well as in a HeLa cervix cell line, from their gene expression profiles (Figure 3.3B). Since I assumed that parameter values were identical across cell lines, I applied the trained model parameters constrained by time series data of pAkt, pERK, and pc-Fos in the MCF-7, BT-474, and MDA-MB-231 cell lines to the SK-BR-3 and HeLa cell line models. By using their autologous gene expression data and the trained parameters from the other three cell lines, the models correctly predicted experimental observations for both SK-BR-3 and HeLa cell lines (Figure 3.3C).

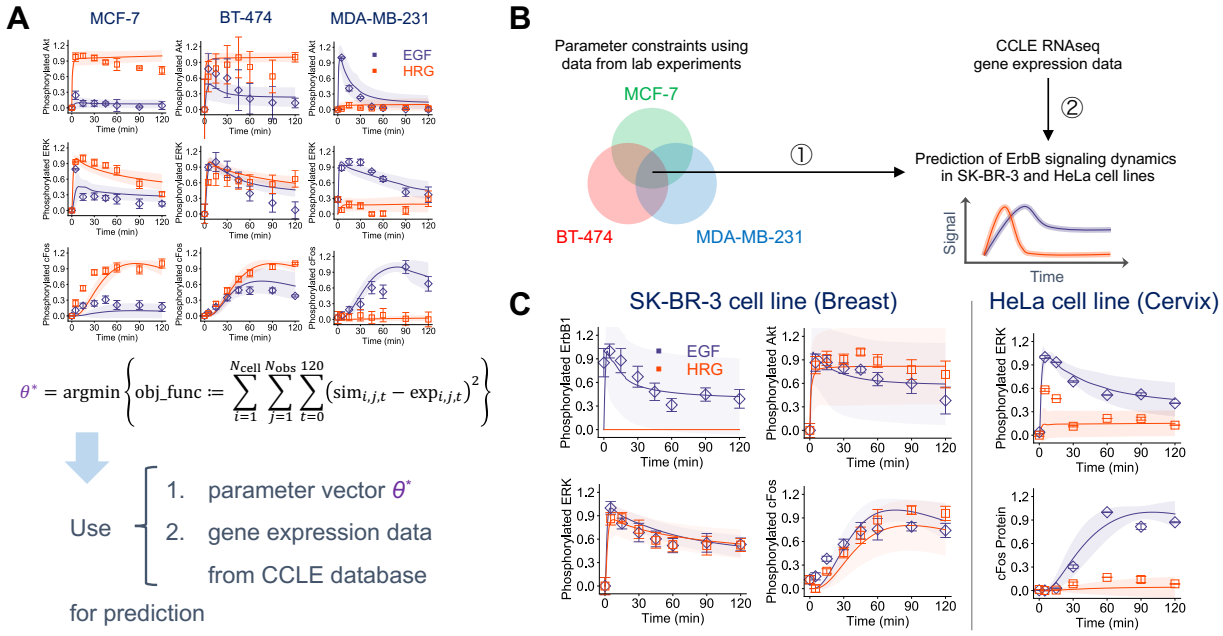


Figure 3.3: Model predicts intracellular signaling dynamics from gene expression data

(**A**) Results of parameter estimation. Points: (Blue diamonds, EGF; red squares, HRG) denote experimental data and lines denote the averaged simulation results of 30 parameters, and shaded areas indicate standard deviation. For all panels, error bars denote standard error for three independent experiments. (**B**) Strategy for model-based prediction. (**C**) Predicted signaling dynamics and experimental observations. Points: Experimental data (blue, EGF; red, HRG); solid lines: Averaged in silico simulation results of the 30 parameters; shaded areas: Simulation standard deviation.

Summarizing, my model-based predictions demonstrated that gene expression profiles in individual cell lines are tightly linked with their intracellular signaling dynamics (Figure 3.4). Based on this result, I applied this method to construct personalized kinetic models using patient-specific clinical datasets provided in TCGA database[3]. By using patient-specific modeling, I attempted to achieve better stratification of cancer patients and explore molecular mechanisms that distinguish between good and poor prognoses.

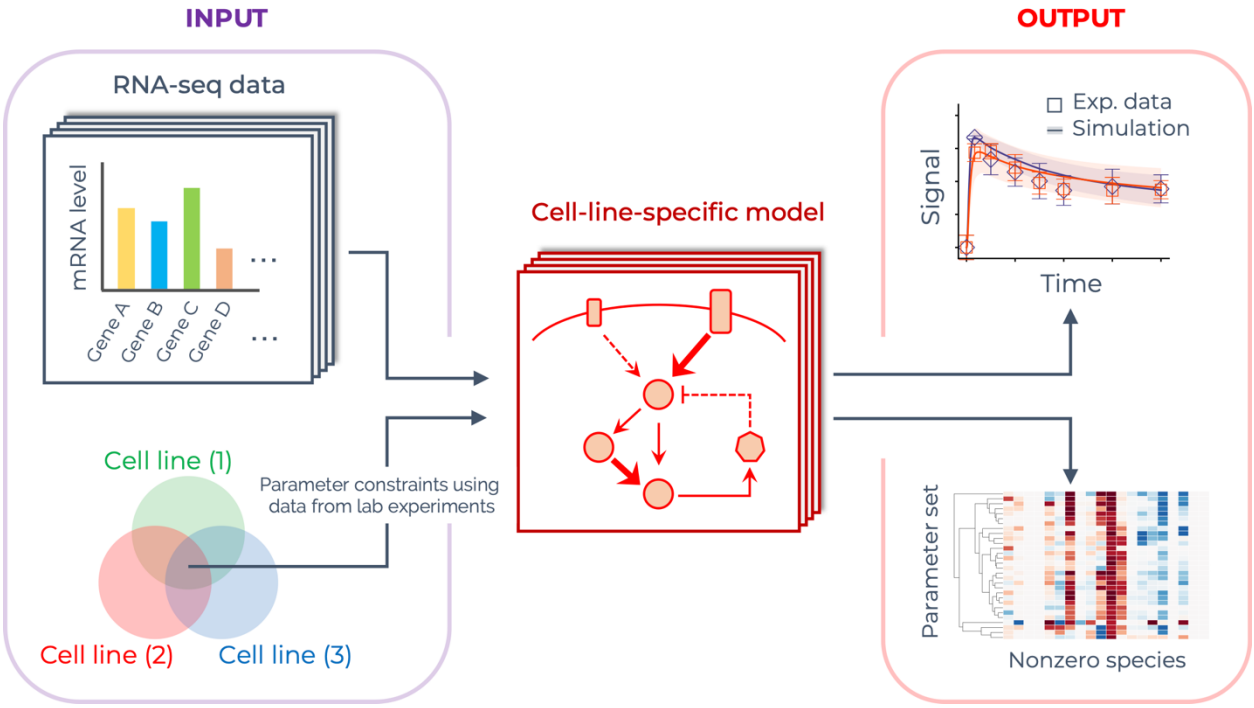


Figure 3.4: Summary of cell-line specific modeling. Adapted from Imoto *et al., Cancers* (2020)

3.4 Development of Pasmopy: a scalable computational toolkit for patient-specific modeling

The dynamics of signaling pathways play key roles in determining cell fate and cancer progression[74]. Therefore, the experimental analysis of patient response data is primarily required for the development of drugs targeting these pathways[75]. However, analyzing signaling dynamics using the patient tissues is generally difficult, even with advanced techniques such as the patient-derived xenograft (PDX) model. This is due to the limitations of the current PDX models, including the inability to reconstitute human immune cell systems, low success rates, and a high cost to establish cell lines and maintain the original cell properties[76–78]. From this perspective, patient-specific modeling that can simulate intracellular signaling dynamics derived from a patient *in silico* is one of the most promising approaches. However, some issues need to be addressed for personalized modeling: how to construct large-scale kinetic models, how to personalize hundreds of mechanistic models, and how to utilize them as biomarkers. To address these issues, I created Pasmopy, a scalable toolkit for *in silico* patient-specific mathematical modeling[41] (Figure 3.5).

Pasmopy offers the following unique features: (i) mechanistic model construction from human interpretable texts and sentences regarding biochemical systems without mathematical modeling knowledge (Figure 3.6A), (ii) model personalization using transcriptome data of each patient, (iii) patient outcome prediction based on *in silico* signaling dynamics, e.g., amplitude, duration, and area under the curve; and (iv) sensitivity analysis for prediction of potential drug targets. To enable the construction of

large-scale kinetic models with ease, I developed Text2Model by which human interpretable texts on biochemical events are converted into an executable model. Text2Model is a Python class and is available in Pasmopy and BioMASS (see Appendix 5.2.1). It currently contains a list of 14 reaction rules on gene regulation and biochemical reactions, including binding, dissociation, phosphorylation, transcription, translation, synthesis, degradation, and translocation, which can be automatically converted into kinetic equations (Figure 3.6B). Importantly, users can add new terminology and reaction rules. Pasmopy is compatible with BioMASS[40], which allows parameterization and network analysis of large-scale biological models and is more specialized for personalized modeling.

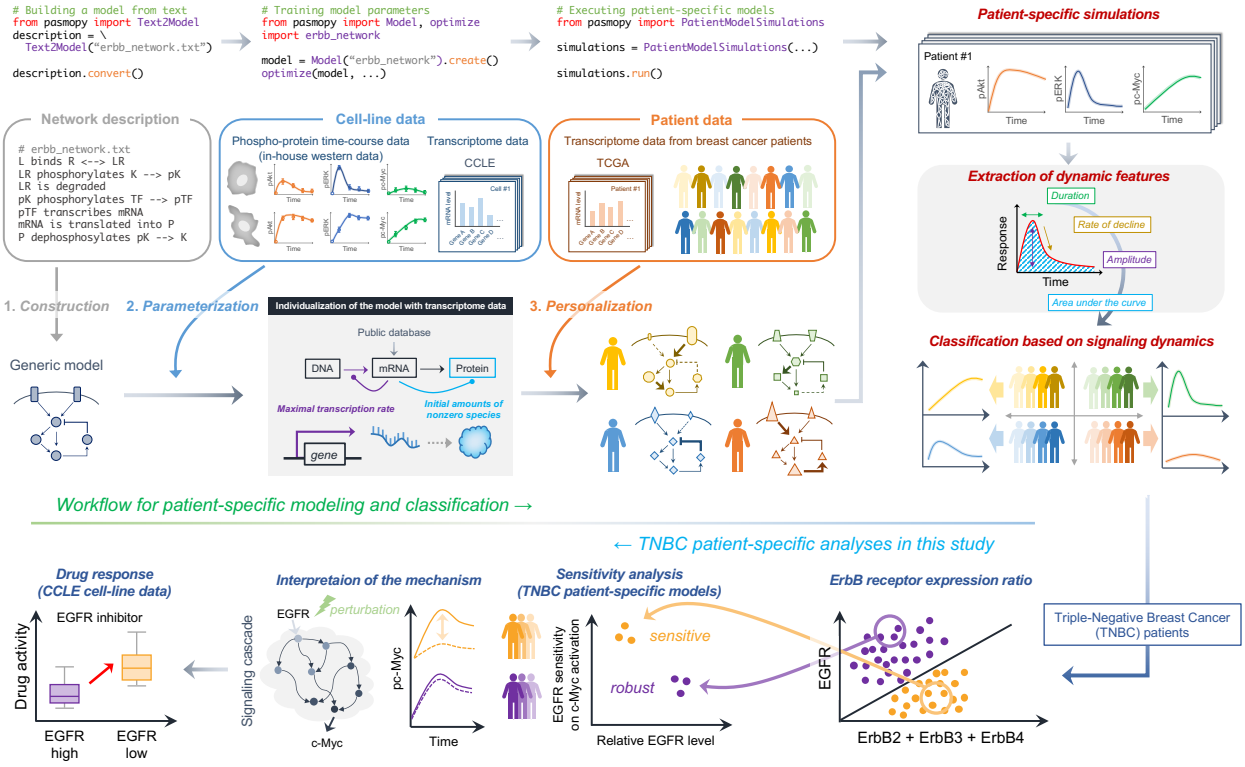
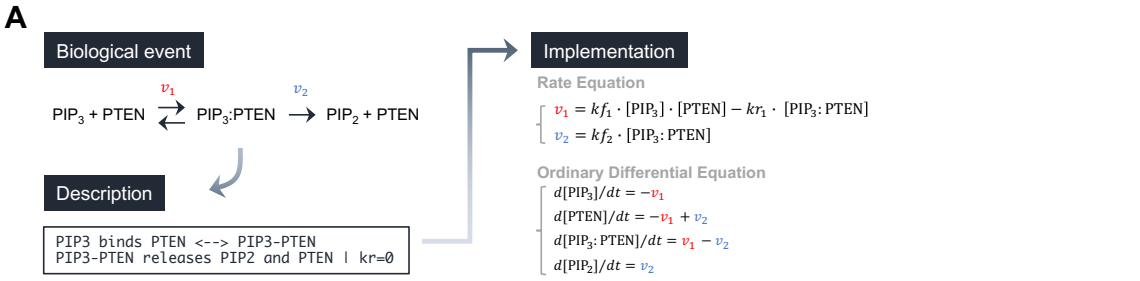


Figure 3.5: Overview of the workflow

A workflow for identifying cancer prognostic factors based on signaling dynamics from mechanistic modeling. A text file describing the biochemical reactions is converted into an executable model (1. Construction). The model parameters are trained on phospho-protein time-course data obtained from growth factor-stimulated cultured cell lines (2. Parameterization). The model is personalized by incorporating individual gene expression profiles (3. Personalization). The patients are classified based on in silico signaling responses from personalized simulations. In this study, the patient group with triple-negative breast cancer (TNBC) was further analyzed (bottom, right to left). Based on the examination of signaling properties, the patients with TNBC were classified into two subclusters by ErbB receptor expression ratios. Sensitivity analysis indicated that patients with higher EGFR expression ratios were less sensitive to EGFR inhibitors. This hypothesis was validated using drug-response data obtained from cancer cell lines.

In this study, I constructed a mathematical model of the ErbB receptor signaling network[38] and c-Myc induction[39] (Figure 3.6C). The model includes the activation and dimerization of four ErbB receptors (ErbB1 / EGFR, ErbB2, ErbB3, and ErbB4), the Ras-ERK cascade, and the Akt-PI3K pathway, which was adapted from the model by Birtwistle et al.[38] and integrated the process of c-Myc induction and stabilization by ERK and Akt signals[39], which was constructed specifically for this study. The resulting model has 319 rate equations, 228 species, and 648 parameters. Of the 648 parameters such as kinetic constants and weighting factors, 220 were estimated from phospho-protein time-course data obtained from four breast cancer cell lines stimulated with EGF or HRG for up to 120 min (see below and Materials and Methods section).

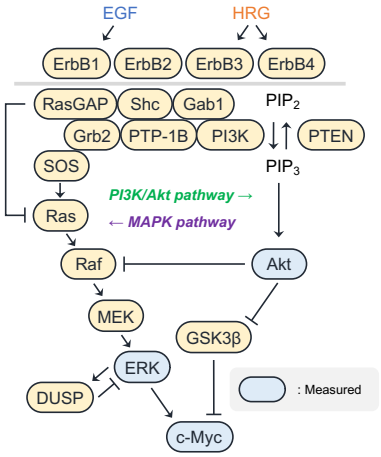


B

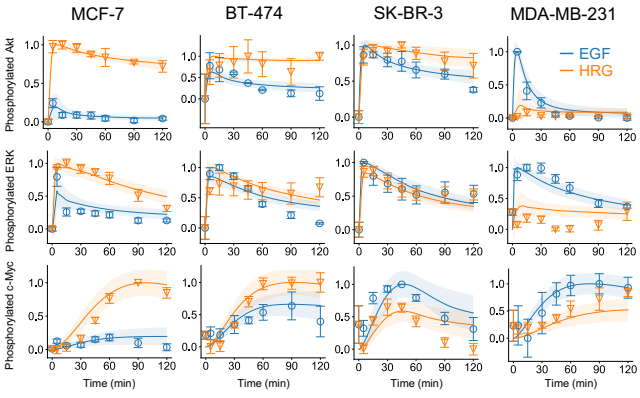
Event & Word	Rate equation	Mass balance
A + B ⇄ AB A binds B <--> AB	Mass action $v = kf[A][B] - kr[AB]$	d[A]/dt = - v d[B]/dt = - v d[AB]/dt = + v
B → A → Ap B phosphorylates A --> Ap	Michaelis-Menten equation $v = V[B][A] / (K + [A])$	d[A]/dt = - v d[Ap]/dt = + v
Ap → A Ap is dephosphorylated --> A	Michaelis-Menten equation $v = V[Ap] / (K + [Ap])$	d[A]/dt = + v d[Ap]/dt = - v
B → a B transcribes a	Hill equation $v = V[B]^n / (K^n + [B]^n)$	d[a]/dt = + v
a → A a is translated into A	First-order kinetics $v = kf[a]$	d[A]/dt = + v
∅ → A A is synthesized	Zero-order kinetics $v = kf$	d[A]/dt = + v

C

```
EGF binds ErbB1 <--> EGF_ErbB1
HRG binds ErbB3 <--> HRG_ErbB3
HRG binds ErbB4 <--> HRG_ErbB4
            .
            .
            .
RasGTP phosphorylates Raf --> RafP
RafP phosphorylates MEK --> MEKP
MEKP phosphorylates ERK --> ERKP
            .
            .
            .
ERKPP phosphorylates cMyc --> cMycP
GSK3b phosphorylates cMycP --> cMycPP
cMycPP is degraded
```



D



E

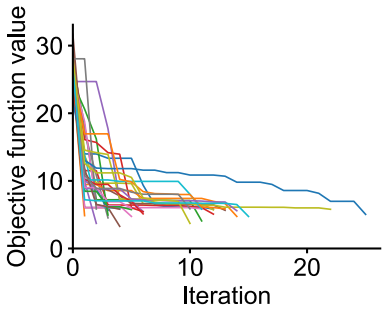


Figure 3.6: Construction and parameterization of the mechanistic model

(**A**) The strategy for implementing ordinary differential equations (ODEs) from the text descriptions of the biological events. (**B**) Representative biological events and words that can be converted into rate equations and ODEs. (**C**) Cancer signaling network and its conversion into an ODE model in this study. (**D**) The model parameter was trained on time-series Akt, ERK, and c-Myc phosphorylation levels obtained from four breast cancer cell lines: MCF-7, BT-474, SK-BR-3, and MDA-MB-231 stimulated with growth factors. The points (blue squares, EGF; orange triangles, HRG) denote experimental data, solid lines denote simulations, and shaded areas denote standard deviation. For all panels, error bars denote standard error for three independent experiments. (**E**) Objective function traces from 30 optimization runs.

3.5 Using transcriptomic data to personalize the mechanistic model

Modeling biological systems usually requires the initial abundance of chemical species and the kinetic parameters of the reaction. Here, to determine the kinetic parameters of patient-specific models, I first assumed that the reaction parameters are unique to the molecular species involved in a reaction event and remain the same even if gene mutations are present in the species. Ultimately, instead, I assumed that such genomic mutations are reflected in the gene expression signatures. eQTL[72] analysis and transcriptome-wide association studies (TWAS)[23] which link genomic mutations to gene expression signatures, empirically support this assumption. Accordingly, unknown parameters of the model that are common to all patients and cultured cell lines were obtained by fitting the model to the phospho-protein time-course data obtained from the cultured cell lines (Figure 3.6D).

In brief, the ErbB network model has been trained against the growth factor-stimulated time-course datasets of phosphorylated Akt, ERK, and c-Myc obtained from MCF-7, BT-474, SK-BR-3, and MDA-MB-231 cancer cell lines (which represent four breast cancer subtypes: luminal A, luminal B, HER2+, and triple-negative, respectively), along with their corresponding CCLE[30] transcriptome data, which are used for determining the nonzero initial conditions (protein levels of the species) in the model (see below and gene list in Table 3.1). By minimizing the objective function, i.e., the residual sum of squares between simulation and experimental measurements, 30 good-fitting parameter sets that reproduced experimental observations in these four breast cancer

cell lines were obtained (Figure 3.6E). These 30 parameter sets were also used as kinetic parameters for the patient model.

Table 3.1: Gene list for predicting each protein level or maximal transcription rate

The TPM value for each gene was used to estimate the nonzero initial conditions of model species (ErbB1, ErbB2, ErbB3, ErbB4, Grb2, Shc, PI3K, PTEN, SOS, Gab1, Ras, MEK, ERK, Akt, Raf, RasGAP, PTP1B, and GSK3β) or the maximal transcription rate (DUSP, c-Myc).

Protein	Gene symbol
ErbB1	*EGFR*
ErbB2	*ERBB2*
ErbB3	*ERBB3*
ErbB4	*ERBB4*
Grb2	*GRB2*
Shc	*SHC1*
	SHC2
	SHC3
	SHC4
PI3K	*PIK3CA*
	PIK3CB
	PIK3CD
	PIK3CG

Protein	Gene symbol
PTEN	*PTEN*
SOS	*SOS1*
	SOS2
Gab1	*GAB1*
Ras	*HRAS*
	KRAS
	NRAS
MEK	*MAP2K1*
	MAP2K2
ERK	*MAPK1*
	MAPK3
Akt	*AKT1*
	AKT2

Protein	Gene symbol
Raf	*ARAF*
	BRAF
	RAF1
RasGAP	*RASA1*
	RASA2
	RASA3
PTP1B	*PTPN1*
GSK3β	*GSK3B*
DUSP	*DUSP5*
	DUSP6
	DUSP7
c-Myc	*MYC*

To personalize the model, individual TCGA transcriptome data were analyzed and used to infer the initial amounts of nonzero species or maximal transcription rate for each patient. Since I used the cultured cell line data to estimate the parameters of the model, I needed to normalize the patient's transcriptome data to the data obtained from cultured cell lines. First, transcriptome profiles of 413 breast cancer patients and 51 breast cancer cell lines were obtained from TCGA[3] and CCLE[30], respectively, and their batch effects were removed using ComBat-seq[48] (Figure 2.1). By performing this step, cell line transcriptomes could be merged with the patient transcriptomes, and the parameters estimated from phospho-protein cell line data could also be used as patient parameters. Several samples were excluded due to low total read counts in their sequence data (see Methods and Figure 2.1B), finally resulting in 377 patient data points for modeling (Table 2.1). To make the models patient-specific, the clinical transcriptomic data for 38 genes (see gene list in Table 3.1) were incorporated into the model as the maximum transcription rate or the initial number of nonzero species (see Methods). If the initial value of a model species is zero and its expression was induced by upstream signals, e.g., *c-myc* mRNA or *dusp* mRNA, the maximal transcription rate can be estimated from its mRNA level. Unless stated otherwise, transcriptome data, including mRNA levels, was used to estimate the translated protein level. This way, I computationally predicted protein levels from their corresponding mRNA levels in TCGA. I confirmed that the simulated protein levels were consistent with the experimentally measured protein levels of four breast cancer subtypes in the Library of Integrated Network-based Cellular Signatures (LINCS) database[79] (Figure 3.7).

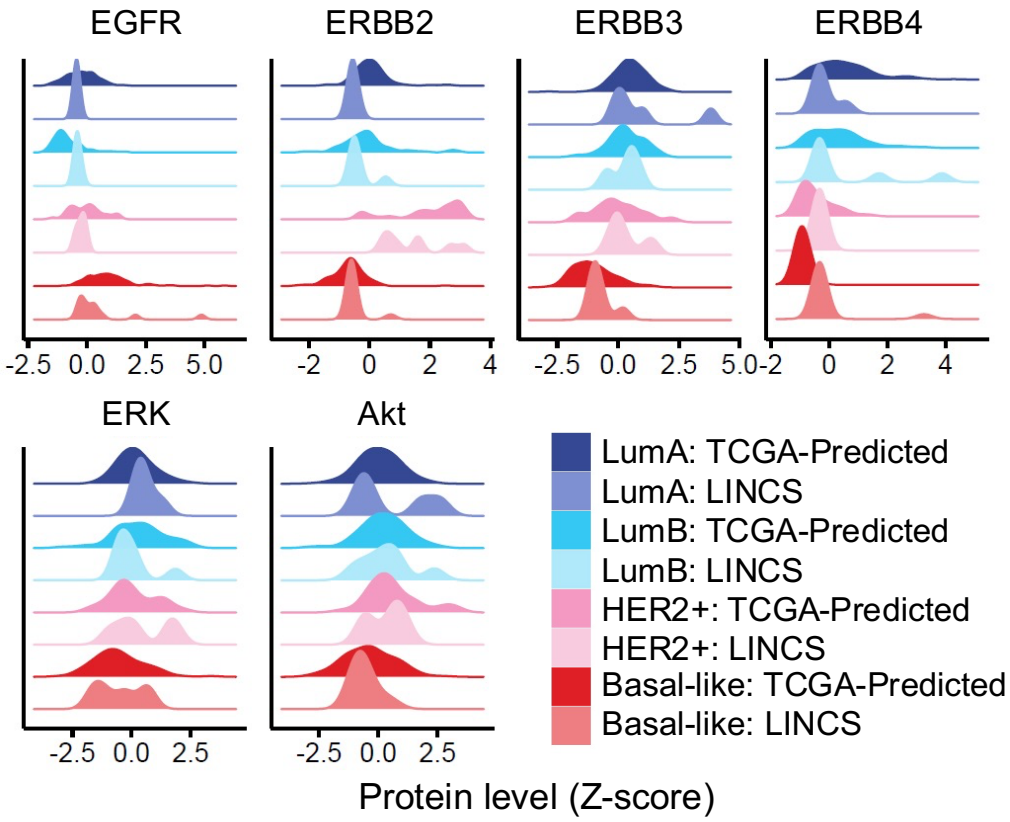


Figure 3.7: Comparison of predicted and actual protein levels for each subtype

Distribution of the protein levels predicted in the model and those measured using ELISA extracted from a previous study by Niepel et al (Niepel et al., 2013) for each PAM50 subtype. The predicted protein levels were computed as the weighted sum of the TPM (Transcripts Per Million) values for each gene. The information about cell-line PAM50 subtype classification was obtained from a study by Jiang et al (Jiang et al., 2016).

3.6 Stratification of TNBC patients based on signaling dynamics

After determining the model parameters, I performed numerical simulations to predict how each patient would respond to EGF and HRG stimulation *in silico*. I performed simulations for 377 patients and extracted quantitative information, such as amplitude, duration, drop rate, and cumulative response (see Methods for their definition), from the *in silico* dynamics of Akt, ERK, and c-Myc activation in each patient. This was followed by clustering patients based on their dynamic features (Figure 3.8). Among these characteristics, I used amplitude (the maximum activation level), for the classification of breast cancer patients. Even though this dynamic feature cannot distinguish Luminal A and Luminal B subtypes, it could distinguish TNBC from other subtypes (Figure 3.9A). Notably, my network-based classifier divided TNBC patients into two clusters: clusters 1 and 2 for patients with poor and good prognoses, respectively (Figure 3.9B and C). A classical PAM50 classification method[80–82] based on the expression signatures of 50 genes was suitable for subtype classification but not for the prediction of TNBC prognosis (Figure 3.10).

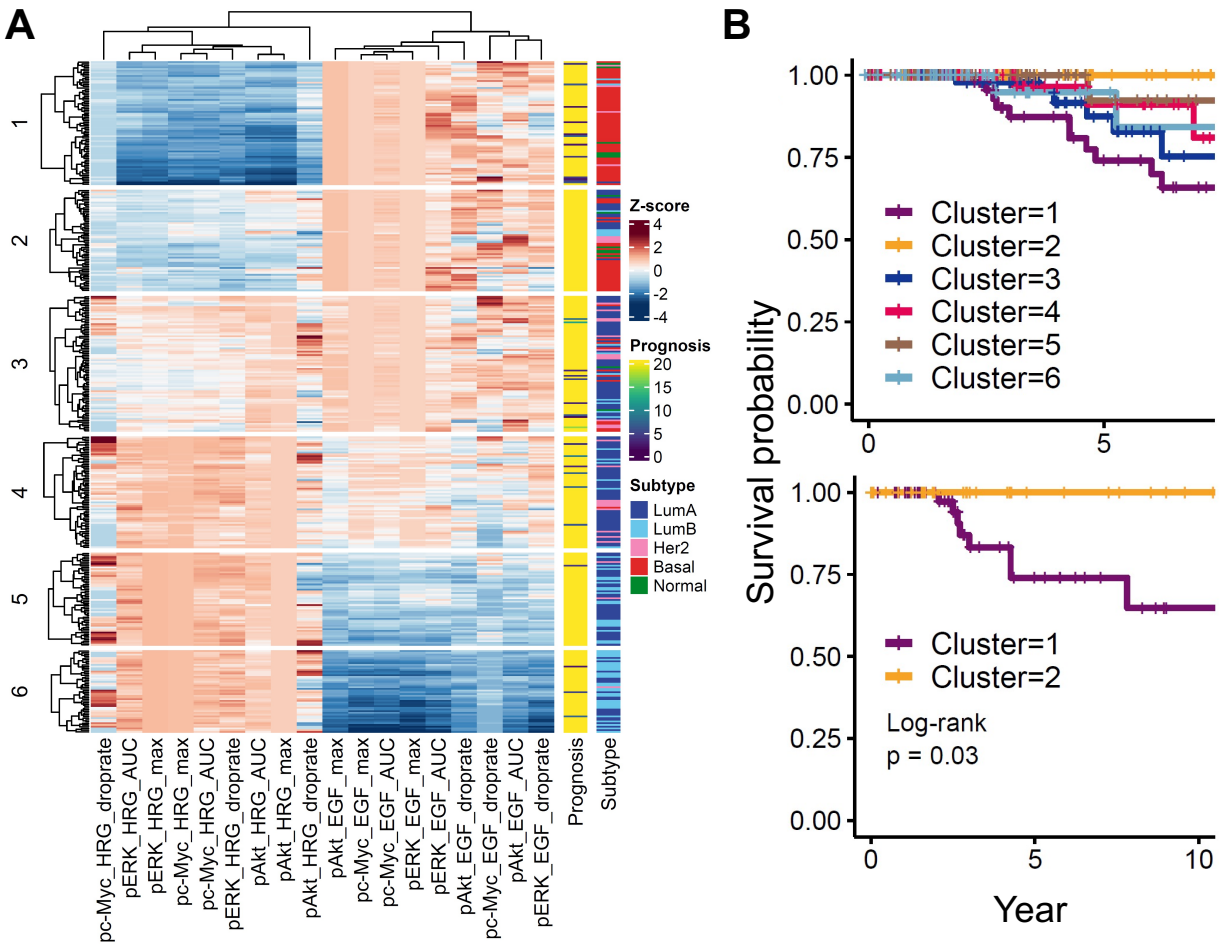


Figure 3.8: Clustering of patients by different types of dynamical features in Pasmopy

(**A**) Heatmap representing the stratification of breast cancer patients based on features provided in Pasmopy. Each row corresponds to one patient. (**B**) Kaplan-Meier survival curves of all patients in all clusters (top) and of patients with the basal-like subtype in clusters 1 and 2 (bottom).

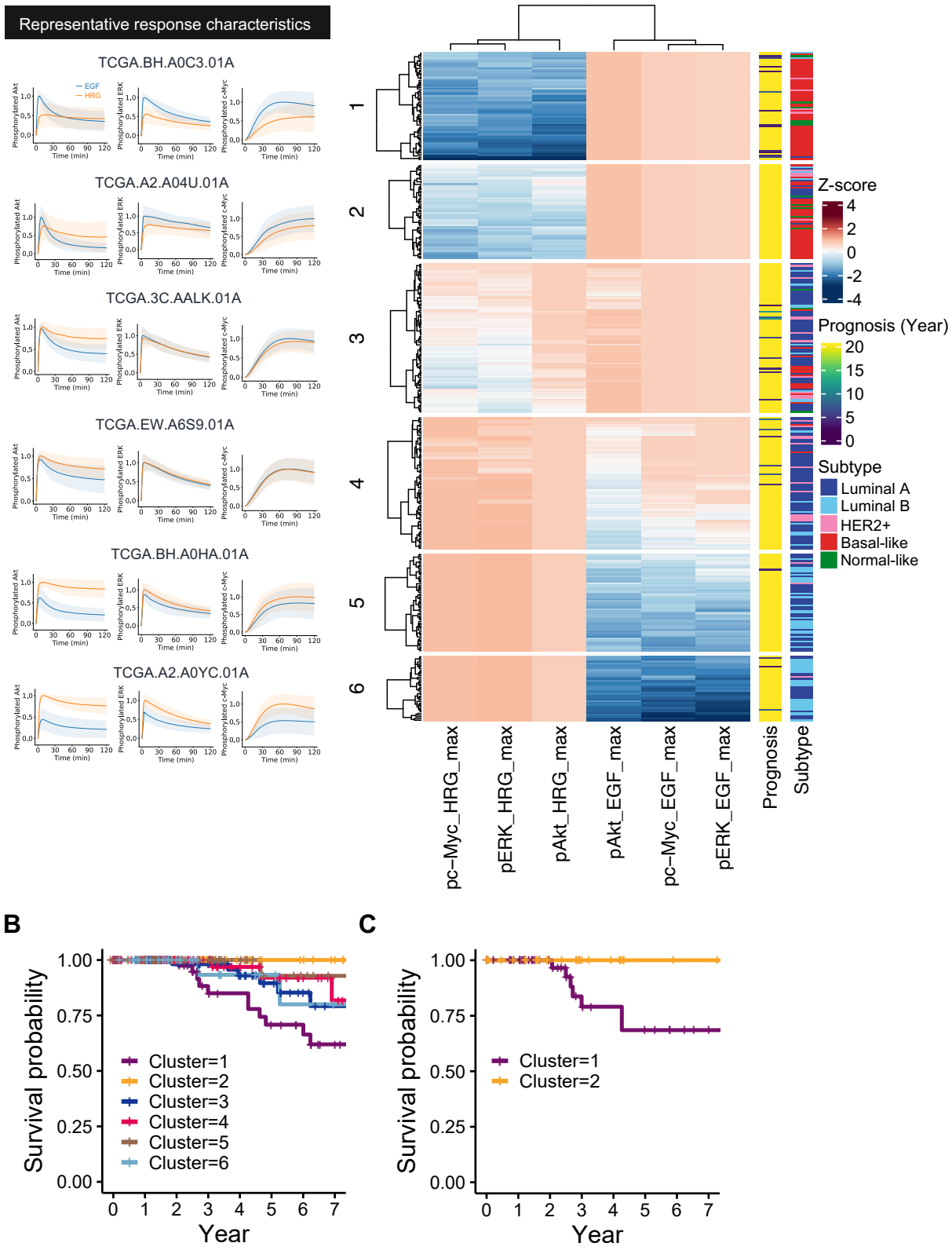

A
Representative response characteristics
TCGA.BH.A0C3.01A
TCGA.A2.A04U.01A
TCGA.3C.AALK.01A
TCGA.EW.A6S9.01A
TCGA.BH.A0HA.01A
TCGA.A2.A0YC.01A
EGF
HRG
Phosphorylated Akt
Phosphorylated ERK
Phosphorylated c-Myc
Time (min)
Z-score
Prognosis (Year)
Subtype
Luminal A
Luminal B
HER2+
Basal-like
Normal-like
pc−Myc_HRG_max
pERK_HRG_max
pAkt_HRG_max
pAkt_EGF_max
pc−Myc_EGF_max
pERK_EGF_max
Prognosis
Subtype
B
C
Survival probability
Year
Cluster=1
Cluster=2
Cluster=3
Cluster=4
Cluster=5
Cluster=6

Figure 3.9: Stratification of triple-negative breast cancer (TNBC) patients based on ErbB signaling dynamics

(**A**) The patients are classified based on personalized simulations. The prognostic score for patients who deceased within n-1 to n years are donated by n, and patients who were alive after 20 years are denoted in yellow. The representative signal response characteristics were extracted from the topmost portion of each cluster. The blue and orange solid lines denote simulations with EGF and HRG stimulation, respectively. Shaded areas denote standard deviation. (**B** and **C**) Kaplan-Meier survival curves of all patients for all clusters (**B**) and of patients with the basal-like subtype for clusters 1 and 2 (**C**).

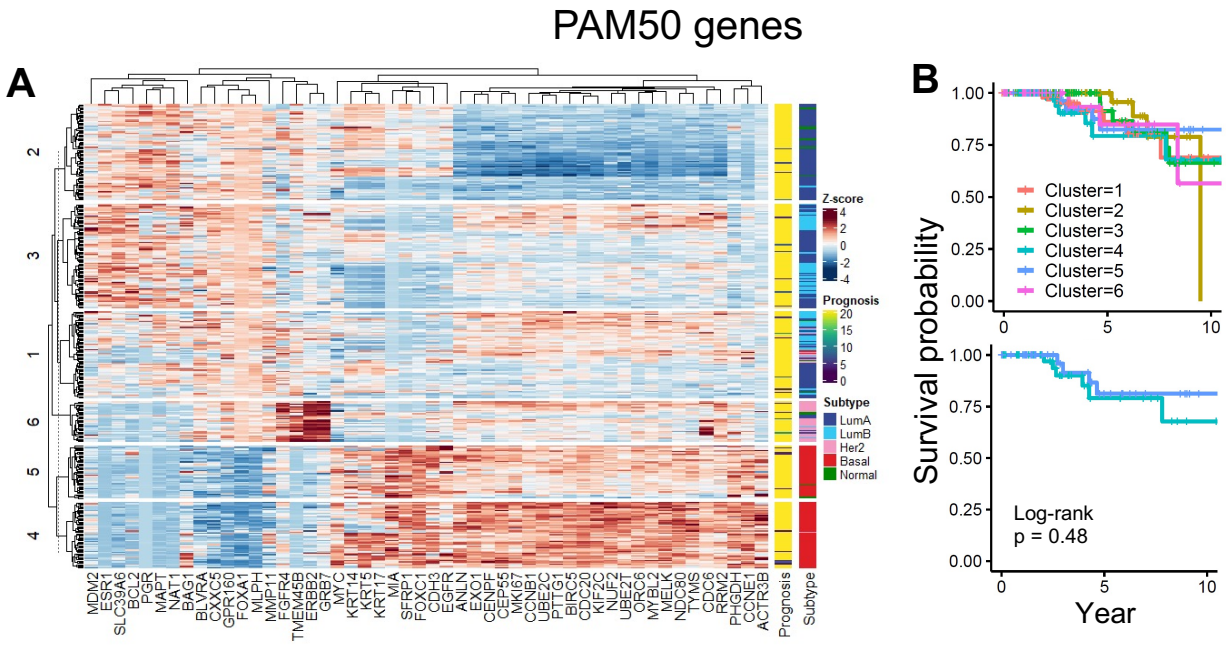


Figure 3.10: Clustering of patients based on PAM50 gene expression levels

(**A**) Clustering of patients based on the expression level of PAM50 (Nielsen et al., 2010) gene sets. mRNA levels (log2(TPM+1)) for each gene were normalized to the Z-score; red indicates higher expression and blue indicates lower expression. (**B**) (top) Kaplan-Meier survival curves for patients in each cluster; (bottom) Kaplan-Meier survival curves of patients with the basal-like subtype in clusters 4 and 5 in which more than half of the patients consist of the basal-like subtype. The p-value was obtained using log-rank test.

To identify crucial genes distinguishing clusters 1 and 2, I checked 253 differentially expressed transcripts (see Methods for the criteria for gene selection). However, there was no clear trend between these two clusters (Figure 3.11). From this result, I concluded that dynamic modeling is more suitable than the standard gene expression profiles for the stratification of TNBC patients and that the *in silico* signaling dynamics of the ErbB signaling network can be utilized as a prognostic marker for TNBC.

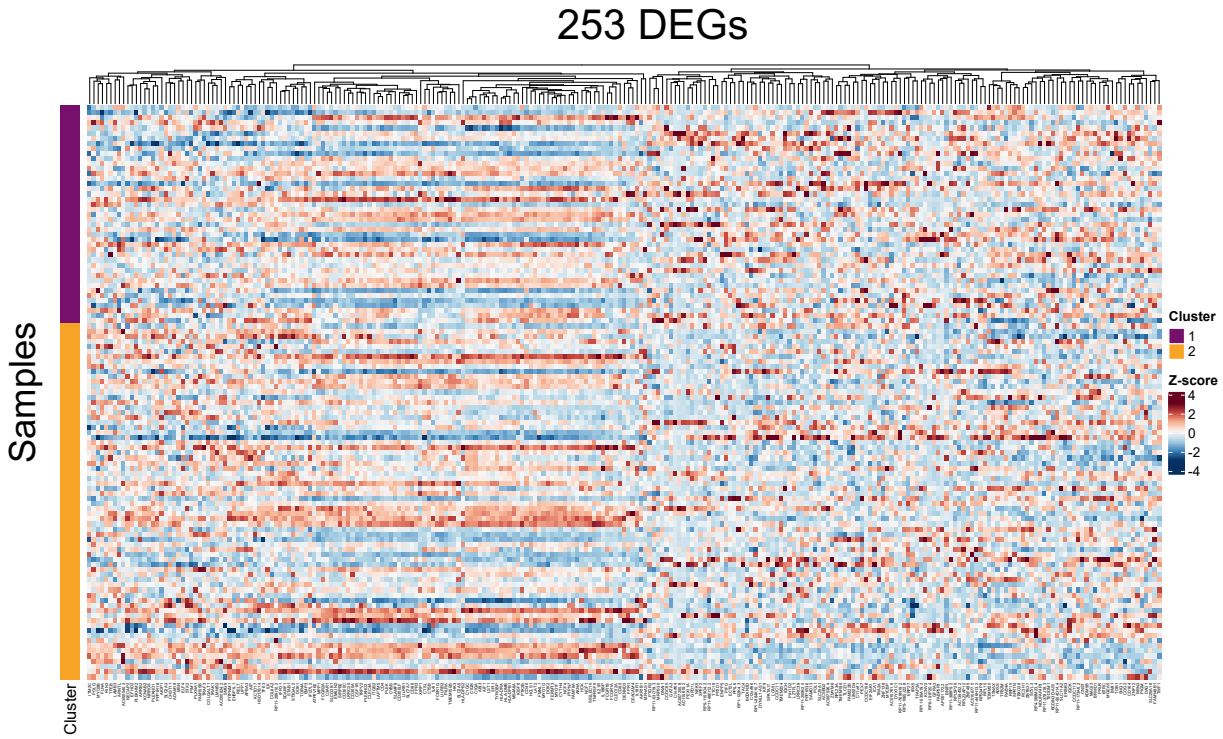


Figure 3.11: Patients clustering with differentially expressed genes (DEGs)

Heatmap of differentially expressed transcripts associated with cluster 1 and cluster 2. mRNA levels (log2(TPM+1)) for each gene were normalized to the Z-score.

3.7 Identification of the mechanisms affecting patient prognosis

Our clustering results showed that TNBC patients displaying poor prognoses were associated with lower Akt, ERK, and c-Myc activities under HRG stimulation. Together with the model structure, I hypothesized that the HRG receptors (ErbB3, ErbB4, and their heterodimerization partner ErbB2) in this patient group could not efficiently transmit the downstream signal due to competitive interference from higher levels of EGFR. Consistent with this hypothesis, further analysis showed that the expression ratios of the EGFR to the ErbB2, 3, and 4 receptors were higher in the poor prognosis group (Figure 3.12A and B). This result was also supported by the protein abundances predicted from the models (Figure 3.13). To further investigate the mechanisms that distinguish prognosis, I randomly sampled patients from each group and performed a sensitivity analysis that examined how perturbations to the initial model species conditions (inferred from gene expression level) affected c-Myc activity (model output). The results indicated that higher EGFR expression is associated with lower sensitivity to EGFR inhibitors (Figure 3.12C).

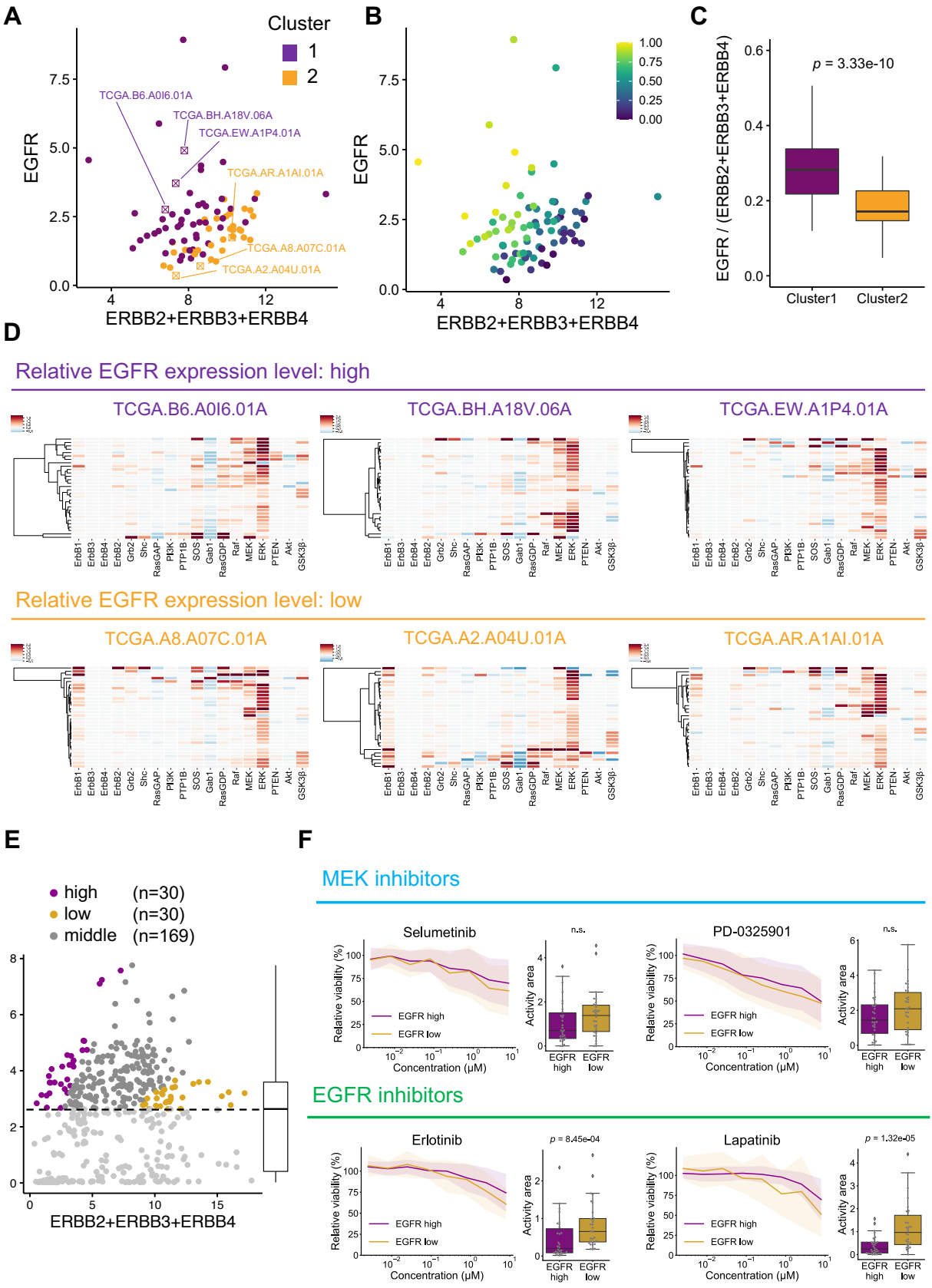

A
Cluster
1
2
TCGA.B6.A0I6.01A
TCGA.BH.A18V.06A
TCGA.EW.A1P4.01A
TCGA.AR.A1AI.01A
TCGA.A8.A07C.01A
TCGA.A2.A04U.01A
EGFR
ERBB2+ERBB3+ERBB4
B
EGFR
ERBB2+ERBB3+ERBB4
C
p = 3.33e-10
EGFR / (ERBB2+ERBB3+ERBB4)
Cluster1
Cluster2
D
Relative EGFR expression level: high
TCGA.B6.A0I6.01A
TCGA.BH.A18V.06A
TCGA.EW.A1P4.01A
Relative EGFR expression level: low
TCGA.A8.A07C.01A
TCGA.A2.A04U.01A
TCGA.AR.A1AI.01A
ErbB1 ErbB3 ErbB4 ErbB2 Grb2 Shc RasGAP PI3K PTP1B SOS Gab1 RasGDP Raf MEK ERK PTEN Akt GSK3β
E
high (n=30)
low (n=30)
middle (n=169)
EGFR
ERBB2+ERBB3+ERBB4
F
MEK inhibitors
Selumetinib
PD-0325901
Relative viability (%)
Concentration (μM)
EGFR high
EGFR low
Activity area
n.s.
EGFR inhibitors
Erlotinib
Lapatinib
p = 8.45e-04
p = 1.32e-05

Figure 3.12: ErbB receptor expression ratios are critical for determining drug sensitivity

(**A** and **B**) Scatter plots of the epidermal growth factor receptor (EGFR) expression ratios to the sum of other ErbB receptor families: ERBB2, ERBB3, and ERBB4. Each dot represents one patient. (**A**) Purple and orange dots denote individual patients in clusters 1 and 2, respectively. (**B**) The corresponding response scores: the sum of the maximum level of three observables (pERK, pAkt, and pc-Myc) in response to HRG stimulation. Higher scores indicate a lower maximum level when stimulated with EGF. (**C**) Box plots showing the ErbB receptor expression ratios in patients from clusters 1 and 2. The p-value was calculated using the Brunner-Munzel test. (**D**) Sensitivity analysis of c-Myc activation on representative in silico TNBC patients with high (upper panel) and low (lower panel) EGFR expression ratios. (**E**) The EGFR expression ratios to the sum of other ErbB receptor families in the cell lines provided in the CCLE. Based on the ratio, cell lines are classified into three groups, namely, “high”: top 30, “low”: bottom 30, and “middle”: the other 169 cell lines. (**F**) Analysis of response profiles against anticancer drugs (MEK inhibitors: selumetinib, PD-0325901; EGFR inhibitors: erlotinib, lapatinib). The solid lines and shaded areas in dose-response curves denote the average and standard deviation of relative viability of 30 cell lines in each cluster, respectively. The efficacy and potency of a drug are simultaneously quantified based on the “activity area” and the *p*-values were calculated using the Brunner-Munzel test with a significance level of 0.05.

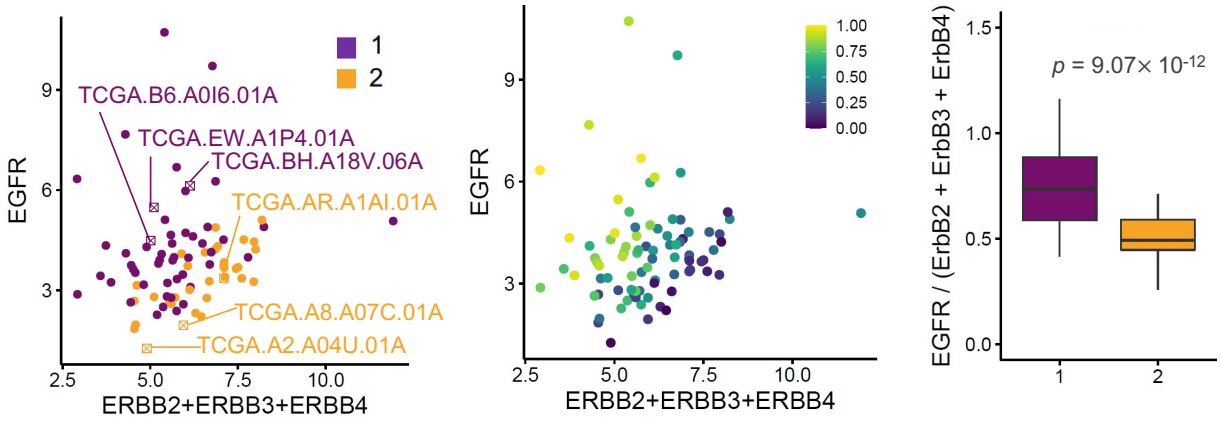


Figure 3.13: The ErbB receptor expression ratio: predicted protein level

(left, middle) Scatter plots from the predicted protein levels of epidermal growth factor receptor (EGFR) and sum of ERBB2, ERBB3, and ERBB4 from patients with the basal-like subtype in clusters 1 and 2. Colors represent clusters (left) and the response score: the sum of the maximum levels of three observables (pERK, pAkt, and pc-Myc) in response to HRG stimulation (middle). (right) Box plots showing the ErbB receptor expression ratio in patients from clusters 1 and 2. The *p*-value was calculated using the Brunner-Munzel test.

To test this hypothesis, I used the CCLE drug response data[30] for validation analysis. First, similar to patient-specific models, cell line-specific models were constructed using their gene expression values and classified based on their dynamic features (Figure 3.14). However, the available breast cancer cell line data (n = 2 for both clusters 1 and 2) were not enough to satisfy the statistical tests. Yet, breast cancer cell lines in cluster 1 (relative EGFR expression level: high) seemed less sensitive to EGFR inhibitors than those in cluster 2 (relative EGFR expression level: low).

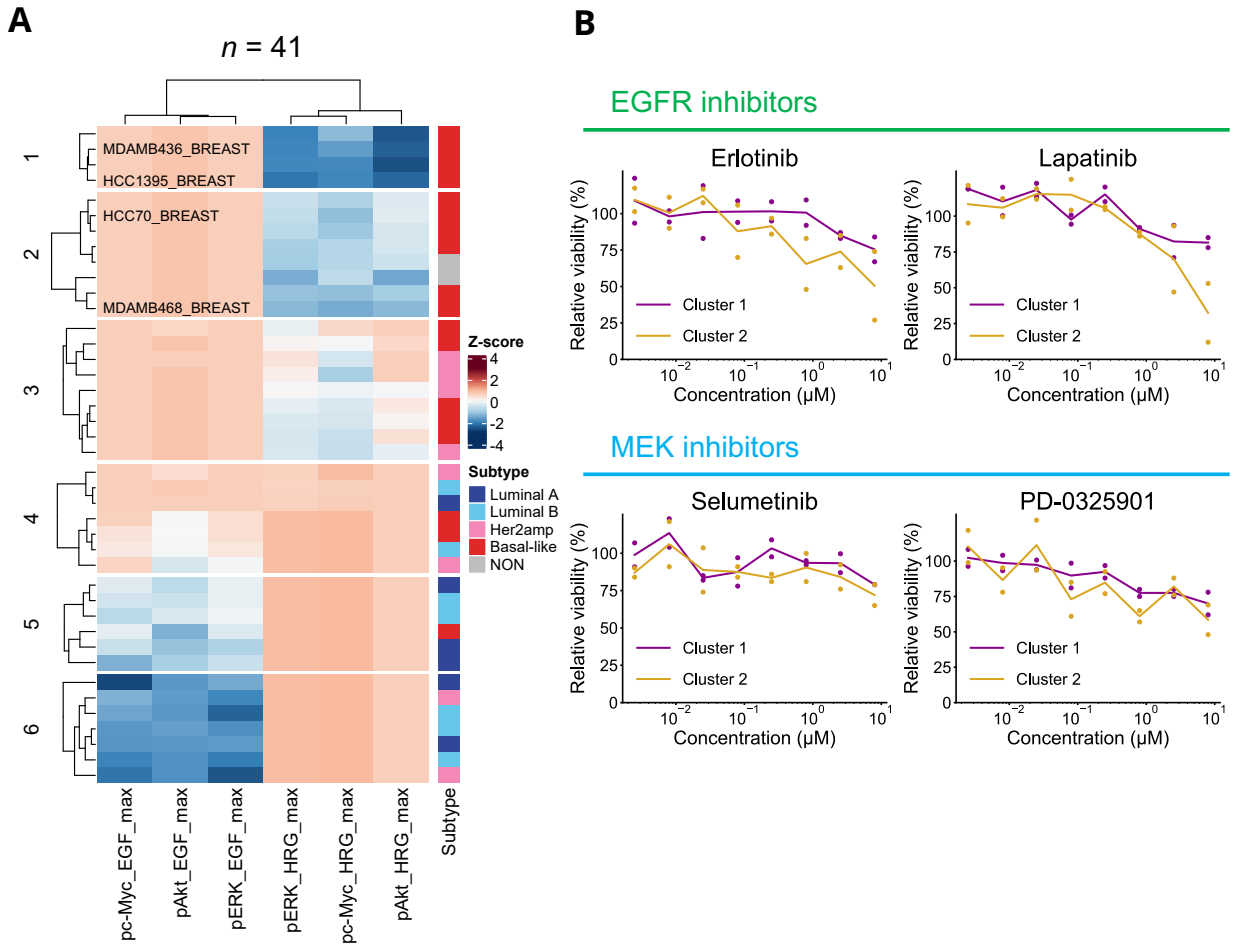


Figure 3.14: Classification using the breast cancer cell line-specific model

(**A**) Classifications of the 41 breast cancer cell lines provided in the Cancer Cell Line Encyclopedia (CCLE) through the mechanistic model. Each row corresponds to one breast cancer cell line. (**B**) Drug response analysis to the cell lines in cluster 1 (purple) and cluster 2 (orange). The available cell lines in the CCLE drug response data are MDAMB436 and HCC1395 (cluster 1), and HCC70 and MDAMB468 (cluster 2). Drug activity is characterized by the relative viability in response to EGFR inhibitors: erlotinib and lapatinib (top); and MEK inhibitors: selumetinib and PD-0325901 (bottom). Dots and solid lines denote individual relative viabilities and median values, respectively.

To further validate this, I collected all types of cancer cell lines (*n* = 229) from the CCLE, classified them in terms of ErbB receptor expression ratio, and analyzed drug sensitivity (Figure 3.12D). Drug efficacy and potency were quantified by the "activity area," which was the area over the dose-response curve[30]. I found that cell lines with higher EGFR expression levels showed significantly lower sensitivity to EGFR inhibitors (erlotinib and lapatinib). There was no statistical significance regarding other inhibitors, such as MEK inhibitors (selumetinib and PD-0325901) (Figure 3.12E and F).

3.8 Applying model-based stratification to other types of cancer

Next, I tried to find out if the same ErbB network model could stratify patients with different types of cancer. I selected colon cancer[37] because EGFR inhibitors are used to treat it[83]. After individualization of the ErbB network models by adding 189 individual transcriptomic datasets provided in TCGA database (TCGA-COAD), the models successfully classified their prognoses according to the c-Myc argmax (the time at which the signal intensity reached the maximum) (Figure 3.15A-C). Patients in cluster 4 showed poorer prognoses than those in other clusters, even though their signaling dynamics were similar to those in cluster 3. To predict the mechanistic cause of the difference between clusters 3 and 4, I performed a sensitivity analysis on a time-integrated c-Myc response. The results of this analysis implied that *in silico* patients in cluster 4 showed lower EGFR (ErbB1) sensitivity than those in cluster 3 (Figure 3.15D), indicating that patients in cluster 4 may be more resistant to anti-EGFR treatments. In fact, a recent study found a significant correlation between c-Myc expression and anti-EGFR antibody resistance in metastatic colorectal cancer[84]. Thus, my mathematical analysis potentially provides a mechanistic insight explaining individual patients' responses to anti-EGFR therapy.

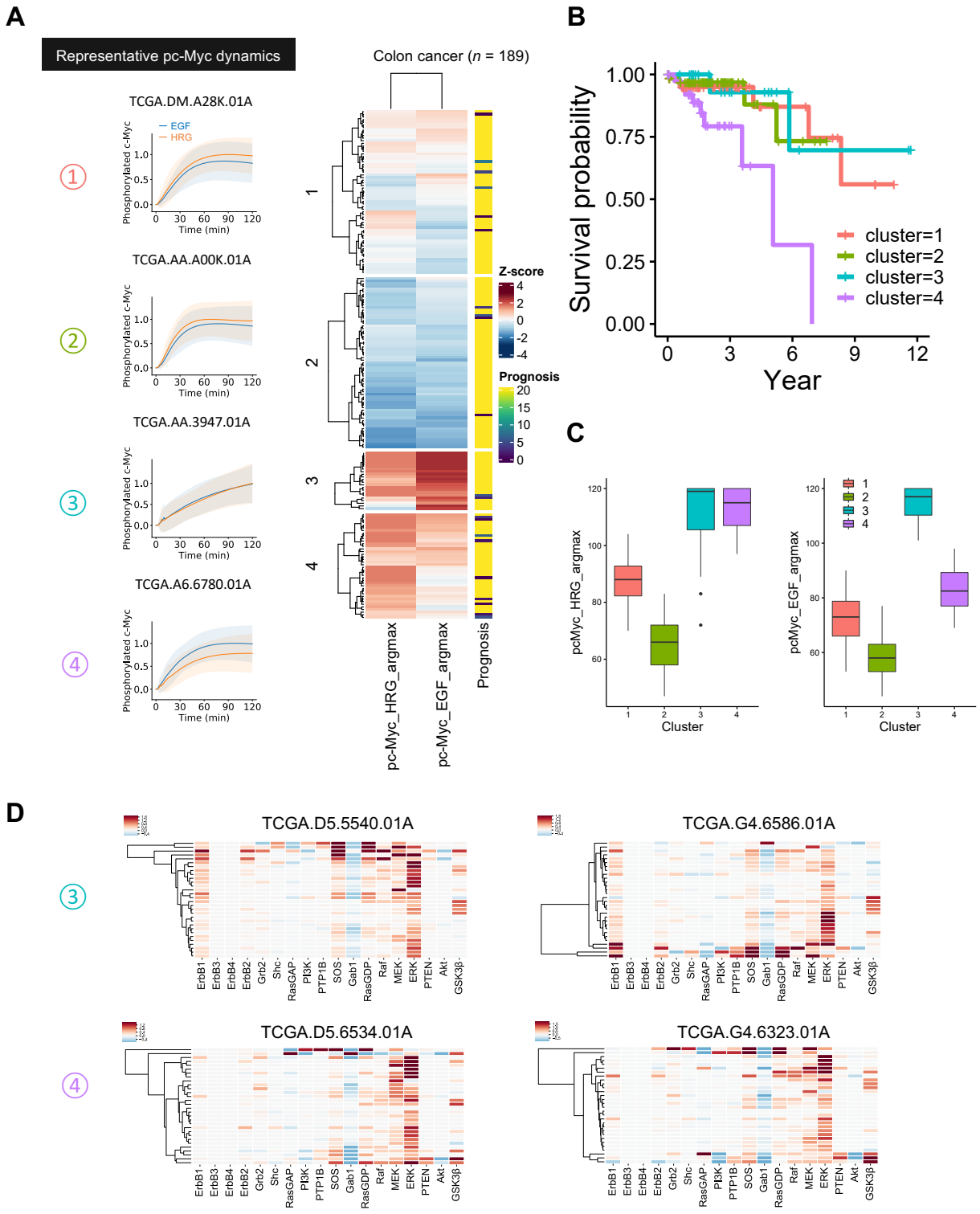


Figure 3.15: Applying model-based patient stratification to colon cancer

(**A**) A total of 189 patients were classified based on personalized simulations. The representative pc-Myc dynamics were extracted from the topmost portion of each cluster. The blue and orange solid lines denote simulations with EGF and HRG stimulation, respectively. Shaded areas denote SD. The metric ‘‘argmax’’ denotes the time at which the simulated signal intensity reached the maximum. (**B**) Kaplan-Meier survival curves of all patients for all clusters. (**C**) Boxplots showing individual argmax values in each cluster. (**D**) Sensitivity analysis on the time-integrated response of EGF-induced pc-Myc for randomly sampled in silico patients in clusters 3 and 4.

4 Discussion

4.1 Summary and Conclusion

Identifying prognostic factors and potential therapeutic drugs for individual patients is crucial for the development of personalized medicine. Recent studies indicate that the temporal dynamics of signaling activities and transcription factors are critical for cell fate determination[85–91]. Fey et al. also found that the c-Jun N-terminal kinase (JNK) signaling metrics can be used to predict the prognosis of neuroblastoma[92].

Therefore, I hypothesized that the signaling dynamics in individual cancer patients could be used for the classification and prediction of prognosis and drug responses. To this end, I developed a scalable computational framework, Pasmopy, for patient-specific modeling and classification of cancers based on *in silico* signaling dynamics[93]. Besides developing these unique classification features, I used this framework to construct mechanistic models from texts describing biochemical reactions instead of from mathematical equations. In this study, I developed a model for the ErbB receptor signaling network from the text as a proof of concept[94]. This method of building models will facilitate future studies investigating the underlying mechanisms of various biological processes.

Notably, I found that the selected gene panels used in the current ErbB model (Table 3.1) were not capable of classifying TNBC prognosis (Figure 4.1), and were not sufficient to identify the molecular mechanism (e.g., EGFR/ErbBs ratio) or drug response. EGFR overexpression has been reported in up to 78 % of patients with TNBC[95]. Consistently, I randomly sampled "*in silico* patients" and performed sensitivity analyses,

which suggested that patients with higher EGFR expression ratios were less sensitive to EGFR inhibition. To test this model-based prediction, I used publicly available cell line data and confirmed that cancer cell lines with higher EGFR expression ratios were less sensitive to anticancer drugs targeting EGFR, such as erlotinib and lapatinib. Thus, this framework not only allows the classification of patients but also provides potential mechanistic insights into the regulation of signaling pathways and drug resistance.

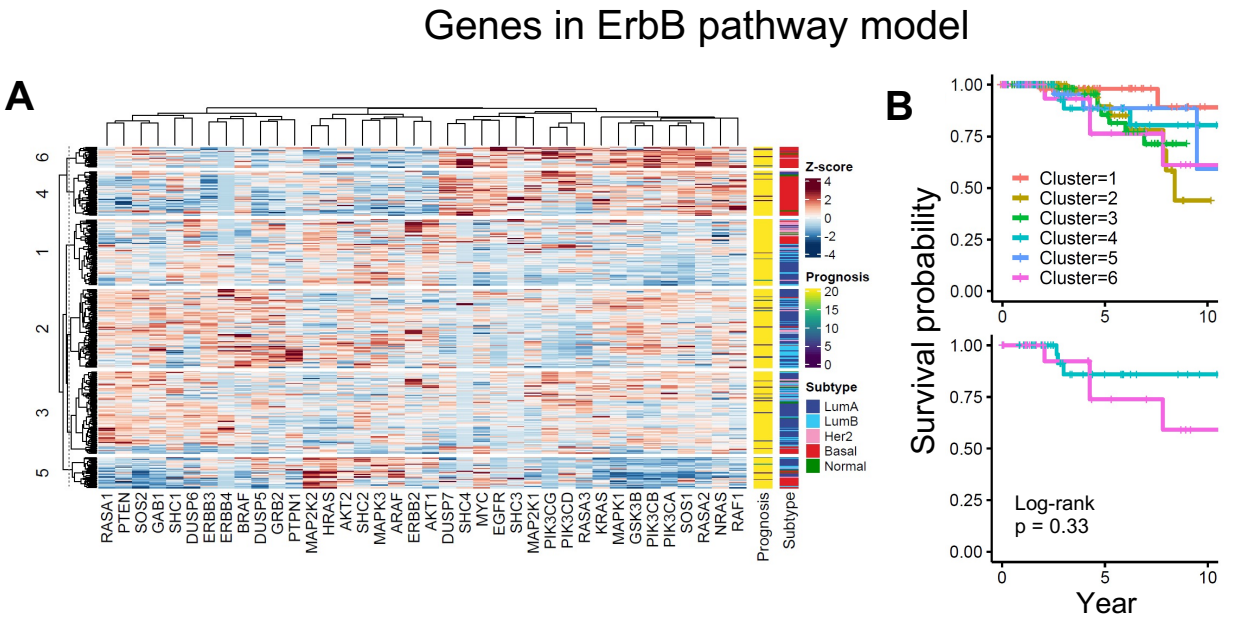


Figure 4.1: Clustering of patients based on gene expression levels of ErbB pathway model

(**A**) Clustering of patients based on gene set expression levels in the ErbB pathway model. mRNA levels (log2(TPM+1)) for each gene were normalized to the Z-score; red indicates higher expression and blue indicates lower expression. (**B**) (top) Kaplan-Meier survival curves for patients in each cluster; (bottom) Kaplan-Meier survival curves of patients with the basal-like subtype in clusters 4 and 6 in which more than half of patients consist of the basal-like subtype. The *p*-value was obtained using log-rank test.

Another advantage of my method is that it enables computational analysis with a small number of data inputs. In this study, the model parameter was trained against experimental data consisting of four cell lines, three observables, two growth factors, and eight time points. The number of training datasets was much smaller than the one used in an earlier work[29], in which datasets from 120 cell lines treated with 7 different drugs at up to 9 concentrations each were used to predict anticancer drug response in different cell lines. I confirmed that the model-predicted sensitive reactions in the ErbB network

were highly conserved across 30 independent parameter sets. This indicates that the parameter identifiability obtained from my modeling approach does not significantly affect the uncertainty of the model output.

The models were personalized to each patient by incorporating individual gene expression data. Although similar approaches have been used in previous studies for JNK signaling[92] and HGF/Met signaling pathway to stratify neuroblastoma patients[96], they needed to rescale protein levels based on their mRNA levels fold change between tumor and healthy tissue. However, this scaling method constricts the potential of mRNA information. I extended this method and used transcriptome data to infer the maximal translation rate of the corresponding proteins. This method allows the development of larger models from genome-wide transcriptome datasets.

Finally, I hypothesized that the critical molecular mechanism governing cancer prognosis might be shared, at least in part, by different types of cancers. This would explain why the same model (i.e., short-term ErbB signaling dynamics within 120 min) can be used to classify prognosis in both breast and colon cancer patients. Therefore, my study suggests that modeling approaches can be used to investigate the specificity and commonality of different types of cancer and to evaluate drug repositioning (Figure 4.2).

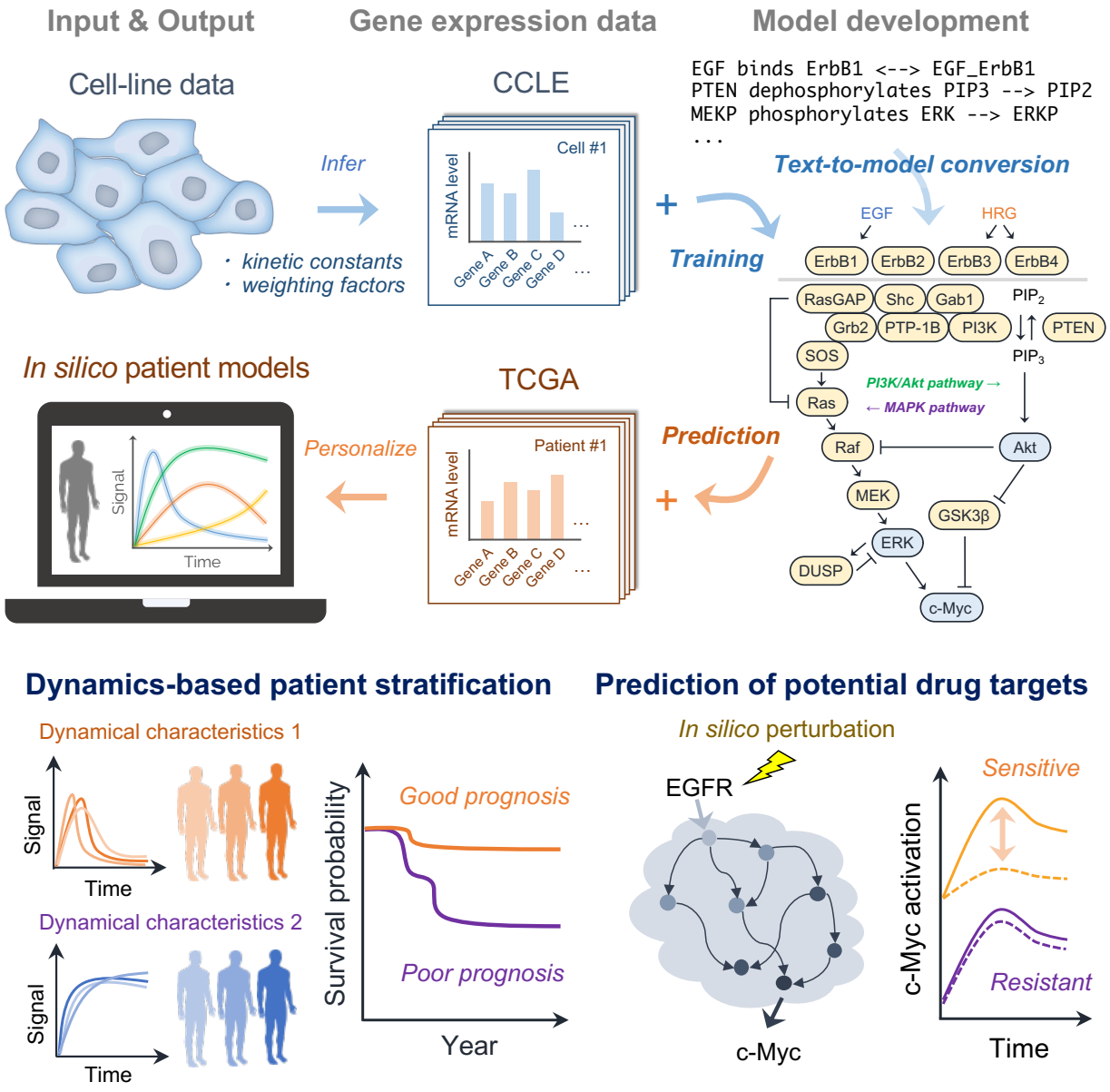


Figure 4.2: Graphical summary of this study. Adapted from Imoto et al. *iScience* (2022)

4.2 Future perspectives

This study focuses on the classification of patients with cancer-based on the dynamics of ErbB receptor signaling pathways. However, the possibility of other signaling pathways being involved cannot be ignored. My framework will be able to address these issues by expanding the network to include other receptors and players in cell cycle regulation, apoptotic pathways, or metabolic pathways. In the current study, I used TCGA and CCLE datasets after the normalization of the TPM value. Therefore, users need to reconsider their data normalization approach when other methods such as microarrays or qRT-PCR are used.

In this study, I did not consider somatic mutations. To support this, there was no clear trend involving mutation types, at least for two TNBC clusters (Figure 4.3). However, gene mutations are tightly linked to treatment strategies in some types of cancer, such as lung cancer[97]. Therefore, in the future, parameterization of mutational information might be crucial to being able to adapt the model to a wide range of cancer types.

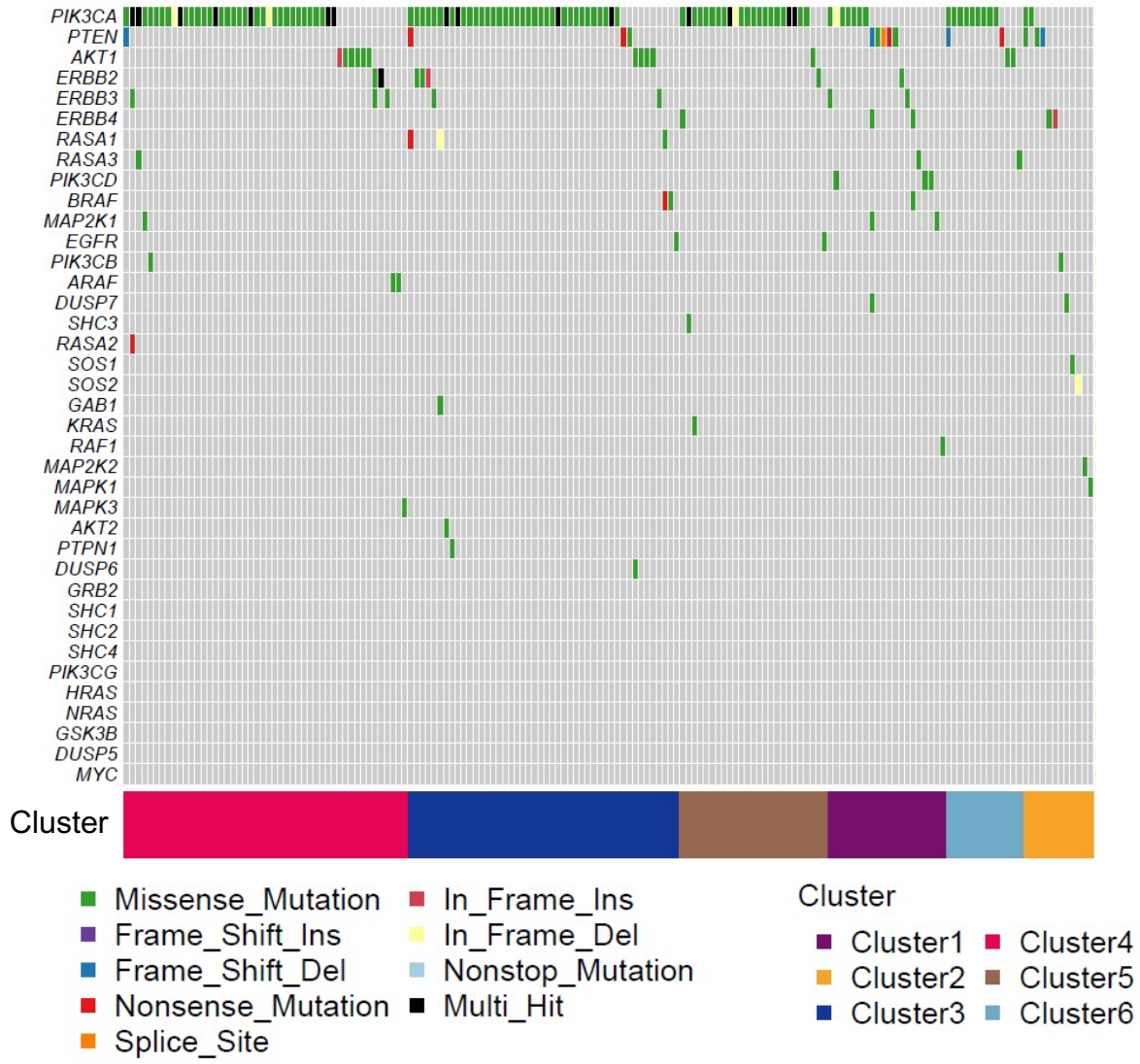


Figure 4.3: Onco-plot of somatic single nucleotide variants (SNVs)

Brick plot showing somatic mutations of the gene panels in the current ErbB pathway model, grouped according to the cluster in Figure 3. Each horizontal line describes a single gene, vertical lines represent different samples, and colors denote the type of mutations. In this analysis, we used data from Somatic Variant Calling obtained with the Mutect2 (Cibulskis et al., 2013) workflow.

The computational approach presented in this study enables us to predict potential drug targets in individual patients from their gene expression profiles. The next key challenge will be to find ways to inhibit the predicted target. For example, recent studies have reported that, in the MAPK pathway, treating cancer with a single type of RAF inhibitor can trigger the contradictory activation of this pathway[98], which can even accelerate tumor growth[99]. Boris Kholodenko demonstrated in 2015 that fundamental thermodynamic principles governing allosteric inhibitor effects can explain both paradoxical RAF kinase activation and common resistance mechanisms to RAF inhibitors[100]. Several years later, rule-based models of MAPK signaling based on this finding showed that the combination of structurally different RAF inhibitors can effectively inhibit the pathway activity[101,102]. Therefore, to pave the way for personalized treatments, next-generation patient-specific models will need to incorporate structural, thermodynamic, kinetic, and cellular data.

Nowadays, it has become much easier to access large-scale, comprehensive datasets. In addition to the increasing availability of data, remarkable advances in statistical and machine-learning approaches have enabled us to predict biological functions from big datasets. Recently, we have witnessed notable achievements in the field of structural biology, specifically AlphaFold2[103,104] and RoseTTAFold[105,106], which can predict three-dimensional protein structure from protein sequences with astonishingly high accuracy. Although transformative, such machine learning and statistical models have shortcomings. Most notably, they often fail to explain predicted outcomes with detailed mechanistic reasoning[107–109]. On the other hand, mechanistic models, as shown

in this study, provide us with an interpretable explanation for underlying mechanisms because they explicitly describe biochemical events in a system of interest.

A mechanistic understanding of complex biological systems is urgently needed to tackle current problems in biology, pharmacology, and medicine. Therefore, I anticipate that the method presented in this study will contribute to next-generation modeling approaches that investigate dysfunction mechanisms in individual patients and improve healthcare.

5 Appendix

5.1 Modeling biological systems using *BioMASS*

BioMASS is a computational framework for modeling and analyzing biological signaling systems in Python. It provides useful tools for text-to-model conversion, numerical simulation, parameter estimation, network analysis, and result visualization. BioMASS requires Python3.8 or later to run. Install BioMASS from PyPI using:

```
$ pip install biomass
```

In the following examples, you will be briefly guided through the basics of how to use this software, and I will use biomass version 0.10.0 (https://pypi.org/project/biomass/0.10.0/).

5.1.1 Code availability

I strongly believe that open source is essential for pushing interdisciplinary science forward because it contributes to the development of technology and eventually solves real-world problems. Everyone can freely access this software on GitHub.

- Source code: https://github.com/biomass-dev/biomass
- Online documentation: https://biomass-core.readthedocs.io
    - Installation guide: https://biomass-core.readthedocs.io/en/latest/installation.html
    - Tutorial: https://biomass-core.readthedocs.io/en/latest/tutorial/cfos.html

5.2 Model preparation

I will use a mechanistic model of the c-Fos expression network dynamics[15] to show how to build computational models, estimate parameter values from experimental data, and identify sensitive components in complex biochemical systems. Please install:

- `biomass==0.10.0` for simulation, parameterization, and analysis of the model
- `tqdm` for visualizing progress bars

5.2.1 Text-to-model conversion

In this framework, each model is managed in a folder that includes several Python modules. A brief description of each file/folder is below (Table 5.1):

Table 5.1: Python modules required for building BioMASS models

Name	Content
`name2idx/`	Names of model parameters and species
`reaction_network.py`	Flux vector and reaction indices grouped according to biological processes
`ode.py`	Differential equation, parameters and initial condition
`observable.py`	Observables, simulations and experimental data
`search_param.py`	Lower and upper bounds of model parameters to be estimated

Name	Content
`problem.py`	An objective function to be minimized, i.e., the distance between model simulation and experimental data
`viz.py`	Plotting parameters for customizing figure properties

Text2Model lets you create a BioMASS model from the text. You simply describe biochemical reactions, and the molecular mechanisms extracted from the text are converted into an executable model. Below is an example of using Text2Model to build a simple Michaelis-Menten two-step enzyme catalysis model:

$$E + S \rightleftarrows ES \rightarrow E + P$$

An enzyme, E, binds to a substrate, S, to form a complex, ES, which in turn releases a product, P, regenerating the original enzyme.

Prepare a text file describing biochemical reactions (e.g., michaelis_menten.txt)

```
E + S <--> ES | kf=0.003, kr=0.001 | E=100, S=50
ES --> E + P | kf=0.002, kr=0

@obs Substrate: u[S]
@obs E_free: u[E]
@obs E_total: u[E] + u[ES]
@obs Product: u[P]
@obs Complex: u[ES]

@sim tspan: [0, 100]
```

The number at the beginning of each line denotes the line number, therefore, you do not need to write it. The text file you need to prepare can be divided into three parts: the reaction layer, the observable layer, and the simulation layer. For more details, please visit the online documentation: https://pasmopy.readthedocs.io/en/latest/model_development.html

Reaction layer

In the reaction layer, you need to describe biochemical reactions. Each reaction described in line number i will be converted into an i^{th} rate equation. To specify parameters or initial conditions, you can put that information after `|`.

You can impose parameter constraints by specifying the line number in the parameter section.

```
1 # Nucleocytoplasmic Shuttling of DUSP
2 DUSPc translocates from the cytoplasm to the nucleus <--> DUSPn
3 pDUSPc translocates from the cytoplasm to the nucleus <--> pDUSPn |2|
```

In the example above, you can assume that import and export rates were identical for DUSP (line 2) and pDUSP (line 3).

If the amount of a model species should be held fixed (never consumed) during simulation, you can add `fixed` prefix:

```
# [Ligand] will be held fixed to 10.0 during simulation
Ligand binds Receptor <--> LR | kf = 1e-6, kr = 1e-1 | fixed Ligand = 10.0
```

To describe more complex rate equations, you can use `@rxn` prefix:

```
@rxn Reactant --> Product: define rate equation here
```

You can also supply your terminology in a reaction rule via:

```
from pasmopy import Text2Model

# Supply "releases" to the reaction rule: "dissociate"
mm_kinetics = Text2Model("michaelis_menten.txt")
mm_kinetics.register_word({"dissociate": ["releases"]})
# Now you can use "releases" in your text, e.g., 'ES releases E and P'
mm_kinetics.convert()
```

Observable layer (prefix: @obs)

In the observable layer, you need to specify `biomass.observables`, which can be correlated between model simulations and experimental measurements. You can create an observable by using model parameters (p) and species (u).

Simulation layer (prefix: @sim)

In the simulation layer, you can set simulation conditions, e.g., the simulation time, and the initial concentration of model species

After preparing the text file, you will need to convert it into an executable model:

```
>>> from pasmopy import Text2Model
>>> description = Text2Model("michaelis_menten.txt")
>>> description.convert()
Model information
-----------------
2 reactions
4 species
4 parameters
```

```
>>> from pasmopy import create_model, run_simulation
>>> model = create_model("michaelis_menten")
>>> run_simulation(model)
```

Then you will get simulation results in the figure/ directory in the model folder (Figure 5.1).

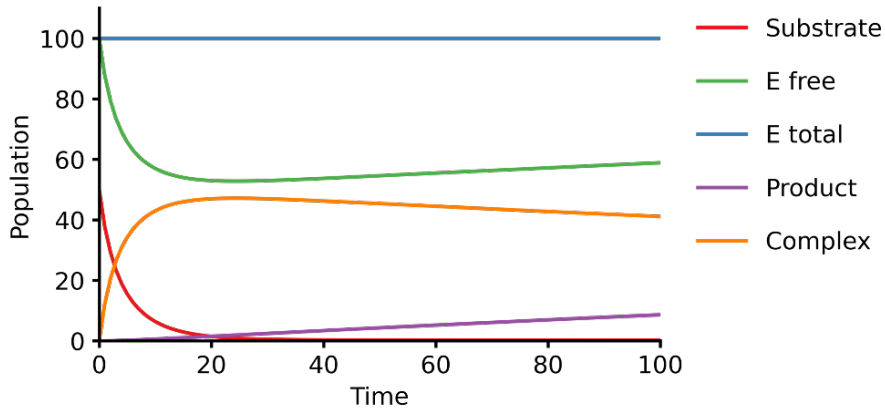


Figure 5.1: Simulation result of Michaelis-Menten two-step enzyme catalysis model

Next, Below is an example of Text2Model in action to illustrate the EGF signaling pathway[7].

```
 1 EGF + EGFR <--> Ra | kf=0.003, kr=0.06 | EGFR=100
 2 Ra + Ra <--> R2 | kf=0.01, kr=0.1
 3 R2 <--> RP | kf=1, kr=0.01
 4 RP is dephosphorylated --> R2 | V=450, K=50
 5 RP + PLCg <--> RPL | kf=0.06, kr=0.2 | PLCg=105
 6 RPL is phosphorylated <--> RPLP | kf=1, kr=0.05
 7 RPLP <--> RP + PLCgP | kf=0.3, kr=0.006
 8 PLCgP is dephosphorylated --> PLCg | V=1, K=100
 9 RP + Grb2 <--> RG | kf=0.003, kr=0.05 | Grb2=85
10 RG + SOS <--> RGS | kf=0.01, kr=0.06 | SOS=34
11 GS <--> RP + GS | kf=0.03, kr=4.5e-3
12 GS <--> Grb2 + SOS | kf=1.5e-3, kr=1e-4
13 RP + Shc <--> RSh | kf=0.09, kr=0.6 | Shc=150
```

```
RSh <--> RShP | kf=6, kr=0.06
RShP <--> ShP + RP | kf=0.3, kr=9e-4
ShP is dephosphorylated --> Shc | V=1.7, K=340
RShP + Grb2 <--> RShG | kf=0.003, kr=0.1
RShG <--> RP + ShG | kf=0.3, kr=9e-4
RShG + SOS <--> RShGS | kf=0.01, kr=2.14e-2
RShGS <--> ShGS + RP | kf=0.12, kr=2.4e-4
ShP + Grb2 <--> ShG | kf=0.003, kr=0.1
ShG + SOS <--> ShGS | kf=0.03, kr=0.064
ShGS <--> ShP + GS | kf=0.1, kr=0.021
RShP + GS <--> RShGS | kf=0.009, kr=4.29e-2
PLCgP <--> PLCgP_I | kf=1, kr=0.03

@obs Total_phosphorylated_Shc: u[RShP] + u[RShG] + u[RShGS] + u[ShP] + u[ShG] + u[ShGS]
@obs Total_Grb2_coprecipitated_with_Shc: u[RShG] + u[ShG] + u[RShGS] + u[ShGS]
@obs Total_phosphorylated_Shc_bound_to_EGFR: u[RShP] + u[RShG] + u[RShGS]
@obs Total_Grb2_bound_to_EGFR: u[RG] + u[RGS] + u[RShG] + u[RShGS]
@obs Total_SOS_bound_to_EGFR: u[RGS] + u[RShGS]
@obs ShGS_complex: u[ShGS]
@obs Total_phosphorylated_PLCg: u[RPLP] + u[PLCgP]

@sim tspan: [0, 120]
@sim condition EGF20nM: init[EGF] = 680
@sim condition EGF2nM: init[EGF] = 68
@sim condition Absence_PLCgP_transloc: init[EGF] = 680; p[kf25] = 0; p[kr25] = 0
```

When converting the text into an executable model, you can also display thermodynamic restrictions by setting `show_restrictions` to True.

```
>>> from pasmopy import Text2Model
>>> description = Text2Model("Kholodenko_JBC_1999.txt")
>>> description.convert(show_restrictions=True)
Model information
-----------------
25 reactions
23 species
50 parameters

Thermodynamic restrictions
--------------------------
```

```
{9, 12, 10, 11}
{15, 18, 21, 17}
{18, 22, 20, 19}
{17, 24, 12, 19}
{23, 24, 20, 15}
{23, 12, 22, 21}
```

The output of *thermodynamic restrictions* above shows the cyclic pathways in the biochemical reaction network. These detailed balance relations require the product of the equilibrium constants along a cycle to be equal to 1, since at equilibrium the net flux through any cycle vanishes. To make kinetic models thermodynamically feasible[110], these constraints must be satisfied.

Now I will move on to building the c-Fos model. By comparing the reaction scheme in the original paper (Fig. 1E) and the description below, you can learn how to build computational models via Text2Model.

```
 1 @rxn ERKc --> pERKc: p[V1] * p[a] * u[ppMEKc] * u[ERKc] /  ( p[K1] * (1 + u[pERKc] / p[K2]) + u[ERKc] )
|| ERKc=9.60e02
 2 @rxn pERKc --> ppERKc: p[V2] * p[a] * u[ppMEKc] * u[pERKc] /  ( p[K2] * (1 + u[ERKc] / p[K1]) +
u[pERKc] ) | const V2=2.20e-01, const K2=3.50e02
 3 @rxn pERKc --> ERKc: p[V3] * u[pERKc] /  ( p[K3] * (1 + u[ppERKc] / p[K4]) + u[pERKc] ) | const
V3=7.20e-01, const K3=1.60e02
 4 @rxn ppERKc --> pERKc: p[V4] * u[ppERKc] /  ( p[K4]* (1 + u[pERKc] / p[K3]) + u[ppERKc] ) | const
V4=6.48e-01, const K4=6.00e01
 5 @rxn pERKn --> ERKn: p[V5] * u[pERKn] /  ( p[K5] * (1 + u[ppERKn] / p[K6]) + u[pERKn] )
 6 @rxn ppERKn --> pERKn: p[V6] * u[ppERKn] /  ( p[K6] * (1 + u[pERKn] / p[K5]) + u[ppERKn] ) |5|
 7 ERKc translocates to nucleus (0.94, 0.22) <--> ERKn | const kf=1.20e-02, const kr=1.80e-02
 8 pERKc translocates to nucleus (0.94, 0.22) <--> pERKn | const kf=1.20e-02, const kr=1.80e-02
 9 ppERKc translocates to nucleus (0.94, 0.22) <--> ppERKn | const kf=1.10e-02, const kr=1.30e-02
10 ppERKn transcribes PreduspmRNAn
11 PreduspmRNAn translocates to cytoplasm --> duspmRNAc
12 duspmRNAc is degraded
13 duspmRNAc is translated into DUSPc
14 ppERKc phosphorylates DUSPc --> pDUSPc
15 pDUSPc is dephosphorylated --> DUSPc
16 DUSPc is degraded | const kf=2.57e-04
17 pDUSPc is degraded | const kf=9.63e-05
```

```
DUSPc translocates to nucleus (0.94, 0.22) <--> DUSPn  # assuming cytoplasmic and nuclear volume to 0.94 pl and 0.22 pl
pDUSPc translocates to nucleus (0.94, 0.22) <--> pDUSPn |18|
ppERKn phosphorylates DUSPn --> pDUSPn
pDUSPn is dephosphorylated --> DUSPn
DUSPn is degraded | const kf=2.57e-04
pDUSPn is degraded | const kf=9.63e-05
ppERKc phosphorylates RSKc --> pRSKc || RSKc=3.53e02
pRSKc is dephosphorylated --> RSKc
pRSKc translocates to nucleus (0.94, 0.22) <--> pRSKn
pRSKn phosphorylates CREBn --> pCREBn || CREBn=1.00e03
pCREBn is dephosphorylated --> CREBn
ppERKn phosphorylates Elk1n --> pElk1n || Elk1n=1.51e03
pElk1n is dephosphorylated --> Elk1n
pCREBn & pElk1n transcribes PrecfosmRNAn, repressed by Fn
PrecfosmRNAn translocates to cytoplasm --> cfosmRNAc
cfosmRNAc is degraded
cfosmRNAc is translated into cFOSc
ppERKc phosphorylates cFOSc --> pcFOSc
pRSKc phosphorylates cFOSc --> pcFOSc
pcFOSc is dephosphorylated --> cFOSc
cFOSc is degraded | const kf=2.57e-04
pcFOSc is degraded | const kf=9.63e-05
cFOSc translocates to nucleus (0.94, 0.22) <--> cFOSn
pcFOSc translocates to nucleus (0.94, 0.22) <--> pcFOSn |40|
ppERKn phosphorylates cFOSn --> pcFOSn
pRSKn phosphorylates cFOSn --> pcFOSn
pcFOSn is dephosphorylated --> cFOSn
cFOSn is degraded | const kf=2.57e-04
pcFOSn is degraded | const kf=9.63e-05
DUSPn + ppERKn <--> DUSPn_ppERKn
DUSPn_ppERKn --> DUSPn + pERKn
DUSPn + pERKn <--> DUSPn_pERKn
DUSPn_pERKn --> DUSPn + ERKn
DUSPn + ERKn <--> DUSPn_ERKn
pDUSPn + ppERKn <--> pDUSPn_ppERKn |47|
pDUSPn_ppERKn --> pDUSPn + pERKn |48|
pDUSPn + pERKn <--> pDUSPn_pERKn |49|
pDUSPn_pERKn --> pDUSPn + ERKn |50|
pDUSPn + ERKn <--> pDUSPn_ERKn |51|
pcFOSn transcribes PreFmRNAn
PreFmRNAn translocates to cytoplasm --> FmRNAc
```

```
FmRNAc is degraded
FmRNAc is translated into Fc
Fc is degraded
Fc translocates to nucleus (0.94, 0.22) <--> Fn
Fn is degraded

@add species ppMEKc
@add param Ligand

@obs Phosphorylated_MEKc: u[ppMEKc]
@obs Phosphorylated_ERKc: u[pERKc] + u[ppERKc]
@obs Phosphorylated_RSKw: u[pRSKc] + u[pRSKn] * (0.22 / 0.94)
@obs Phosphorylated_CREBw: u[pCREBn] * (0.22 / 0.94)
@obs dusp_mRNA: u[duspmRNAc]
@obs cfos_mRNA: u[cfosmRNAc]
@obs cFos_Protein: (u[pcFOSn] + u[cFOSn]) * (0.22 / 0.94) + u[cFOSc] + u[pcFOSc]
@obs Phosphorylated_cFos: u[pcFOSn] * (0.22 / 0.94) + u[pcFOSc]

@sim tspan: [0, 5400]
@sim unperturbed: p[Ligand] = 0
@sim condition EGF: p[Ligand] = 1
@sim condition HRG: p[Ligand] = 2
```

For more details about available reaction rules, please see https://biomass-core.readthedocs.io/en/latest/api/reaction_rules.html.

5.2.2 Set the input of the model

The input for the mechanistic c-Fos model is given by an interpolation function of the ppMEK experimental data. Open `ode.py`.

```python
class DifferentialEquation(ReactionNetwork):

    def __init__(self, perturbation):
        super(DifferentialEquation, self).__init__()
        self.perturbation = perturbation

    @staticmethod
```

```
    def _get_ppMEK_slope(t, ligand) -> float:
        assert ligand in ['EGF', 'HRG']
        timepoints = [0, 300, 600, 900, 1800, 2700, 3600, 5400]
        ppMEK_data = {
            'EGF': [0.000, 0.773, 0.439, 0.252, 0.130, 0.087, 0.080, 0.066],
            'HRG': [0.000, 0.865, 1.000, 0.837, 0.884, 0.920, 0.875, 0.789],
        }
        assert len(timepoints) == len(ppMEK_data[ligand])
        slope = [
            (ppMEK_data[ligand][i + 1] - activity) / (timepoints[i + 1] - timepoint)
            for i, (timepoint, activity) in enumerate(zip(timepoints, ppMEK_data[ligand]))
            if i + 1 < len(timepoints)
        ]
        for i, timepoint in enumerate(timepoints):
            if timepoint <= t <= timepoints[i + 1]:
                return slope[i]
        assert False

    # Refined Model
    def diffeq(self, t, y, *x):

        v = self.flux(t, y, x)

        if self.perturbation:
            for i, dv in self.perturbation.items():
                v[i] = v[i] * dv

        dydt = [0] * V.NUM

        if x[C.Ligand] == 1:  # EGF=10nM
            dydt[V.ppMEKc] = self._get_ppMEK_slope(t, 'EGF')
        elif x[C.Ligand] == 2:  # HRG=10nM
            dydt[V.ppMEKc] = self._get_ppMEK_slope(t, 'HRG')
        else:  # Default: No ligand input
            dydt[V.ppMEKc] = 0.0

    ...
```

5.2.3 Normalize simulation results

Experimental data were normalized by dividing them by the maximum value of the responses. To correlate model simulation results with experimental measurements, I will need to normalize the simulation results. Open `observable.py`.

```
class Observable(DifferentialEquation):

    ...

    self.normalization: dict = {}
    for observable in self.obs_names:
        self.normalization[observable] = {"timepoint": None, "condition": []}
```

Here, you can define how you would like to normalize simulation results for each observable. The `normalization[observable]` dictionary accepts two keys, *'timepoint'* and *'condition'*.

- *'timepoint'*: The time point at which simulated values are normalized. If `None`, the maximum value will be used for normalization.
- *'condition'*: The experimental conditions to use for normalization. If empty, all conditions defined in `self.conditions` will be used.

5.2.4 Choose an ODE solver to use

Most systems biology models are non-linear and closed-form solutions are not available. Accordingly, numerical integration methods have to be employed to study them[68]. Open `observable.py` and choose the integration method in `get_steady_state()` and `solve_ode()`.

```
class Observable(DifferentialEquation):

    ...

    def simulate(self, x, y0, _perturbation=None):

        ...

        x[C.Ligand] = 0
        y0 = get_steady_state(self.diffeq, y0, tuple(x), integrator='vode')
        if not y0:
            return False

        for i, condition in enumerate(self.conditions):
            if condition == "EGF":
                x[C.Ligand] = 1
            elif condition == "HRG":
                x[C.Ligand] = 2

            sol = solve_ode(self.diffeq, y0, self.t, tuple(x), method="BDF")

            ...
```

`get_steady_state` runs a model simulation till a steady state for that parameter set. First, I simulate the model with no ligand until the system reaches a steady state, take the final state of the equilibration simulation, and use it as the initial state of the new simulation. By default, LSODA[111] is applied for both integrators.

5.2.5 Set experimental data for parameterization of the model

Open `observable.py`.

```
class Observable(DifferentialEquation):

    ...

    def set_data(self):
```

```
    """
    Notes
    -----
    self.experiments: list of dict
        Time-series experimetal measurements.
    self.error_bars: list of dict
        Error bars shown in figures (e.g., SD or SE).
    """

    self.experiments[self.obs_names.index("Phosphorylated_MEKc")] = {
        "EGF": [0.000, 0.773, 0.439, 0.252, 0.130, 0.087, 0.080, 0.066],
        "HRG": [0.000, 0.865, 1.000, 0.837, 0.884, 0.920, 0.875, 0.789],
    }
    self.error_bars[self.obs_names.index("Phosphorylated_MEKc")] = {
        "EGF": [
            sd / np.sqrt(3) for sd in [0.000, 0.030, 0.048, 0.009, 0.009, 0.017, 0.012, 0.008]
        ],
        "HRG": [
            sd / np.sqrt(3) for sd in [0.000, 0.041, 0.000, 0.051, 0.058, 0.097, 0.157, 0.136]
        ],
    }

    self.experiments[self.obs_names.index("Phosphorylated_ERKc")] = {
        "EGF": [0.000, 0.867, 0.799, 0.494, 0.313, 0.266, 0.200, 0.194],
        "HRG": [0.000, 0.848, 1.000, 0.971, 0.950, 0.812, 0.747, 0.595],
    }
    self.error_bars[self.obs_names.index("Phosphorylated_ERKc")] = {
        "EGF": [
            sd / np.sqrt(3) for sd in [0.000, 0.137, 0.188, 0.126, 0.096, 0.087, 0.056, 0.012]
        ],
        "HRG": [
            sd / np.sqrt(3) for sd in [0.000, 0.120, 0.000, 0.037, 0.088, 0.019, 0.093, 0.075]
        ],
    }

    self.experiments[self.obs_names.index("Phosphorylated_RSKw")] = {
        "EGF": [0, 0.814, 0.812, 0.450, 0.151, 0.059, 0.038, 0.030],
        "HRG": [0, 0.953, 1.000, 0.844, 0.935, 0.868, 0.779, 0.558],
    }
    self.error_bars[self.obs_names.index("Phosphorylated_RSKw")] = {
        "EGF": [
            sd / np.sqrt(3) for sd in [0, 0.064, 0.194, 0.030, 0.027, 0.031, 0.043, 0.051]
```

```
            ],
            "HRG": [
                sd / np.sqrt(3) for sd in [0, 0.230, 0.118, 0.058, 0.041, 0.076, 0.090, 0.077]
            ],
        }

        self.experiments[self.obs_names.index("Phosphorylated_cFos")] = {
            "EGF": [0, 0.060, 0.109, 0.083, 0.068, 0.049, 0.027, 0.017],
            "HRG": [0, 0.145, 0.177, 0.158, 0.598, 1.000, 0.852, 0.431],
        }
        self.error_bars[self.obs_names.index("Phosphorylated_cFos")] = {
            "EGF": [
                sd / np.sqrt(3) for sd in [0, 0.003, 0.021, 0.013, 0.016, 0.007, 0.003, 0.002]
            ],
            "HRG": [
                sd / np.sqrt(3) for sd in [0, 0.010, 0.013, 0.001, 0.014, 0.000, 0.077, 0.047]
            ],
        }

        # ------------------------------------------------------------------------

        self.experiments[self.obs_names.index("Phosphorylated_CREBw")] = {
            "EGF": [0, 0.446, 0.030, 0.000, 0.000],
            "HRG": [0, 1.000, 0.668, 0.460, 0.340],
        }
        self.error_bars[self.obs_names.index("Phosphorylated_CREBw")] = {
            "EGF": [sd / np.sqrt(3) for sd in [0, 0.0, 0.0, 0.0, 0.0]],
            "HRG": [sd / np.sqrt(3) for sd in [0, 0.0, 0.0, 0.0, 0.0]],
        }
        # ------------------------------------------------------------------------

        self.experiments[self.obs_names.index("cfos_mRNA")] = {
            "EGF": [0, 0.181, 0.476, 0.518, 0.174, 0.026, 0.000],
            "HRG": [0, 0.353, 0.861, 1.000, 0.637, 0.300, 0.059],
        }
        self.error_bars[self.obs_names.index("cfos_mRNA")] = {
            "EGF": [sd / np.sqrt(3) for sd in [0.017, 0.004, 0.044, 0.004, 0.023, 0.007, 0.008]],
            "HRG": [sd / np.sqrt(3) for sd in [0.017, 0.006, 0.065, 0.044, 0.087, 0.023, 0.001]],
        }
        # ------------------------------------------------------------------------

        self.experiments[self.obs_names.index("cFos_Protein")] = {
```

```
            "EGF": [0, 0.078, 0.216, 0.240, 0.320, 0.235],
            "HRG": [0, 0.089, 0.552, 0.861, 1.000, 0.698],
        }
        self.error_bars[self.obs_names.index("cFos_Protein")] = {
            "EGF": [sd / np.sqrt(3) for sd in [0, 0.036, 0.028, 0.056, 0.071, 0.048]],
            "HRG": [sd / np.sqrt(3) for sd in [0, 0.021, 0.042, 0.063, 0.000, 0.047]],
        }

        self.experiments[self.obs_names.index("dusp_mRNA")] = {
            "EGF": [0.000, 0.177, 0.331, 0.214, 0.177, 0.231],
            "HRG": [0.000, 0.221, 0.750, 1.000, 0.960, 0.934],
        }
        self.error_bars[self.obs_names.index("dusp_mRNA")] = {
            "EGF": [sd / np.sqrt(3) for sd in [0.033, 0.060, 0.061, 0.032, 0.068, 0.050]],
            "HRG": [sd / np.sqrt(3) for sd in [0.027, 0.059, 0.094, 0.124, 0.113, 0.108]],
        }

    @staticmethod
    def get_timepoint(obs_name) -> List[int]:
        """
        Time points at which experimental data was taken.
        """
        if obs_name in [
            "Phosphorylated_MEKc",
            "Phosphorylated_ERKc",
            "Phosphorylated_RSKw",
            "Phosphorylated_cFos",
        ]:
            return [0, 300, 600, 900, 1800, 2700, 3600, 5400]  # (Unit: sec.)
        elif obs_name == "Phosphorylated_CREBw":
            return [0, 600, 1800, 3600, 5400]
        elif obs_name == "cfos_mRNA":
            return [0, 600, 1200, 1800, 2700, 3600, 5400]
        elif obs_name in ["cFos_Protein", "dusp_mRNA"]:
            return [0, 900, 1800, 2700, 3600, 5400]
        assert False
```

5.2.6 Set lower/upper bounds of parameters to be estimated

Open `search_param.py`.

```
class SearchParam(object):

    ...

    def get_region(self):

        ...

        search_rgn = np.zeros((2, len(x) + len(y0)))

        search_rgn[:, C.V1] = [7.33e-2, 6.60e-01]
        search_rgn[:, C.K1] = [1.83e2, 8.50e2]
        search_rgn[:, C.V5] = [6.48e-3, 7.20e1]
        search_rgn[:, C.K5] = [6.00e-1, 1.60e04]
        search_rgn[:, C.V10] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K10] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.n10] = [1.00, 4.00]
        search_rgn[:, C.kf11] = [8.30e-13, 1.44e-2]
        search_rgn[:, C.kf12] = [8.00e-8, 5.17e-2]
        search_rgn[:, C.kf13] = [1.38e-7, 4.84e-1]
        search_rgn[:, C.V14] = [4.77e-3, 4.77e1]
        search_rgn[:, C.K14] = [2.00e2, 2.00e6]
        search_rgn[:, C.V15] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K15] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.kf18] = [2.20e-4, 5.50e-1]
        search_rgn[:, C.kr18] = [2.60e-4, 6.50e-1]
        search_rgn[:, C.V20] = [4.77e-3, 4.77e1]
        search_rgn[:, C.K20] = [2.00e2, 2.00e6]
        search_rgn[:, C.V21] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K21] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.V24] = [4.77e-2, 4.77e0]
        search_rgn[:, C.K24] = [2.00e3, 2.00e5]
        search_rgn[:, C.V25] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K25] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.kf26] = [2.20e-4, 5.50e-1]
        search_rgn[:, C.kr26] = [2.60e-4, 6.50e-1]
        search_rgn[:, C.V27] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K27] = [1.00e2, 1.00e4]
        search_rgn[:, C.V28] = [np.exp(-10), np.exp(10)]
```

```
        search_rgn[:, C.K28] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.V29] = [4.77e-2, 4.77e0]
        search_rgn[:, C.K29] = [2.93e3, 2.93e5]
        search_rgn[:, C.V30] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K30] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.V31] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K31] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.n31] = [1.00, 4.00]
        search_rgn[:, C.kf32] = [8.30e-13, 1.44e-2]
        search_rgn[:, C.kf33] = [8.00e-8, 5.17e-2]
        search_rgn[:, C.kf34] = [1.38e-7, 4.84e-1]
        search_rgn[:, C.V35] = [4.77e-3, 4.77e1]
        search_rgn[:, C.K35] = [2.00e2, 2.00e6]
        search_rgn[:, C.V36] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K36] = [1.00e2, 1.00e4]
        search_rgn[:, C.V37] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K37] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.kf40] = [2.20e-4, 5.50e-1]
        search_rgn[:, C.kr40] = [2.60e-4, 6.50e-1]
        search_rgn[:, C.V42] = [4.77e-3, 4.77e1]
        search_rgn[:, C.K42] = [2.00e2, 2.00e6]
        search_rgn[:, C.V43] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K43] = [1.00e2, 1.00e4]
        search_rgn[:, C.V44] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K44] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.kf47] = [1.45e-4, 1.45e0]
        search_rgn[:, C.kr47] = [6.00e-3, 6.00e1]
        search_rgn[:, C.kf48] = [2.70e-3, 2.70e1]
        search_rgn[:, C.kf49] = [5.00e-5, 5.00e-1]
        search_rgn[:, C.kr49] = [5.00e-3, 5.00e1]
        search_rgn[:, C.kf50] = [3.00e-3, 3.00e1]
        search_rgn[:, C.kf51] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.kr51] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.V57] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.K57] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.n57] = [1.00, 4.00]
        search_rgn[:, C.kf58] = [8.30e-13, 1.44e-2]
        search_rgn[:, C.kf59] = [8.00e-8, 5.17e-2]
        search_rgn[:, C.kf60] = [1.38e-7, 4.84e-1]
```

```
        search_rgn[:, C.kf61] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.kf62] = [2.20e-4, 5.50e-1]
        search_rgn[:, C.kr62] = [2.60e-4, 6.50e-1]
        search_rgn[:, C.kf63] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.KF31] = [np.exp(-10), np.exp(10)]
        search_rgn[:, C.nF31] = [1.00, 4.00]
        search_rgn[:, C.a] = [1.00e2, 5.00e2]
```

Note that the lower bound must be smaller than the upper bound.

5.2.7 Create a new model

BioMASS core functions require `ModelObject` in the first argument.

```
>>> from biomass import create_model
>>> model = create_model('cfos_model')  # Create a new BioMASS model object.
```

In the following examples, you will use the BioMASS model object: `model` created here for parameter estimation, visualization of simulation results, and sensitivity analysis.

5.3 Parameter estimation

An important step in the development of a mathematical model for a biological system is to identify model parameters. Parameters are adjusted to minimize the distance between model simulation and experimental data.

5.3.1 Using `optimize()` function

This function estimates unknown model parameters from experimental data by applying differential evolution[44].

- Set simulation conditions and the corresponding experimental data in `observable.py`
- Define an objective function to be minimized (`objective()`) in `problem.py`
- Set lower/upper bounds of parameters to be estimated in `search_param.py`

```
from tqdm import tqdm
from biomass import optimize

# Get 30 parameter sets, it will take more than a few hours
for x_id in tqdm(range(1, 31)):
    optimize(model, x_id=x_id, disp_here=False, optimizer_options={"workers": -1})
```

Here, `"workers"` specifies the number of processes to use (default: 1). Set to a larger number (e.g. the number of CPU cores available) for parallel execution of optimizations. For detailed information about `optimizer_options`, please refer to https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html.

The progress list will be saved in `out/{x_id}/` in your model directory:

```
differential_evolution step 1: f(x)= 4.96181
differential_evolution step 2: f(x)= 3.555
differential_evolution step 3: f(x)= 2.50626
differential_evolution step 4: f(x)= 2.00657
differential_evolution step 5: f(x)= 1.83556
differential_evolution step 6: f(x)= 1.28031
differential_evolution step 7: f(x)= 0.973207
differential_evolution step 8: f(x)= 0.741667
differential_evolution step 9: f(x)= 0.741667
differential_evolution step 10: f(x)= 0.735682
differential_evolution step 11: f(x)= 0.717266
```

```
differential_evolution step 12: f(x)= 0.603178
differential_evolution step 13: f(x)= 0.56934
differential_evolution step 14: f(x)= 0.56934
differential_evolution step 15: f(x)= 0.549331
differential_evolution step 16: f(x)= 0.459069
differential_evolution step 17: f(x)= 0.447772
differential_evolution step 18: f(x)= 0.430385
differential_evolution step 19: f(x)= 0.37085
differential_evolution step 20: f(x)= 0.37085
```

5.3.2 Data export and visualization

To visualize the distribution of estimated parameter values (Figure 5.2), run `savefig()` function.

```
from biomass.result import OptimizationResults

res = OptimizationResults(model)
# Export estimated parameters in CSV format
res.to_csv()
# Visualize estimated parameter sets
res.savefig(figsize=(16,5), boxplot_kws={"orient": "v"})
```

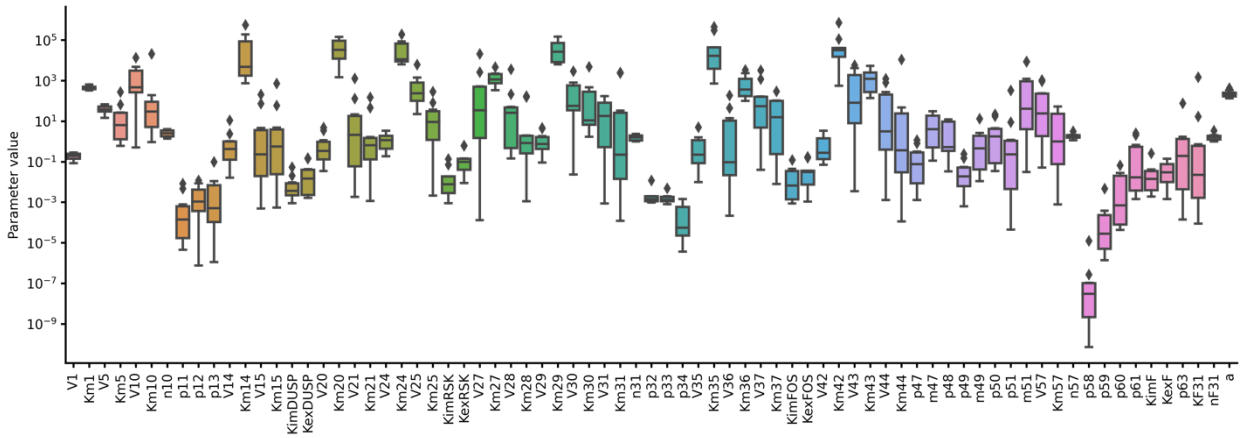


Figure 5.2: Estimated parameter values

You can also visualize objective function traces for different optimization runs (Figure 5.3).

```
res.trace_obj()
```

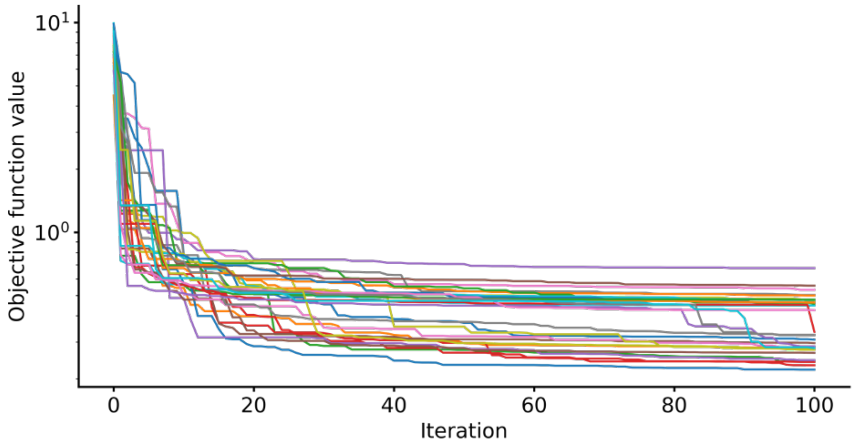


Figure 5.3: Objective function traces

5.3.3 Visualization of simulation results

Use `run_simulation()` function to get time-course simulation results.

```
from biomass import run_simulation


run_simulation(model, viz_type='average', show_all=False, stdev=True)
```

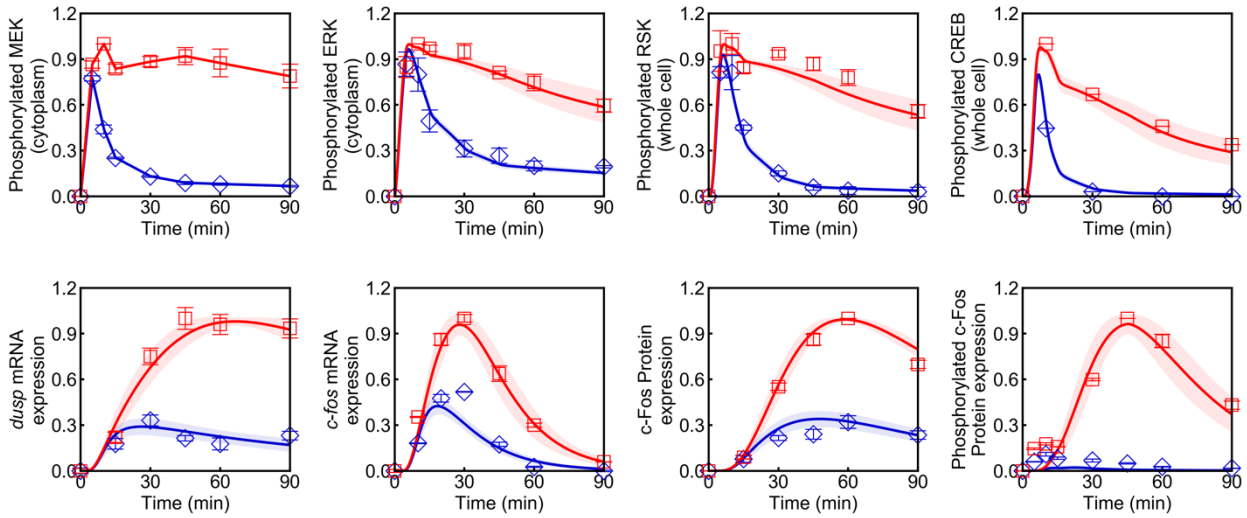


Figure 5.4: Time-course simulation results

Points (blue diamonds, EGF; red squares, HRG) denote experimental data, solid lines denote simulations.

5.4 Sensitivity analysis

Sensitivity analysis examines how perturbations to the processes in the model affect the quantity of interest, e.g., the integral of the pc-Fos concentration. To perform sensitivity analysis on reaction rates (`target='reaction'`), you will need to modify `reaction_network.py` in the model folder as follows:

```
class ReactionNetwork(object):


    def __init__(self) -> None:
        """
        Reaction indices grouped according to biological processes.
        This is used for sensitivity analysis (target='reaction').
        """
        super(ReactionNetwork, self).__init__()

        self.reactions: Dict[str, List[int]] = {
            "ERK_activation": [i for i in range(1, 7)],
```

```
            "ERK_dephosphorylation_by_DUSP": [i for i in range(47, 57)],
            "ERK_transport": [i for i in range(7, 10)],
            "RSK_activation": [24, 25],
            "RSK_transport": [26],
            "Elk1_activation": [29, 30],
            "CREB_activation": [27, 28],
            "dusp_production_etc": [i for i in range(10, 14)],
            "DUSP_transport": [18, 19],
            "DUSP_stabilization": [14, 15, 20, 21],
            "DUSP_degradation": [16, 17, 22, 23],
            "cfos_production_etc": [i for i in range(31, 35)],
            "cFos_transport": [40, 41],
            "cFos_stabilization": [35, 36, 37, 42, 43, 44],
            "cFos_degradation": [38, 39, 45, 46],
            "Feedback_from_F": [i for i in range(57, 64)],
        }


        ...
```

Then, run the following code:

```
from biomass import run_analysis


run_analysis(model, target='reaction', metric='integral', style='barplot',
options={'overwrite': True})
```

The single parameter sensitivity of each reaction is defined by

$$C_i^M = d \ln M / d \ln v_i$$

where v_i is the i^{th} reaction rate, v is reaction vector v = (v1, v2, …) and *M* is a signaling metric, e.g., time-integrated response, duration. Sensitivity coefficients are calculated using finite difference approximations with 1% changes in the reaction rates (Figure 5.5).

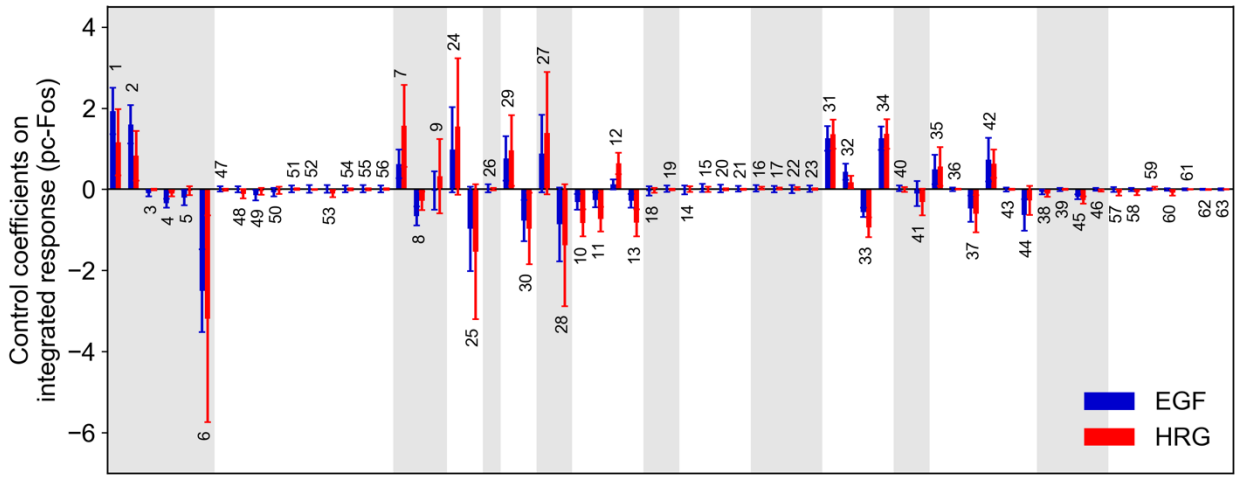


Figure 5.5: Control coefficients for integrated pc-Fos are shown by bars (blue, EGF; red, HRG). Numbers above bars indicate the reaction indices, and error bars correspond to simulation standard deviation.

Bibliography

1. Foulkes, W. D., Smith, I. E. & Reis-Filho, J. S. Triple-Negative Breast Cancer. *N. Engl. J. Med.* **363**, 1938–1948 (2010).

2. Weigelt, B., Baehner, F. L. & Reis-Filho, J. S. The contribution of gene expression profiling to breast cancer classification, prognostication and prediction: A retrospective of the last decade. *Journal of Pathology* **220**, 263–280 (2010).

3. Weinstein, J. N. *et al.* The cancer genome atlas pan-cancer analysis project. *Nat. Genet.* **45**, 1113–1120 (2013).

4. Citri, A. & Yarden, Y. EGF-ERBB signalling: Towards the systems level. *Nat. Rev. Mol. Cell Biol.* **7**, 505–516 (2006).

5. Schoeberl, B., Eichler-Jonsson, C., Gilles, E. D. & Muüller, G. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nat. Biotechnol.* **20**, 370–375 (2002).

6. Hatakeyama, M. *et al.* A computational model on the modulation of mitogen-activated protein kinase (MAPK) and Akt pathways in heregulin-induced ErbB signalling. *Biochem. J.* **373**, 451–463 (2003).

7. Kholodenko, B. N., Demin, O. V, Moehren, G. & Hoek, J. B. Quantification of short term signaling by the epidermal growth factor receptor. *J. Biol. Chem.* **274**, 30169–30181 (1999).

8. Schoeberl, B. *et al.* Therapeutically targeting ErbB3: A key node in ligand-induced activation of the ErbB receptor-PI3K axis. *Sci. Signal.* **2**, (2009).

9. Chen, W. W. *et al.* Input-output behavior of ErbB signaling pathways as revealed by a mass action model trained against dynamic data. *Mol. Syst. Biol.* **5**, (2009).

10. Hass, H. *et al.* Predicting ligand-dependent tumors from multi-dimensional signaling features. *npj Syst. Biol. Appl.* **3**, 27 (2017).

11. Kemmer, S. *et al.* Disentangling ERBB Signaling in Breast Cancer Subtypes—A Model-Based Analysis. *Cancers (Basel).* **14**, 2379 (2022).

12. Yarden, Y. & Sliwkowski, M. X. Untangling the ErbB signalling network. *Nat. Rev. Mol. Cell Biol.* **2**, 127–137 (2001).

13. Kholodenko, B. N. Cell-signalling dynamics in time and space. *Nature Reviews Molecular Cell Biology* **7**, 165–176 (2006).

14. Nagashima, T. *et al.* Quantitative transcriptional control of ErbB receptor signaling undergoes graded to biphasic response for cell differentiation. *J. Biol. Chem.* **282**, 4045–4056 (2007).

15. Nakakuki, T. *et al.* Ligand-specific c-Fos expression emerges from the spatiotemporal control of ErbB network dynamics. *Cell* **141**, 884–896 (2010).

16. Murphy, L. O., Smith, S., Chen, R. H., Fingar, D. C. & Blenis, J. Molecular, interpretation of ERK signal duration by immediate early gene products. *Nat. Cell Biol.* **4**, 556–564 (2002).

17. Ryl, T. *et al.* Cell-Cycle Position of Single MYC-Driven Cancer Cells Dictates Their Susceptibility to a Chemotherapeutic Drug. *Cell Syst.* **5**, 237-250.e8 (2017).

18. Imoto, H. & Okada, M. Signal-dependent regulation of early-response genes and cell cycle: a quantitative view. *Curr. Opin. Syst. Biol.* **15**, 100–108 (2019).

19. Mina, M. *et al.* Promoter-level expression clustering identifies time development of transcriptional regulatory cascades initiated by ERBB receptors in breast cancer cells. *Sci. Rep.* **5**, 1–13 (2015).

20. Dejure, F. R. & Eilers, M. MYC and tumor metabolism: chicken and egg. *EMBO J.* (2017). doi:10.15252/embj.201796438

21. Dagogo-Jack, I. & Shaw, A. T. Tumour heterogeneity and resistance to cancer therapies. *Nature Reviews Clinical Oncology* **15**, 81–94 (2018).

22. Ozaki, K. *et al.* Functional SNPs in the lymphotoxin-α gene that are associated with susceptibility to myocardial infarction. *Nat. Genet.* **32**, 650–654 (2002).

23. Gusev, A. *et al.* Integrative approaches for large-scale transcriptome-wide association studies. *Nat. Genet.* **48**, 245–252 (2016).

24. Van't Veer, L. J. *et al.* Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415**, 530–536 (2002).

25. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V. & Fotiadis, D. I. Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal* **13**, 8–17 (2015).

26. Clarke, M. A. & Fisher, J. Executable cancer models: successes and challenges. *Nat. Rev. Cancer* **20**, 343–354 (2020).

27. Fischer, H. P. Mathematical modeling of complex biological systems: From parts lists to understanding systems behavior. *Alcohol Research and Health* **31**, 49–59 (2008).

28. Kitano, H. Computational systems biology. *Nature* **420**, 206–210 (2002).

29. Fröhlich, F. *et al.* Efficient Parameter Estimation Enables the Prediction of Drug Response Using a Mechanistic Pan-Cancer Pathway Model. *Cell Syst.* **7**, 567-579.e6 (2018).

30. Barretina, J. *et al.* The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* **483**, 603–607 (2012).

31. Kolch, W., Halasz, M., Granovskaya, M. & Kholodenko, B. N. The dynamic control of signal transduction networks in cancer cells. *Nat. Rev. Cancer* **15**, 515–527 (2015).

32. Saez-Rodriguez, J. & Blüthgen, N. Personalized signaling models for personalized treatments. *Mol. Syst. Biol.* **16**, (2020).

33. Ebata, K., Yamashiro, S., Iida, K. & Okada, M. Building patient-specific models for receptor tyrosine kinase signaling networks. *FEBS Journal* **289**, 90–101 (2022).

34. Hernandez-Boussard, T. *et al.* Digital twins for predictive oncology will be a paradigm shift for precision cancer care. *Nature Medicine* **27**, 2065–2066 (2021).

35. Arteaga, C. L. & Engelman, J. A. ERBB receptors: From oncogene discovery to basic science to mechanism-based cancer therapeutics. *Cancer Cell* **25**, 282–303 (2014).

36. Xu, J., Chen, Y. & Olopade, O. I. MYC and Breast Cancer. *Genes Cancer* **1**, 629–640 (2010).

37. Muzny, D. M. *et al.* Comprehensive molecular characterization of human colon and rectal cancer. *Nature* **487**, 330–337 (2012).

38. Birtwistle, M. R. *et al.* Ligand-dependent responses of the ErbB signaling network:

Experimental and modeling analyses. *Mol. Syst. Biol.* **3**, (2007).

39. Lee, T., Yao, G., Nevins, J. & You, L. Sensing and integration of Erk and PI3K signals by Myc. *PLoS Comput. Biol.* **4**, (2008).

40. Imoto, H., Zhang, S. & Okada, M. A Computational Framework for Prediction and Analysis of Cancer Signaling Dynamics from RNA Sequencing Data—Application to the ErbB Receptor Signaling Pathway. *Cancers (Basel).* **12**, 2878 (2020).

41. Imoto, H., Yamashiro, S. & Okada, M. A text-based computational framework for patient -specific modeling for classification of cancers. *iScience* **25**, 103944 (2022).

42. Degasperi, A. *et al.* Evaluating strategies to normalise biological replicates of western blot data. *PLoS One* **9**, (2014).

43. Kiyatkin, A. & Aksamitiene, E. Multistrip western blotting to increase quantitative data output. *Methods Mol. Biol.* **536**, 149–161 (2009).

44. Storn, R. & Price, K. Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **11**, 341–359 (1997).

45. Van Der Walt, S., Colbert, S. C. & Varoquaux, G. The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Eng.* **13**, 22–30 (2011).

46. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).

47. Colaprico, A. *et al.* TCGAbiolinks: An R/Bioconductor package for integrative analysis of TCGA data. *Nucleic Acids Res.* **44**, e71 (2016).

48. Zhang, Y., Parmigiani, G. & Johnson, W. E. ComBat-seq: batch effect adjustment for RNA-seq count data. *NAR Genomics Bioinforma.* **2**, (2020).

49. Yu, K. *et al.* Comprehensive transcriptomic analysis of cell lines as models of primary tumors across 22 tumor types. *Nat. Commun.* **10**, (2019).

50. Robinson, M. D. & Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.* **11**, (2010).

51. Wagner, G. P., Kin, K. & Lynch, V. J. Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory Biosci.* **131**, 281–285 (2012).

52. Durinck, S., Spellman, P. T., Birney, E. & Huber, W. Mapping identifiers for the integration of genomic datasets with the R/ Bioconductor package biomaRt. *Nat. Protoc.* **4**, 1184–1191 (2009).

53. Cibulskis, K. *et al.* Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nat. Biotechnol.* **31**, 213–219 (2013).

54. Shoemaker, J. E. & Doyle, F. J. Identifying fragilities in biochemical networks: Robust performance analysis of fas signaling-induced apoptosis. *Biophys. J.* **95**, 2610–2623 (2008).

55. Hornberg, J. J. *et al.* Control of MAPK signalling: From complexity to what really matters. *Oncogene* **24**, 5533–5542 (2005).

56. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).

57. Waskom, M. seaborn: statistical data visualization. *J. Open Source Softw.* **6**, 3021 (2021).

58. Heinrich, R., Neel, B. G. & Rapoport, T. A. Mathematical models of protein kinase

signal transduction. *Mol. Cell* **9**, 957–970 (2002).

59. Kholodenko, B., Yaffe, M. B. & Kolch, W. Computational approaches for analyzing information flow in biological networks. *Science Signaling* **5**, (2012).

60. Karr, J. R. *et al.* A whole-cell computational model predicts phenotype from genotype. *Cell* **150**, 389–401 (2012).

61. Goldberg, A. P. *et al.* Emerging whole-cell modeling principles and methods. *Current Opinion in Biotechnology* **51**, 97–102 (2018).

62. Macklin, D. N. *et al.* Simultaneous cross-evaluation of heterogeneous E. coli datasets via mechanistic simulation. *Science (80-.).* **369**, (2020).

63. Bouhaddou, M. *et al.* A mechanistic pan-cancer pathway model informed by multi-omics data interprets stochastic cell fate responses to drugs and mitogens. *PLoS Comput. Biol.* **14**, 1–37 (2018).

64. Raue, A. *et al.* Lessons Learned from Quantitative Dynamical Modeling in Systems Biology. *PLoS One* **8**, (2013).

65. Fröhlich, F., Kaltenbacher, B., Theis, F. J. & Hasenauer, J. Scalable Parameter Estimation for Genome-Scale Biochemical Reaction Networks. *PLoS Comput. Biol.* **13**, 1–18 (2017).

66. Stapor, P. *et al.* Mini-batch optimization enables training of ODE models on large-scale datasets. *Nat. Commun.* **13**, (2022).

67. Raue, A. *et al.* Data2Dynamics: A modeling environment tailored to parameter estimation in dynamical systems. *Bioinformatics* **31**, 3558–3560 (2015).

68. Maiwald, T. & Timmer, J. Dynamical modeling and multi-experiment fitting with PottersWheel. *Bioinformatics* **24**, 2037–2043 (2008).

69. Santra, T. Fitting mathematical models of biochemical pathways to steady state perturbation response data without simulating perturbation experiments. *Sci. Rep.* **8**, (2018).

70. Kholodenko, B. N., Hoek, J. B., Westerhoff, H. V. & Brown, G. C. Quantification of information transfer via cellular signal transduction pathways. *FEBS Lett.* **414**, 430–434 (1997).

71. Dai, X., Cheng, H., Bai, Z. & Li, J. Breast cancer cell line classification and Its relevance with breast tumor subtyping. *J. Cancer* **8**, 3131–3141 (2017).

72. Nica, A. C. & Dermitzakis, E. T. Expression quantitative trait loci: Present and future. *Philos. Trans. R. Soc. B Biol. Sci.* **368**, (2013).

73. Kimura, S. & Konagaya, A. A Genetic Algorithm with Distance Independent Diversity Control for High Dimensional Function Optimization. *J. Japanese Soc. Artif. Intell.* **18**, 193–202 (2003).

74. Purvis, J. E. & Lahav, G. Encoding and decoding cellular information through signaling dynamics. *Cell* **152**, 945–956 (2013).

75. Zhong, L. *et al.* Small molecules in targeted cancer therapy: advances, challenges, and future perspectives. *Signal Transduction and Targeted Therapy* **6**, (2021).

76. Inoue, T., Terada, N., Kobayashi, T. & Ogawa, O. Patient-derived xenografts as in vivo models for research in urological malignancies. *Nature Reviews Urology* **14**, 267–283 (2017).

77. Whittle, J. R., Lewis, M. T., Lindeman, G. J. & Visvader, J. E. Patient-derived xenograft models of breast cancer and their predictive power. *Breast Cancer Research* **17**, (2015).

78. Yoshida, G. J. Applications of patient-derived tumor xenograft models and tumor organoids. *Journal of Hematology and Oncology* **13**, (2020).

79. Niepel, M. *et al.* Profiles of basal and stimulated receptor signaling networks predict drug response in breast cancer lines. *Sci. Signal.* **6**, (2013).

80. Jiang, G. *et al.* Comprehensive comparison of molecular portraits between cell lines and tumors in breast cancer. *BMC Genomics* **17**, (2016).

81. Koboldt, D. C. *et al.* Comprehensive molecular portraits of human breast tumours. *Nature* **490**, 61–70 (2012).

82. Nielsen, T. O. *et al.* A comparison of PAM50 intrinsic subtyping with immunohistochemistry and clinical prognostic factors in tamoxifen-treated estrogen receptor-positive breast cancer. *Clin. Cancer Res.* **16**, 5222–5232 (2010).

83. Xie, Y. H., Chen, Y. X. & Fang, J. Y. Comprehensive review of targeted therapy for colorectal cancer. *Signal Transduction and Targeted Therapy* **5**, (2020).

84. Strippoli, A. *et al.* C-myc expression is a possible keystone in the colorectal cancer resistance to egfr inhibitors. *Cancers (Basel).* **12**, (2020).

85. Marshall, C. J. Specificity of receptor tyrosine kinase signaling: Transient versus sustained extracellular signal-regulated kinase activation. *Cell* **80**, 179–185 (1995).

86. Kholodenko, B. N., Hancock, J. F. & Kolch, W. Signalling ballet in space and time. *Nat. Rev. Mol. Cell Biol.* **11**, 414–426 (2010).

87. Johnson, H. E. & Toettcher, J. E. Signaling Dynamics Control Cell Fate in the Early Drosophila Embryo. *Dev. Cell* **48**, 361-370.e3 (2019).

88. Manning, C. S. *et al.* Quantitative single-cell live imaging links HES5 dynamics with cell-state and fate in murine neurogenesis. *Nat. Commun.* **10**, (2019).

89. Purvis, J. E. *et al.* p53 dynamics control cell fate. *Science (80-.).* **336**, 1440–1444 (2012).

90. Sasagawa, S., Ozaki, Y. I., Fujita, K. & Kuroda, S. Prediction and validation of the distinct dynamics of transient and sustained ERK activation. *Nat. Cell Biol.* **7**, 365–373 (2005).

91. Rauch, N., Rukhlenko, O. S., Kolch, W. & Kholodenko, B. N. MAPK kinase signalling dynamics regulate cell fate decisions and drug resistance. *Current Opinion in Structural Biology* **41**, 151–158 (2016).

92. Fey, D. *et al.* Signaling pathway models as biomarkers: Patient-specific simulations of JNK activity predict the survival of neuroblastoma patients. *Sci. Signal.* **8**, 1–16 (2015).

93. Imoto, H., Yamashiro, S., Murakami, K. & Okada, M. Protocol for stratification of triple-negative breast cancer patients using in silico signaling dynamics. *STAR Protoc.* **3**, 101619 (2022).

94. Imoto, H. & Yamashiro, S. pasmopy/breast_cancer: Protocol for stratification of TNBC patients. (2022). doi:10.5281/ZENODO.6781265

95. Park, H. S. *et al.* High EGFR gene copy number predicts poor outcome in triple-negative breast cancer. *Mod. Pathol.* **27**, 1212–1222 (2014).

96. Jafarnejad, M. *et al.* Mechanistically detailed systems biology modeling of the HGF/Met pathway in hepatocellular carcinoma. *npj Syst. Biol. Appl.* **5**, (2019).

97. Collisson, E. A. *et al.* Comprehensive molecular profiling of lung adenocarcinoma: The cancer genome atlas research network. *Nature* **511**, 543–550 (2014).

98. Zhang, C. *et al.* RAF inhibitors that evade paradoxical MAPK pathway activation. *Nature* **526**, 583–586 (2015).

99. Sanchez-Laorden, B. *et al.* BRAF inhibitors induce metastasis in RAS mutant or inhibitor-resistant melanoma cells by reactivating MEK and ERK signaling. *Sci. Signal.* **7**, (2014).

100. Kholodenko, B. N. Drug Resistance Resulting from Kinase Dimerization Is Rationalized by Thermodynamic Factors Describing Allosteric Inhibitor Effects. *Cell Rep.* **12**, 1939–1949 (2015).

101. Rukhlenko, O. S. *et al.* Dissecting RAF Inhibitor Resistance by Structure-based Modeling Reveals Ways to Overcome Oncogenic RAS Signaling. *Cell Syst.* **7**, 161-179.e14 (2018).

102. Kholodenko, B. N., Rauch, N., Kolch, W. & Rukhlenko, O. S. A systematic analysis of signaling reactivation and drug resistance. *Cell Rep.* **35**, (2021).

103. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).

104. Jumper, J. & Hassabis, D. Protein structure predictions to atomic accuracy with AlphaFold. *Nature Methods* **19**, 11–12 (2022).

105. Baek, M. *et al.* Accurate prediction of protein structures and interactions using a

three-track neural network. *Science (80-.).* **373**, 871–876 (2021).

106. Humphreys, I. *et al.* Computed structures of core eukaryotic protein complexes. *Science (80-.).* **374**, (2021).

107. Yu, M. K. *et al.* Visible Machine Learning for Biomedicine. *Cell* **173**, 1562–1565 (2018).

108. Yang, J. H. *et al.* A White-Box Machine Learning Approach for Revealing Antibiotic Mechanisms of Action. *Cell* **177**, 1649-1661.e9 (2019).

109. Baker, R. E., Peña, J. M., Jayamohan, J. & Jérusalem, A. Mechanistic models versus machine learning, a fight worth fighting for the biological community? *Biol. Lett.* **14**, (2018).

110. Ederer, M. & Gilles, E. D. Thermodynamically feasible kinetic models of reaction networks. *Biophys. J.* **92**, 1846–1857 (2007).

111. Petzold, L. Automatic Selection of Methods for Solving Stiff and Nonstiff Systems of Ordinary Differential Equations. *SIAM J. Sci. Stat. Comput.* **4**, 136–148 (1983).