Sagittarius: Extrapolating Heterogeneous Time-Series Gene Expression Data

Addie Woicik[1], Mingxin Zhang[1,2], Janelle Chan[1], Jianzhu Ma[3], Sheng Wang[1#]

[1]School of Computer Science and Engineering, University of Washington, Seattle, WA

[2]Department of Information and Communications Engineering, Tokyo Institute of Technology, Tokyo, Japan

[3]Institute for Artificial Intelligence, Peking University, Beijing, China

[#]Email: swang@cs.washington.edu

Abstract

Understanding the temporal dynamics of gene expression is crucial for developmental biology, tumor biology, and biogerontology. However, some time points remain challenging to measure in the lab, particularly during very early or very late stages in a biological process. Here we propose Sagittarius, a transformer-based model that is able to accurately simulate gene expression profiles at time points outside of the range of times measured in the lab. The key idea behind Sagittarius is to learn a shared reference space that generates simulated time series measurements, thereby explicitly modeling unaligned time points and conditional batch effects between time series and making the model widely applicable to diverse biological settings. We show the promising performance of Sagittarius when extrapolating mammalian developmental gene expression, simulating drug-induced expression at unmeasured dose and treatment times, and augmenting datasets to accurately predict drug sensitivity. We also used Sagittarius to simulate mutation profiles for early-stage cancer patients, which further enabled us to discover a gene set related to the Hedgehog signaling pathway that may be related to tumorigenesis in sarcoma patients, including *PTCH1*, *ARID2*, and *MYCBP2*. By augmenting experimental temporal datasets with crucial but difficult-to-measure simulated datapoints, Sagittarius enables deeper insights into the temporal dynamics of heterogeneous transcriptomic processes and can be broadly applied to biological time series extrapolation.

Main

The temporal dynamics of the transcriptome are key to the study of developmental biology,[1,2] tumor biology,[3,4] immunobiology,[5,6] and pharmacogenomics.[7,8] As bulk- and single-cell RNA-sequencing technologies have become cheaper,[4,9–11] more transcriptomic datasets include gene expression measurements at multiple time points.[12–19] However, although such datasets are becoming more common, it often remains a significant challenge to measure transcriptomic profiles at very early or late stages of a biological process. For instance, senescent and extremely diseased tissue under different experimental conditions can be challenging to measure, but are of extreme interest to studies for aging and therapeutics.

The underlying problem here is temporal extrapolation, where time points of interest are outside the range of time that has experimental measurements. Accurate extrapolation on a single time series is very challenging due to non-stationary features and temporal out-of-domain adaptation.[20] Other works, such as Monocle,[13] Slingshot,[21] and Palantir,[22] aim to impute pseudotime points from a single measurement of time series data, but cannot make use of recent datasets[12–19] that contain measurements at multiple labeled time points to simulate novel measurements. One possible solution for the extrapolation problem is to combine sparse time series measurements from heterogeneous sequences. In particular, mouse[12] and roundworm[23] transcriptomic time series measurements, combined with developmental human measurements, can help simulate early-stage embryonic transcriptomic profiles for human.[24] There are two major challenges in effectively utilizing other sequences: unaligned measured time points and batch effects between experimental conditions. Existing methods are unable to simultaneously consider the full sequence of measured time points[25,26] or take into account the temporal batch effects between time series.[27–30]

To address these limitations we propose Sagittarius, a model that maps heterogeneous gene expression time series to the same reference space based on inferred biological age rather than the observed age, enabling multiple sparsely measured time series to jointly inform extrapolation to diverse time series. Sagittarius leverages a novel transformer-based architecture with multi-head attention[31] to map the heterogeneous set of time series from the irregular, unaligned, sparse measurement space to the regular reference space shared by all time series, using high-frequency embeddings of the timestamp[29,32] and experimental condition labels of each time series to define the mapping. After mapping to the shared reference space, we can accurately simulate new genomic profiles at extrapolated time points, as well as simulate measurements for unmeasured combinations of experimental conditions.

We evaluated Sagittarius on three diverse applications in developmental biology, pharmacogenomics, and cancer genomics. On the Evo-devo development dataset,[12] we show that Sagittarius has a Pearson correlation of 0.976 for gene expression profile extrapolation, compared to the best existing method's correlation of 0.926. Sagittarius further enabled organ-specific transcriptomic velocity analysis that clearly illustrates the diverging tissue trajectory during development, and which we further verified with scRNA-seq datasets from Tabula Muris Senis.[16] To evaluate Sagittarius's robustness to extremely sparse measurements, we next applied it to the LINCS pharmacogenomics dataset,[15] where it was able to simulate drug-induced expression with a correlation of 0.89 for test cell line, drug, dose, and time perturbation experiments, although only 1.77% of possible drug and cell line combinations are measured in the dataset. Furthermore, the model's shared reference space and simulated expression enable us to perform a novel drug repurposing task across perturbation combinations that do not share a drug or a cell line. Sagittarius obtained the best average Spearman correlation of 0.49 on two large-scale drug response datasets,[33,34] as well as an average cell line Spearman correlation of 0.816 and 0.789 for cancer gene essentiality prediction on both CRISPR-[35] and shRNA-based[36] measurements respectively, compared to 0.261 and 0.278 using only available *in vitro* data. We finally applied Sagittarius to the sarcoma and thyroid carcinoma cancer types in The Cancer Genome Atlas (TCGA) dataset.[37] Sagittarius was able to accurately simulate mutation profiles for patients with very long survival times, usually representing early-stage cancer patients with driver mutations that are difficult to measure in the clinic. For example, when using sarcoma patients with a post-biopsy survival time longer than 37 months as test data and all other patients for training, Sagittarius had a 0.77 AUROC for simulating mutation profiles, a 12.3% improvement over existing methods. This leads us to discover a novel early-stage gene set related to the Hedgehog signaling pathway and *GLI* oncogene, which can potentially drive tumorigenesis in early-stage sarcoma patients.

Results

Overview of Sagittarius

Given a heterogeneous, unaligned, sparse, and irregular time series dataset of genomic measurements, Sagittarius is able to simulate gene expression profiles for unmeasured time points (**Fig. 1)**. After training, a user may obtain the simulated expression from Sagittarius for unmeasured time points of an experiment present in the dataset or for new combinations of experimental variables that are not present in the initial dataset, such as the human heart in **Fig. 1d**, provided that both the human species and heart organ were measured at least once in the training dataset (**Fig. 1a**).

The key idea behind Sagittarius is to learn a shared reference space (**Fig. 1c)**, which underlies all heterogeneous time series in the dataset. The common reference space addresses two main challenges: temporal extrapolation and batch effects between experimental conditions. First, the common reference space pools dynamic information from all of the sparse time series to a single, global understanding of the underlying temporal trajectory in the data. Given the complete range of biological ages that are measured by at least one time series in the dataset, this then enables accurate extrapolation for time series with measurements that only cover a subset of the dataset's complete biological age range. Second, the compression and alignment to the common reference space (**Fig. 1b,c**) disentangles the experimental variables, like species and organ, from the time series representations, both facilitating a comparison based on biological age rather than measured age and enabling easy simulation of time series for unmeasured combinations of experimental variables (**Fig. 1d**).

Sagittarius is able to infer relative time relationships between different experimental conditions, thereby leveraging measurements within a related range of one time series to simulate accurate profiles for extrapolated time points in another time series. The generative network and continuous transformer are efficient, enabling Sagittarius to simulate new observations powered by large datasets. In addition to downstream analyses such as developmentally dynamic gene modeling and tumorigenesis driver identification, Sagittarius can be applied to complex pharmacogenomic datasets containing both a dose and treatment time continuous variable, extending the common reference space into multiple temporal dimensions. This further enables drug repurposing and drug response prediction using the simulated drug-induced gene expression data from Sagittarius, extrapolating dose, treatment time, and perturbation combinations.

Extrapolating heterogeneous gene expression to unmeasured time points using Sagittarius

To assess the merit of our approach, we evaluated whether Sagittarius can simulate gene expression profiles for a time point later than the measured time points using gene expression time series from multiple experimental conditions. We used the Mammalian Organ Development Evo-devo time series dataset,[12] which contains bulk RNA-seq data from 7 organs across 7 species, measured at a total of 91 distinct time points, where each time series ranges between 9 and 23 measured time points, with the fewest measurements for chicken and the most for human. The provided time points give the developmental stage of the species, but not the aligned biological age between species. Furthermore, the ranges of development that are covered by each species differ; primates include measurements during senescence, while rhesus macaque and chicken do not contain early embryonic data. Therefore, the Evo-devo dataset can assess whether our method

can handle unaligned absolute time points as well as differing biological age ranges measured across species.

To initially validate our model, we hid the last four measured time points from each species' organ time series and provided the remainder of the Evo-devo dataset to Sagittarius as training data. After training, we then simulated gene expression vectors for each species' organs at the four hidden time points and compared them to the measured expression vectors in the Evo-devo dataset. Sagittarius achieved an average Pearson correlation between the simulated and measured gene expression vectors of 0.976 when ranking by genes, and a Pearson correlation of 0.367 when ranking by time points, with a 0.109 average root mean squared error (RMSE), compared to 0.926, 0.070, and 0.163 respectively for the best-performing comparison approach (**Supplementary Fig. 1** and **Methods**). We attribute our improved performance to the alignment of all species in the shared reference space, enabling Sagittarius to make predictions for aging patterns in one species's organ based on its trajectory's similarity to other time series in the dataset that include later developmental measurements, even if they correspond to a different species and organ.

To further validate Sagittarius's improved gene expression simulation, we then subdivided our results into individual species and organs. We first noticed that our method still achieves the best performance on all organs and on 6 out of 7 species (**Fig. 2**). The best simulated transcriptomic profiles were from the mouse testis extrapolated time series. Importantly, after hiding the last four measured time points to use as test data for this task, the final training time point for mouse testis development is postnatal day 0. This demonstrates the benefit of the shared reference space for the time series, as other species with later developmental stages included in the dataset enable Sagittarius to effectively transfer knowledge and patterns to the later developmental stages in the mouse testis dataset. In contrast, all methods struggle on the human extrapolation task. We believe that this is because, after removing the four latest measurements for each species from the training set, the human extrapolation task involves time points that are much later developmentally than any still present in the training dataset, and is therefore the most difficult for any method to accurately simulate. Although significantly better than the next-best-performing method (Fisher transform[38] followed by one-sided t-test $p < 0.05$ for compare-by-time Pearson correlation), Sagittarius's second-worst performing species is chicken. We believe that this reflects the fact that chicken, the only bird species in the dataset, is less evolutionarily related to the mammalian species in the dataset,[12] thereby highlighting that Sagittarius has a larger improvement over existing approaches when the species in the reference space follow more similar developmental trajectories.

After finding that chicken measurements were more difficult to simulate than other, more related

176 species, we wanted to study how Sagittarius grouped different tissues across developmental time
177 points. Sagittarius's improved extrapolation performance led us to consider whether we could
178 simulate samples for time points that would be impossible to measure experimentally and
179 thereby gain new insights into tissue differentiation and aging.
180

181 **Tissue transcriptomics velocity analysis reveals organ-specific aging genes**

182 To further examine the biological insights from Sagittarius's extrapolated expression profiles, we
183 next simulated gene expression profiles for each mouse organ at 180 different time points. We
184 emphasized early mouse embryonic development, so 50 of our simulated time points were earlier
185 than any mouse measurements present in the dataset; our latest generation time point
186 corresponded to a 63-day-old mouse, the latest measured mouse time point. By simulating early
187 time points, we expect to observe a hypothetical trajectory that includes organogenesis, which
188 takes place between embryonic days 6.5 and 8.5 in mouse development.[39,40] That is, we expect that
189 the earliest simulated time points result in very similar expression profiles across the different
190 queried organs, which would not have differentiated at this stage. At later time points, we expect
191 that the organ time series diverge according to germ layers, before finally separating by organ.
192 We visualized the uniform manifold approximation and projection[41] (UMAP) embedding of the
193 simulated time series results (**Fig. 3a,b**), as well as the top principal components[42]
194 (**Supplementary Fig. 2**). In particular, we found that the UMAP representations of the
195 hypothetical mouse organ development diverged according to organ at later developmental
196 stages (**Fig. 3a**), but generally started from a common, central location in the embedding space
197 (**Fig. 3b)**. This indicates that the developmental stage, rather than tissue differentiation,
198 dominates the simulated gene expression measurements at the earliest time points, while tissue-
199 specific genes begin to separate the embeddings at later developmental stages. At later time
200 points, we found that the simulated expression values for brain and cerebellum were more closely
201 grouped together, as well as early expression for the heart, ovary, and testis, consistent with the
202 ectoderm, mesoderm, and endoderm tissue germ layer classifications.[12] This supports the existing
203 biological theory that expression trajectories are most shared between organs at early
204 developmental stages before differentiation by germ layer and finally organ,[12,13,39,43,44] and shows
205 Sagittarius's ability to extrapolate to unmeasured early developmental stages by discerning
206 common developmental trajectories for each organ across species.
207

208 Given the increasing tissue-specific signal in Sagittarius's simulated gene expression vectors at
209 later time points, we then investigated which genes most contributed to the differentiation of
210 organ trajectories during development. Excluding the heart and cerebellum, which we found to
211 be the most developmentally distinct for many genes in **Fig. 3a,b**, we aimed to identify a gene

that had similar expression across all organs at early developmental time points and diverse expression patterns at later developmental time points. We found that *Xrn2* expression levels significantly differed across organs at later time points (ANOVA p-value < 1e-98), although all organs were comparable at early simulated time points (ANOVA p-value > 0.05). *Xrn2* is also one of 12 protein-coding genes predictive of liver cancer prognosis, where high *Xrn2* expression levels indicate worse outcomes,[45] and we found that the liver in particular had lower simulated expression levels than other organs at later developmental time points (**Fig. 3c**).

We then sought to further validate Sagittarius's organ-specific extrapolation potential using the Tabula Muris Senis single cell RNA-seq dataset.[16] Although the Evo-devo dataset contains up to 14 bulk measurements for each mouse organ, the latest measurement is at postpartum day 63. In contrast, the Tabula Muris Senis dataset contains measurements ranging from a 1-month-old to a 30-month-old mouse. We consequently simulated transcriptomic profiles for 140 time points, beginning from postpartum day 14. We compared the Pearson correlation of the gene expression over time between the simulated profiles and the Tabula Muris Senis data, and found that for genes including *Egflam*, *Smoc1*, *Slc6a2*, and especially *Rpl38*, which previous work has suggested could regulate developmental processes in a tissue-specific way,[46] Sagittarius's simulated aging trajectory better aligned with the tissue trajectories in the Tabula Muris Senis dataset than the Evo-devo measured mouse data alone (**Fig. 3d**). This again shows the value of the shared reference space, which can identify patterns from species with later measured developmental time points like human and rhesus macaque to inform simulated transcriptomes for mouse aging. After applying Sagittarius to the Evo-devo dataset with the continuous time variable, we next considered whether the model could successfully extrapolate unmeasured experimental combinations in settings with multiple temporal variables.

Sagittarius simulates drug-induced expression for unmeasured cell line perturbations

We next sought to evaluate Sagittarius to extremely sparse multivariate data with multiple continuous temporal variables, thereby exponentially increasing the space of possible experimental settings. We applied Sagittarius to the larger, high-dimensional LINCS L1000 pharmacogenomics dataset.[15] In the LINCS dataset, compounds are experimentally applied to cell lines at specific doses and for a given treatment time before the gene expression vector of 978 genes is measured, although only 1.77% of possible drug and cell line combinations are measured. Sagittarius models each treatment experiment in two continuous dimensions: dose and treatment time. Each cell line is never experimentally treated with many of the drugs, and the perturbations that are tested have sparse measurements over dose and time (**Fig. 4a**).

To validate Sagittarius's ability to extrapolate to new perturbation experiments, we considered each dose and time combination for a single drug and cell line to make up the measured sequence for that combination. We then designed three extrapolation tasks: complete generation, combination & dose, and combination & time (**Fig. 4b** and **Methods**). For each task, we trained Sagittarius on a subset of the LINCS dataset, withholding the remaining measurements as test data. We then compared the Spearman correlation of Sagittarius's simulated drug-induced gene expression vector to a conditional Variational Autoencoder's (cVAE's)[25] simulated expression vector for each of our test perturbations (**Fig. 4c-e**). We found that Sagittarius achieved an average Spearman correlation of 0.84 per test drug for the complete generation task, relative to 0.79 for the cVAE (Fisher transform and one-sided t-test p-value < 5e-2); an average correlation of 0.922 for the combination & dose task, relative to 0.876 for the cVAE (Fisher transform and one-sided t-test p-value < 5e-92); and an average correlation of 0.921 for the combination & time task, relative to 0.809 for the cVAE (Fisher transform and one-sided t-test p-value < 5e-301). This indicates that Sagittarius can simulate accurate drug-induced gene expression vectors for unmeasured drug treatment experiments at doses and times that are not contained in the training data by aligning all perturbations experiments to the shared reference space. The simulated drug-induced transcriptomic profile enables an easy, unbiased search approach to drug sensitivity markers. This can greatly increase our understanding of the molecular basis of cancer and of drug response.

A drug sensitivity similarity network enables novel drug repurposing

As Sagittarius can compare the dose and time effects of each drug treatment experiment in the shared reference space, we then investigated the drug-induced expression similarity of the perturbation experiments for cancer drug repurposing. For each measured treatment in the dataset, we simulated drug-induced gene expression vectors at 78 different dosages with a fixed treatment time of 6 hours. We constructed a k-nearest-neighbors (KNN) graph G_{KNN} of perturbation experiments, weighting network edges by the similarity of Sagittarius's simulated drug-induced expression values for the two nodes' experiments. We next applied Louvain community detection to G_{KNN}, resulting in four large communities. We used an independent drug response dataset from Genomics of Drug Sensitivity in Cancer (GDSC)[33] to identify the half-maximal inhibitory concentration (IC_{50}) for every drug and cell line perturbation combination that appeared in both the LINCS and GDSC dataset. Finally, we labeled each of our four communities in G_{KNN} with the average IC_{50} of all nodes within the community that had a label in the GDSC dataset (**Fig. 5a**).

We found that the communities in G_{KNN} demonstrated a pattern with respect to sensitivity, with average IC_{50} dosages of 1.68, 1.83, 1.90, and 2.40 uM respectively. In previous works, gene expression has been widely used to identify the drug-induced and diseased-induced gene expression signatures in drug repurposing studies,[47–49] partly due to the scale at which analyses can be efficiently performed and validated. As Sagittarius can accurately simulate expression for any perturbation combinations, we next sought to apply Sagittarius to drug repurposing. To evaluate this, we investigated G_{KNN} and found that nearby perturbation experiments in the KNN network indicate potential drug repurposing opportunities. For example, we identified the 8-experiment subgraph from the most sensitive community, shown in the inset of **Fig. 5a**.

The subnetwork's MCF7 and A549 cell line perturbations are all also measured in the GDSC dataset, with IC_{50} values of 0.40, 0.89, 1.03, 0.49, 0.67, and 0.83 uM respectively. This demonstrates the network's potential for drug repurposing: Sagittarius connects to recent work for drug repurposing based on cell-line gene expression signatures,[50] as the subnetwork includes A549 treated with Vorinostat, Gefitinib, and Selumetinib, each of which A549 is sensitive to; Sagittarius also connects to existing work on repurposing for similar cell lines given a drug's mechanism of action,[51] as the subnetwork includes both A549 and MCF7 treated with Gefitinib, to which they are sensitive.

Importantly, as Sagittarius can simulate drug-induced expression profiles for any drug and cell line combination at any dosage and time, our network also enables drug repurposing for entirely new treatment combinations, where neither the drug nor the cell line needs to be the same as known successful therapy. The 8-perturbation subnetwork also includes PC3 treated with Pictilisib, which was colocated with the other perturbation experiments because Sagittarius simulated differential expression signatures similar to those it simulated for the other six experiments, although neither PC3 nor Pictilisib are present elsewhere in the subnetwork. Although these experiments were not present in the GDSC dataset, previous work[52] found that Pictilisib inhibited proliferation of PC3 with an IC_{50} of 0.28 uM. Similarly, HT29 treated with Nintedanib is also placed in this subnetwork based on Sagittarius's simulated profile, representing another unique drug and unique cell line for the sensitive subnetwork. Nintedanib was also found to have inhibited proliferation in the HT29 cell line with an IC_{50} of 1.40 uM,[53] and was shown to have significant antitumor activity in HT29 mouse xenograft models.[53,54] This implies that Sagittarius can simulate perturbation experiments to identify candidate drug repurposing targets across cell lines, cancer types, and therapeutic compounds, creating new opportunities for inexpensive and unbiased drug screening as an initial step in the precision medicine pipeline.

Augmenting drug-induced expression improves drug response prediction

Given its drug repurposing potential, we next systematically evaluated Sagittarius on two large-scale cell line drug response prediction datasets, GDSC and the Cancer Therapeutic Response Portal (CTRP) dataset.[34] Drug-induced expression profiles have been useful for drug response prediction,[55] but are expensive to measure compared to basal cell line expression, making Sagittarius's simulated drug-induced profiles especially valuable. We constructed a fully connected neural network model to predict the GDSC IC_{50} label for all drug perturbations on all cell lines. We compared one version of this model trained on perturbation experiment data from Sagittarius to another version trained on the experimentally measured LINCS drug-induced expression data for perturbation combinations that also appeared in the LINCS dataset. The Sagittarius-backed network achieved an average Spearman correlation of 0.46 per cell line, compared to 0.004 for the experimentally-measured data version (**Fig. 5b**). In this case, the model trained only on the experimentally measured dataset had such poor performance largely because the dataset, which was the intersection of cancer drug and cell line perturbation experiments contained in both LINCS and GDSC, was extremely small, while Sagittarius was able to simulate datapoints that were not present in the measured dataset. Sagittarius most markedly improved drug response prediction for the NSCLC cell line A549, which is the second most frequently measured LINCS cell line, and for the drugs Bosutinib, Selumetinib, Vismodegib, and Olaparib, which are among the most frequently measured drugs in the LINCS dataset (**Supplementary Fig. 5**). This shows that Sagittarius can take advantage of the many perturbation experiments to inform better predictions for each drug and cell line, even when applied to unmeasured or sparsely measured combinations.

We then repeated the experiment using drug sensitivity labels from CTRP. The model trained with Sagittarius's data had an average Spearman correlation of 0.52 per cell line, a 13.0% improvement over model trained only with the available experimentally measured data (**Fig. 5c**). The data from Sagittarius again had the largest benefit for the NSCLC cell line A549, as well as the prostate cancer cell line PC3, and for the drugs Neratinib and GSK-461364, which again are frequently measured in the LINCS dataset, although it struggled with the HER2-positive breast cancer cell line SKBR3, which is less frequently measured (**Supplementary Fig. 6**). For both GDSC and CTRP, Sagittarius was able to learn relationships between dose response curves for different drug and cell line perturbations to predict other experiments' treatment sensitivities, confirming its ability to accurately predict drug response for new cancer drugs and cell lines.

Improved cancer-essential gene prediction using drug-induced expression

In addition to drug response analysis, we also considered whether Sagittarius could predict cancer gene essentiality using drug-induced expression from the LINCS dataset. We used the Cancer Dependency Map (DepMap) dataset as labels for gene essentiality, independently considering both the DEMETER version,[36] which uses short hairpin RNAs (shRNAs) to identify the genes most crucial for cell viability and proliferation, and the CERES version,[35] which uses CRISPR-Cas9 essentiality screens to estimate gene dependency. We created a restricted dataset containing each cancer cell line and gene pair in the DepMap dataset for which the cell line was also present in the LINCS dataset. Then, for each cell essentiality entry in the restricted dataset, we found a candidate LINCS drug whose target matched the knocked-out gene in the cell essentiality pair, hypothesizing that the drug's inhibitory effect on a cell line is related to the cell line's dependency on the target gene.[56] Using data from Sagittarius, we assigned each cell line and gene pair in our restricted dataset to an inferred treatment vector for the cell line and candidate drug. We trained a neural network regression model on this dataset to predict DepMap's cell essentiality score for the drug's target gene. We evaluated the benefit of Sagittarius's simulated data by comparing this model to a neural network regression model trained on experimentally measured LINCS data for DepMap pairs where a candidate drug existed for the given cell line and gene.

The model trained using Sagittarius's simulated data obtained a 0.789 average Spearman correlation between the predicted and DEMETER gene essentiality scores for each cell line, relative to 0.278 for the model trained only on experimentally available data (**Fig. 5d**). The Sagittarius-backed model also had an average cell line Spearman correlation of 0.816 for the CERES dataset, relative to 0.261 model trained directly on the measured LINCS data (**Fig. 5e**). The Sagittarius data particularly improved both DEMETER and CERES predictions for well-measured LINCS cell lines, such as the THP1 leukemia cell line and YAPC pancreatic cell lines (**Supplementary Fig. 7**), confirming Sagittarius's ability to simulate drug response data that can identify the therapeutic potential of both a compound and a drug target gene for diverse cancer types. We attribute the strong performance across many different cancer types and drugs to the shared reference space, where dose- and treatment-time response can be compared across cancer cell lines and compounds. We therefore believe that Sagittarius's transcriptomic profile simulation can bring benefit to future studies towards understanding the molecular basis and mechanisms of cancer drug response.

Simulating mutation profiles for early-stage cancer patients

Having extrapolated transcriptomic time series in one- and two-continuous dimensions, we then sought to apply Sagittarius to cancer survival time data, this time aiming to simulate somatic

mutation profiles, rather than gene expression profiles, for cancer patients. It remains very challenging to measure genomic profiles from patients with nascent tumors, as they are rarely diagnosed at this stage, and yet these initial mutations can be the most informative as to the cancer's mechanisms and potential early-intervention therapies before other passenger mutations accumulate.[57] Nevertheless, measuring genomic data at scale, and particularly data from biopsied tumor tissue, remains a significant challenge for nascent cancers.

We propose a novel problem formulation where we model a cancer type as a sequence of patients, ordered by their survival time. In particular, we are interested in extrapolation to later time points in the sequence, indicating longer patient survival times, because these represent the mutation profiles of nascent tumors that are often very difficult to measure experimentally because they have not yet been diagnosed (**Fig. 6a**).[58] We used The Cancer Genome Atlas (TCGA) dataset[37] of gene mutation profiles for cancer patients from 24 cancer types. Although this formulation can help us extrapolate, it uses one time series for the entire cancer type. Therefore, it does not represent the heterogeneity within a cancer type.[59,60] This problem is more severe when some patients in the sequence have censored survival times, resulting in a time point label in the sequence that is potentially very different from the patient's actual survival time. To mitigate this issue, we propose a method to remove a patient from the cancer type sequence if their mutation profile is very different from other patients with a similar survival time. In particular, we trained a neural network to predict a patient's survival time given their initial mutation profile, and define similarity by the gradient of the network's loss. We then considered the trained model loss per patient and retained only the censored patients with an individual loss comparable to the most challenging observed patients in that cancer type (see **Methods**).[61,62] For the sarcoma (SARC) cancer type, this led to the inclusion of 31 patients with a censored death event, expanding the SARC time series to 115 patients (**Fig. 6b**). After this filtering step, we considered all remaining censored patients' final follow-up time to be the same as their death event time. Therefore, the remaining patients in each cancer type sequence are more similar, and represent the majority component of the cohort. We then divided the time series for a single cancer type based on patient survival into a train and test split (**Fig. 6a** and **Methods**), and evaluated the average mutation area under the receiver operating characteristic (AUROC) for the test patients.

We focused on the SARC and thyroid carcinoma (THCA) cancer types as case studies, restricting the number of mutated genes we evaluated on to those most variable over time in more than one cancer type (see **Methods)**. In the THCA case study, Sagittarius had an average test set AUROC of 0.72, a 49.0% improvement over a mean model trained solely on the observed THCA data (**Fig. 6c**). In the SARC case study, Sagittarius had an average test set AUROC of 0.73, which was an 11% improvement over a mean model trained solely on the observed SARC data (**Fig. 6d**).

For the model trained with *k*=57 SARC training patients, Sagittarius particularly improved the mutation AUROC for a test patient with an overall survival time of 76 months (**Fig. 6e**). Of the evaluated genes, the patient's clinical mutation profile is positive for a mutation in *LRP1B*, which previous work has suggested leads to improved patient outcomes with immune checkpoint inhibitors (ICIs) in sarcoma.[63] The mean method predicts the most likely mutations for a patient with a 76-month survival time as *TP53*, *TTN*, *MUC16*, *DNAH5*, and *OBSCN*, reflecting the most common mutations for the SARC patients with more severe disease progression and shorter survival times, and assigns 0 probability to the *LRP1B* mutation. Sagittarius, on the other hand, leverages cancer survival information from other cancer types as well as patterns within the SARC training data to predict *TP53*, *TTN*, *RYR2*, *LRP1B*, and *ADGRV1* as the most likely mutations. *ADGRV1* has been found to be mutated in approximately 45% of skin cutaneous melanomas.[64] Furthermore, the correct inclusion of *LRP1B* in this list of likely-mutated genes indicates that Sagittarius may have learned that patients with an *LRP1B* mutation are associated with good ICI response for multiple cancer types,[63] and can translate that knowledge to the SARC patient.

Tumorigenesis in the Hedgehog signaling pathway by simulating early-stage sarcoma mutation profiles

Having confirmed our ability to simulate mutation profiles for sarcoma patients with longer survival times, we retrained Sagittarius on our entire filtered dataset and then simulated gene mutation profiles for 27 early-stage sarcoma patients (**Supplementary Fig. 10**). On average, Sagittarius predicted that the most-likely mutations were in *DNAH17*, *PREX1*, *EGFLAM*, *FAM47B*, *DSEL*, *ARID2*, *TRPM1*, *NLGN1*, *PTCH1*, and *MYCBP2*.

We found that many of these genes are related to the Hedgehog (HH) signaling pathway and improper activation of the *GLI* oncogene (**Fig. 6f**), which some previous studies have connected to improved survival outcomes in sarcoma patients.[65,66] *PTCH1*, which has also been connected to plexiform fibromyxoma,[67] basal cell carcinoma,[68] and medulloblastoma,[69] is a tumor suppressor gene in the HH pathway, and loss-of-function mutations in *PTCH1* can lead to aberrant activation of the HH pathway and consequent tumorigenesis.[69] Furthermore, studies have found that the *MYC* oncogene directly regulates *GLI1* expression in Burkitt lymphoma cell lines,[70] while the *MYCBP2* gene promotes *MYC* degradation,[71] and lymphoblastic leukemia patients have been found to have both high *c-MYC* expression and low *MYCBP2* expression.[72] Similarly, the *ARID2* gene directly interacts with *GLI1*[73] as a cord subunit of the SWI/SNF chromatin remodeling complex.[73,74] In addition, the *PREX1* gene is a member of the PI3K-Akt signaling pathway, which has been associated with *GLI* code regulation[75] and cross-talk with the HH pathway in melanoma.[68,76] The *DNAH17* gene encodes a protein that makes up a subunit of the primary

cilium's basic structure;[77] in turn, the primary cilia are both positive and negative effectors of the HH signaling pathway.[77,78] In addition to these molecular connections to the *GLI* oncogene, we found that the *EGFLAM* gene has been shown to induce activation of the PI3K-Akt signaling pathway[79] containing *PREX1*. Previous studies have also found that *NLGN1* was significantly enriched with the HH pathway in a study of colorectal carcinoma.[80]

We were therefore able to identify multiple connections to the HH signaling pathway in Sagittarius's simulated early-stage sarcoma mutation profiles, and we connected the most likely predicted mutations with recent work in sarcoma studies. Sagittarius's reasonable simulated profiles indicate that the mutational patterns from other TCGA cancer types with more early-stage measurements in the shared reference space, combined with the sarcoma-specific patterns learned by the model's nonlinear mapping from the latent space, point to the HH signaling pathway and particularly the hyperactivation of the *GLI* oncogene as potentially significant sources of tumorigenesis in sarcomas.

Discussion

Sagittarius enables simulation of extrapolated gene expression profiles from sparse, heterogeneous experimental datasets without requiring aligned time points or batch correction between different experimental conditions. By augmenting the measured data with our simulated data, we are able to trace shared lineages between organs in a germ layer in mouse development. We can also suggest new therapeutic compounds to treat cancer cell lines by comparing simulated drug-induced expression profiles from diverse experiments, which are not limited to sharing a cell line or therapeutic compound with a known successful therapy. Finally, we can simulate early-stage cancer patients' mutation profiles to identify potential tumorigenesis drivers in sarcoma.

Although Sagittarius can extrapolate to new time points, the model still struggles when the developmental time point of interest is outside of the range of any seen developmental stages measured in the training data. We identify this limitation in the Evo-devo dataset extrapolation task, for instance, where the model performs worst on human extrapolation compared to all other species because the time points to simulate come from aging and senescent organs, while the latest measured developmental time points correspond to earlier development.[12] Future work could combine Sagittarius with time series forecasting work to improve extrapolation beyond the measured developmental age range. Similarly, Sagittarius is unable to extrapolate at precise time points. The shared reference space, while enabling transfer between heterogeneous time series without requiring alignment, warps the queried and measured time points to align with

biological age. This enables an understanding of the relative trajectory over time, but does not correspond to exact time points outside of the training set. More aggressive regularization and optimization techniques enforcing the absolute difference in measured age as well as biological age could improve this. Furthermore, Sagittarius models a single time series per experimental condition in the reference space, potentially obscuring some heterogeneity within the condition itself, as we note with our cancer type time series formulation. In future work, we could introduce a hierarchical time series component to Sagittarius, explicitly modeling the heterogeneity in a single measured condition.

Sagittarius is inspired by decades of work in modeling cell dynamics, including the recent works PRESCIENT[81] and pseudodynamics.[82] The key difference between Sagittarius and these works is that their diffusion processes specifically model cell-level lineage tracing and do not extend to genomic profile simulation. Sagittarius, on the other hand, learns a shared trajectory in the common reference space and explicitly simulates expression or mutation profiles to augment measured datasets and improve downstream analyses.

Figure legend

Fig. 1 Sagittarius model overview. **a,** Sagittarius is useful in settings with many diverse time series measurements, such as developmental gene expression data across species and organs, many combinations of which are unmeasured. The measurements in each time series are also sparse and unaligned. **b,** For each time series, Sagittarius computes a conditional high-frequency embedding of the measured time points and a conditional embedding of the gene expression measurements at each time point based on the species and organ. It then uses a continuous, multi-head attention transformer to map the embedded time points and expression vectors to the reference space. **c,** The continuous transformer takes each pair of species- and organ-conditioned time and expression embeddings and learns a mapping to the regular reference space, translating from measured age to a shared biological age. **d,** Users can request simulated expression vectors from Sagittarius, such as the expression profile of a human 2-year-old heart that has not been measured in the original dataset (**a**). Sagittarius maps the request from the regular reference space back to the data space to simulate the unmeasured profile.

Fig. 2 Gene expression simulation for extrapolated time points in later-stage development. **a-f,** Bar plots comparing the performance of Sagittarius and existing approaches when extrapolating to the four latest time points in the Evo-devo dataset. Test sequences are subdivided by species (**a-c**) and by organ (**d-f**). For Pearson correlation, comparing genes (**a,d**) or comparing time points (**b,e**), higher correlations indicate better performance; for RMSE (**c,f**), lower error indicates better performance. The * indicates that Sagittarius outperforms the next-best-performing model in the

metric, with significance levels of t-test p-value < 5e-2 for *, t-test p-value < 5e-3 for **, and t-test p-value < 5e-4 for ***. All t-tests are one-sided, and we use the Fisher transform for the correlation metrics to transform the values to a normal distribution.

Fig. 3 Mouse transcriptomic velocity across organs. **a,b,** UMAP plots showing simulated mouse gene expression from E5.5 to P63 for 7 organs, colored by organ (**a**) and time (**b**). The arrows in (**a**) indicate the transcriptomic velocity of each organ. **c,** Bar plot comparing the simulated expression of *Xrn2* at early development (E5.5-E8) to young mouse (P8-P63) across five organs. *Xrn2* expression is not statistically different between the brain, kidney, liver, ovary, and testis organs at the early development (ANOVA p-value > 0.05), but differs between organs at the young mouse time range, particularly with lower expression levels in the liver relative to other organs (ANOVA p-value < 1e-98). **d,** Bar plot examining the consistency of gene expression temporal patterns between simulated data and scRNA-seq data for *Egflam*, *Smoc1*, *Slc6a2*, and *Rpl38* in different tissues over time. Better predictions are closer to the Tabula Muris Senis dataset correlations for cell types within each tissue that are summarized by the boxes, while the star shows the Pearson correlation from Sagittarius's simulated correlation for aging mouse tissues (140 time points beginning at P14), and the diamond shows the correlation from mouse organs measurements in the Evo-devo dataset.

Fig. 4 Drug-induced gene expression simulation at unmeasured experimental combinations, doses, and times. **a,** The LINCS pharmacogenomic dataset contains gene expression measurements from a set of experiments where a cancer cell line is treated with a therapeutic compound. The set of measured cell lines and compounds is sparse, with less than 1.77% of possible experiments measured. The measured experiments are also only measured at select dose and treatment times, and the entire dataset includes a limited number of dose and treatment times. **b,** Illustration of the three extrapolation tasks we evaluate for the LINCS dataset: complete generation, where we simulate an unmeasured cell line and compound experiment at both a dose and time that are unmeasured by any experiment in the dataset; combination & dose, where we simulate an unmeasured cell line and compound experiment at a time that has been measured in the dataset but a dose that is unmeasured by all experiments; and combination & time, where we simulate an unmeasured cell line and compound experiment at a dose that has been measured in the dataset but a time that is unmeasured by all experiments. **c-e,** Scatter plots comparing the average Spearman correlation of simulated test combinations from Sagittarius and the existing cVAE model for each test drug on the complete generation (**c**), combination & dosage (**d**), and combination & time (**e**) extrapolation tasks.

Fig. 5 Drug and cell line treatment efficacy simulation analysis. **a,** kNN network where each node represents a drug and cell line combination, with edges between the most similar drug-induced

574 expression effect. The four communities in the graph are shown in different colors and labeled
575 according to the average GDSC-measured IC_{50} dose of that community, measured in uM. The
576 inset shows a connected 8-node subgraph from the sensitive community, made up of the NSCLC
577 cell line A549 treated with Selumetinib, Gefitinib, and Vorniostat; the breast cancer cell line MCF7
578 treated with Gefitinib, MK-2206, and Palbociclib; the prostate carcinoma cell line PC3 treated with
579 Pictilisib; and the colorectal adenocarcinoma cell line HT29, treated with Ninetedanib. **b,c,** Bar
580 plot (**b**) and scatter plot (**c**) of Spearman correlation between predicted and GDSC-measured (**b**)
581 or CTRP-measured (**c**) IC_{50} doses per cell line, comparing a neural network trained with imputed
582 data from Sagittarius and a neural network trained without any imputed data. Points above the
583 $y = x$ line are cell lines where Sagittarius's imputed dataset improved the downstream prediction
584 accuracy. **d,e,** Scatter plot of Spearman correlation between predicted and DepMap-measured
585 cancer gene essentiality scores for each cancer line, with the DEMETER (**d**) and CERES (**e**)
586 DepMap versions. All points are above the $y = x$ line, meaning Sagittarius improved downstream
587 gene essentiality prediction performance for all cell lines on both DepMap versions.
588
589 **Fig. 6** Early cancer patient mutation profile simulation. **a,** Illustration of the training and testing
590 splits for a given cancer type in the TCGA extrapolation task, where training patients have the
591 shortest survival times and test patients have longer survival times for that cancer type . **b,** Violin
592 plot of the survival time regression model's absolute error per patient for the SARC cancer type,
593 divided according to the patient's censoring label. We remove all patients with a loss above the
594 dashed line from the dataset, and train Sagittarius only on the patients below the dashed line. **c,d,**
595 Plot of the average simulated mutation profile AUROC for each of the THCA (**c**) and SARC (**d**)
596 cancer type test splits, ordered according to the shortest survival time in that test split. **e,** Scatter
597 plot comparing the per-patient simulated mutation profile AUROC from Sagittarius and the
598 mean comparison approach for the SARC test split including patients with an observed death
599 event more than 37 months after diagnosis. Points above the $y = x$ line indicate that Sagittarius
600 had a better simulated mutation profile than the comparison approach. **f,** Illustration of the ties
601 between the *GLI* oncogene in the Hedgehog (HH) signaling pathway and the *PTCH1*, *PREX1*,
602 *MYCBP2*, *ARID2*, and *DNAH17* genes that Sagittarius predicted as among the most likely
603 mutations in early-stage sarcoma patients.
604
605 **Supplementary Fig. 1** Gene expression simulation performance summary statistics for Evo-devo
606 extrapolation task. **a-c,** Bar plot of Pearson correlation comparing genes (**a**), Pearson correlation
607 comparing time points (**b**), and RMSE (**c**) of the simulated expression profile and measured
608 expression profile when extrapolating to the last four measured timepoints from each species and
609 organ combination in the Evo-devo dataset for Sagittarius and the comparison approaches. For
610 Pearson correlation, comparing genes or comparing time points (**a**,**b**), higher values indicate

better performance; for RMSE (**c**), lower values indicate better performance. Error bars indicate standard error.

Supplementary Fig. 2 Mouse transcriptomic velocity across organs. **a,b,** PCA plot showing simulated mouse gene expression from E5.5 to P63 for 7 organs, colored by organ (**a**) and time (**b**). The arrows in (**a**) indicate the transcriptomic velocity of each organ. The first PC shows most variation with respect to time, while the second shows most variation with respect to organ. Organ annotations in (**a**) are added to help differentiate between organs, especially in the case of overplotting.

Supplementary Fig. 3 Mouse gene expression simulation performance for Evo-devo extrapolation task. **a-c,** Bar plot comparing Sagittarius and existing approaches in terms of Pearson correlation comparing genes (**a**), Pearson correlation comparing time points (**b**), and RMSE (**c**) of the simulated mouse expression profile and measured mouse expression profile of each organ when extrapolating to the final four measured sequence time points in the Evo-devo dataset. For Pearson correlation, comparing genes or comparing time points (**a,b**), higher values indicate better performance; for RMSE (**c**), lower values indicate better performance.

Supplementary Fig. 4 Time series measured in the restricted LINCS dataset. **a,** Heatmap indicating the drug and cell line combinations that have time series measurements included in the LINCS dataset we use after initial processing. Cell lines tend to be either relatively well-measured or very sparsely measured. **b,** Histogram of the sequence lengths for all measured drug and cell line combinations. The length of the sequence is the number of unique dose and treatment time combinations that the therapeutic combination is measured at.

Supplementary Fig. 5 LINCS measurements with the best-performing cell line and drugs for the IC_{50} prediction task with the GDSC dataset. **a,b,** Bar plot of the number of measured drug treatments per cell line (**a**) and cell lines treated per drug (**b**) in the LINCS dataset. The A549 cell line is highlighted as the cell line with the most-improved predictions from Sagittarius's imputed dataset (**a**). The drugs with the most-improved predictions, Selumetinib, Bosutinib, Olaparib, and Vismodegib, are also highlighted (**b**).

Supplementary Fig. 6 LINCS measurements with the best-performing cell line and drugs for the IC_{50} prediction task with the CTRP dataset. **a,b,** Bar plot of the number of measured drug treatments per cell line (**a**) and cell lines treated per drug (**b**) in the LINCS dataset. A549 and PC3, the cell lines for which Sagittarius's simulated data most improves the predictions, are highlighted. SKBR3, which Sagittarius struggles on, is also highlighted (**a**). GSK-461364 and Neratinib are highlighted as the most-improved drugs with Sagittarius's imputed dataset (**b**).

Supplementary Fig. 7 LINCS measurements with the best-performing cell lines for the gene essentiality prediction task with the DEMETER and CERES DepMap datasets. Bar plot of the number of drug treatment experiments measured in the LINCS dataset per cell line. Sagittarius's imputed dataset provided the most benefit are A549, MDAMB231, THP1, HS578T, SKBR3, YAPC, VCAP, OCILY19, and U2OS, which are highlighted.

Supplementary Fig. 8 Distribution of TCGA patients per cancer type. Comparison of patient counts if all patients are used in the analysis, patient counts if only retained patients (including all observed patients and some censored patients) are used in the analysis, and patient counts if only observed patients are used in the analysis. By construction, the number of total patients is larger than the number of retained patients, which is in turn at least as large as the number of observed patients. Retaining some censored patients according to the individual survival prediction loss could improve model power without corrupting the time series formulation.

Supplementary Fig. 9 THCA censored patient analysis. Violin plot of the survival regressor's absolute error for each THCA patient, subdivided into an observed group and a censored group.

Supplementary Fig. 10 SARC training and extrapolation time point distribution. Histogram showing the measured survival time of patients in the SARC time series as the available sarcoma training data and the extrapolation time points used to simulate the expression profile of an early-stage sarcoma patient.

Supplementary Fig. 11 Normalized mutation rate and survival time for Sagittarius's predicted early-stage sarcoma mutations. **a-j,** Bar plot of the Spearman correlation of survival time and a patient's mutation normalized by their total mutation load for the top-10 predicted mutations in simulated early-stage sarcoma patients, *DNAH17* (**a**), *PREX1* (**b**), *EGFLAM* (**c**), *FAM47B* (**d**), *DSEL* (**e**), *ARID2* (**f**), *TRPM1* (**g**), *NLGN1* (**h**), *PTCH1* (**i**), and *MYCBP2* (**j**). We show the Spearman correlation for each cancer type where at least two patients in the time series have a mutation in the gene.

Supplementary Fig. 12 Normalized mutation rate and survival time for *GLI* mutations. **a,b,** Bar plot of the Spearman correlation of survival time and a patient's mutation normalized by their total mutation load for the *GLI2* (**a**) and *GLI3* (**b**) genes, which are transcription factors in the Hedgehog signaling pathway. We show the Spearman correlation for each cancer type where at least two patients in the time series have a mutation in the gene.

Supplementary Fig. 13 Mutation frequency across cancer types for Sagittarius's predicted early-stage sarcoma mutations. **a-j,** Bar plot of the percentage of patients in each cancer type with a mutation in the *DNAH17* (**a**), *PREX1* (**b**), *EGFLAM* (**c**), *FAM47B* (**d**), *DSEL* (**e**), *ARID2* (**f**), *TRPM1* (**g**), *NLGN1* (**h**), *PTCH1* (**i**), and *MYCBP2* (**j**) genes. We show a percentage for each cancer type where at least one patient in the time series has a mutation in the gene.

Supplementary Fig. 14 Mutation frequency across cancer types for *GLI* mutations. **a,b,** Bar plot of the percentage of patients in each cancer type with a mutation in the *GLI2* (**a**) and *GLI3* (**b**) genes, which are transcription factors in the Hedgehog signaling pathway. We show a percentage for each cancer type where at least one patient in the time series has a mutation in the gene.

Methods

We define the input heterogeneous time-series dataset as $\mathcal{D} = \{(x_i, y_i, t_i)\}_{i=1}^{N}$. The $x_i \in \mathbb{R}^{T \times M}$ are the measured time series input for sequence i, where each measurement is M-dimensional and the time series is measured at T timepoints; $y_i \in \{1, \ldots, C_j\}^C$ are the C experimental variables for time series i, with $y_{i,j} \in \{1, \ldots, C_j\}$ for C_j possible values for the jth experimental variable; $t_i \in \mathbb{R}^{B \times T}$ are the B continuous variables for time series i, with $t_{i,j}[r]$ denoting the value of the jth continuous variable associated with the $r$the measurement of time series i, $x_i[r]$. In particular, $B = 1$ in the Evo-Devo[12] and TCGA[37] studies, while $B = 2$ in the LINCS[15] study, where we model both dose and time. We further assume that $(x_i, y_i, t_i) \sim \boldsymbol{\mathcal{X}}$, where $\boldsymbol{\mathcal{X}}$ is the space of all possible measurements. Sagittarius simulates a sample $(\hat{x}|y, t) \sim \boldsymbol{\mathcal{X}}$ for a user-specified combination of experimental and continuous variables that may not be measured in dataset $\mathcal{D}$.

As a first step, Sagittarius embeds the individual measured datapoints $x_i[r]$ into a low-dimensional generative space, conditioned on the associated experimental variables. Formally, we sample from the learned Gaussian space according to

$$\mu_i[r], \sigma_i[r] = q_\xi(x_i[r], y_i) \qquad z_i[r] \sim \mathcal{N}(\mu_i[r], \sigma_i[r]),$$

where $z_i[r] \in \mathbb{R}^d$ with $d \ll M$. For brevity, we often write these two steps jointly as $z(x_i, y_i)[r]$. We regularize this learned Gaussian space by imposing the standard-normal prior, $p(z) = \mathcal{N}(0, I)$.

The second component of the model is a continuous transformer. In order to map time series to the shared reference space, the user defines both a temporal basis range $\left(\theta_j^{(0)}, \theta_j^{(1)}\right)$ for each of the $j = \{1, \ldots, B\}$ continuous variables, as well as $S + 1$, which defines the number of time points in the reference space. To learn robust and compact representations from the input time series, we choose S such that $S + 1 < T$. Given these parameters, Sagittarius defines the fixed temporal grid

$$t_{ref,j} \in \mathbb{R}^{S+1}: t_{ref,j}[\tau] = \theta_j^{(0)} + \tau \frac{\theta_j^{(1)} - \theta_j^{(0)}}{S}$$

for the τth reference point $\tau \in \{0, \cdots, S\}$. We further define the continuous attention embedding function

$$\psi_{h,b}^{enc}(t_{i,b}[r])[v] = sin\left(\omega_{h,b,v}^{enc} t_{i,b}[r] + \alpha_{h,b,v}^{enc}\right)$$

for the bth continuous variable at dimension v of the continuous time embedding for each attention head h, where the continuous embedding dimension V and number of attention heads H are model hyperparameters, and ω and α are fixed scaling and shifting terms. We further combine the embeddings for each of the continuous variables to the complete continuous embeddings

$$\psi_h^{enc}(t_i[r]) = \oplus_{b=1}^{B} \psi_{h,b}^{enc}(t_{i,b}[r]) \qquad \psi_h^{enc}(t_{ref}[\tau]) = \oplus_{b=1}^{B} \psi_{h,b}^{enc}(t_{ref,b}[\tau]),$$

where $\oplus$ indicates vector concatenation.

In the transformer model framework, we define the hth attention head's key for time series i and the regular space's query as

$$k_{h,i}^{enc}[r] = f_{h,v}^{enc}(y_i, \psi_h^{enc}(t_i[r])) \qquad q_h^{enc}[\tau] = g_{h,v}^{enc}\left(\psi_h^{enc}(t_{ref}[\tau])\right),$$

where both $k_{h,i}^{enc}[r]$ and $q_h^{enc}[\tau]$ are d_k-dimensional vectors. We project the embeddings of the measured time series embeddings $z(x_i, y_i)$ to Sagittarius's regular reference space according to

$$z_{ref}(x_i, y_i, t_i)[\tau] = \sum_{h=1}^{H} \sum_{r=1}^{T} z(x_i, y_i)[r] \frac{exp(\langle k_{h,i}^{enc}[r], q_h^{enc}[\tau] \rangle / \sqrt{d_k})}{\sum_{r'=1}^{T} exp(\langle k_{h,i}^{enc}[r'], q_h^{enc}[\tau] \rangle / \sqrt{d_k})},$$

producing the embeddings $z_{ref}(x_i, y_i, t_i)$ in the regular reference space for each of the $S + 1$ values of τ.

The decoder layer of our continuous transformer follows a very similar framework, decoding from the regular reference space back to the time points of interest. Specifically, we let

$$\psi_{h,b}^{dec}(t_{j,b}[r])[v] = sin\left(\omega_{h,b,v}^{dec} t_{j,b}[r] + \alpha_{h,b,v}^{dec}\right)$$

and

$$\psi_h^{dec}(t_j[r]) = \oplus_{b=1}^{B} \psi_{h,b}^{dec}(t_{j,b}[r]) \qquad \psi_h^{dec}(t_{ref}[\tau]) = \oplus_{b=1}^{B} \psi_{h,b}^{dec}(t_{ref,b}[\tau]).$$

We further define

$$k_h^{dec}[\tau] = f_{h,v'}^{dec}\left(\psi_h^{dec}(t_{ref}[\tau])\right) \qquad q_{h,j}^{dec}[r] = g_{h,v'}^{dec}\left(y_j, \psi_h^{dec}(t_j[r])\right)$$

to be the decoding layer's key and query values, respectively. Finally, we convert from the regular time series in the reference space back to the irregular time series with

$$\hat{z}_j(x_i, y_i, t_i, y_j, t_j)[r] = \sum_{h=1}^{H} \sum_{\tau=0}^{S} z_{ref}(x_i, y_i, t_i)[\tau] \frac{exp\left(\langle k_h^{dec}[\tau], q_{h,j}^{dec}[r] \rangle / \sqrt{d_k}\right)}{\sum_{\tau'=0}^{S} exp\left(\langle k_h^{dec}[\tau'], q_{h,j}^{dec}[r] \rangle / \sqrt{d_k}\right)}.$$

If we take $j = i$, then this is equivalent to encoding to and from the same sequence; if we take $j \neq i$, then this encodes one sequence and decodes to another.

Finally, we convert our time series $\hat{z}_j(x_i, y_i, t_i, y_j, t_j)$ back from the latent embedding space to the data space, with

$$\hat{x}_j(x_i, y_i, t_i, y_j, t_j) = p_\theta(\hat{z}_j(x_i, y_i, t_i, y_j, t_j), y_j).$$

We train our model end-to-end with the loss function $\mathcal{L}(\xi, \nu, \upsilon, \nu', \upsilon', \theta)$, which we denote $\mathcal{L}_{Sag}(.)$ for brevity, as

$$\mathcal{L}_{Sag}(.) = \mathbb{E}_{(y_j, t_j)\in \boldsymbol{x}}\left[\mathbb{E}_{(x_i, y_i, t_i)\sim \mathcal{D}}\left[\mathbb{E}_{q_\xi(z_i|x_i, y_i)}\left[log\ p_\theta\left(\hat{x}_j|x_i, y_i, t_i, y_j, t_j\right) - \beta D_{KL}\left(q_\xi(z_i|x_i, y_i) \parallel p(z)\right)\right]\right]\right],$$

where D_{KL} denotes the Kullback-Leibler divergence and β is a regularization weighting hyperparameter.

During model training, we train on both the reconstruction setting ($j = i$) and a simulation setting ($j \neq i$). The specifics for each experiment are described in their respective sections. After model training, we simulate new observations for unseen combinations of experimental variables and at unmeasured time points. All we require is that, for each experimental variable $\widetilde{y_b}$, $\exists(x_i, y_i, t_i) \in \mathcal{D}$: $y_{i,b} = \tilde{y}_b$. We can then produce simulated data from a source time series i, which can either be randomly chosen from the available dataset or selected specifically for the generation task.

Evo-devo dataset processing

The Evo-devo dataset[12] contains gene expression vectors for 7 species and 7 organs measured at multiple pre- and post-natal time points. We first mapped all species' genes to their human orthologs using their provided Ensembl gene IDs and the python `pybiomart` package;[83] if no ortholog was found, we discarded that gene. We then took the intersection of all identified human orthologs for each species as our starting gene list. This identified 5,037 common orthologs across the 7 species. The observations for each species were given as strings, which were measured in different units according to the species. As a pre-processing step, we ordered the observed timepoint labels for each species and thereafter referred to that timepoint by its position in the corresponding species's ordered list to produce a common vocabulary. Finally, for the organ and species combination $y_i = [species_i, organ_i] \in \{1, \ldots, 7\}^2$, we took the indexed timepoint representations t_i for each measured gene expression profile from that experimental correlation to construct the time series x_i. We did this for each of the 48 species and organ combinations in the dataset to produce $\mathcal{D}_{evo-devo} = \{(x_i, y_i, t_i)\}_{i=1}^{48}$.

Existing models and Evo-devo training

For all models that required hyperparameter optimization, we randomly selected 20% of the measured data to use as a validation set. We did model hyperparameter selection on one model initialization that we then used for all later initializations; we used the validation set for training termination on all model initializations. For each model, we stopped training when the validation loss had not dropped for 250 epochs and saved the model parameters with lowest validation loss.

Mean: The mean baseline model, which has no hyperparameters, simply simulates data as $\hat{x}_i[t] = \frac{1}{T}\sum_{r=1}^{T} x_i[r]$; that is, the predicted expression for each gene at any timepoint of interest t is the average of the gene expression across all measured timepoints.

Linear: The linear baseline model, which has no hyperparameters, first defines a weight

$$\lambda_{i,t} = 0 \text{ if } t < min(t_i);$$

$$\lambda_{i,t} = 1 \text{ if } t > max(t_i);$$

$$\lambda_{i,t} = \max_{r \in t_i:\, r \le t} \left(\min_{s \in t_i:\, s \ge t} \left(\frac{t-r}{s-r} \right) \right) \text{ otherwise.}$$

Then, the linear model simulates expression at time t as

$$\hat{x}_i[t] = \max_{r \in t_i:\, r \le t} (1 - \lambda_{i,t}) x_i[r] + \min_{s \in t_i:\, s \ge t} \lambda_{i,t} x_i[s].$$

Note that, in the extrapolation setting, the linear baseline therefore simulates a gene expression vector identical to the expression vector of the nearest temporal measurement.

Neural ODE: We learn a set of single-sequence neural ODE models[28] that take observations from a single (x_i, y_i, t_i) sequence. We train 48 such models, one for each species and organ combination. As the experimental conditions y_i are constant within a single sequence, we reduce the task inputs to (x, t). We computed an ODE for both the forward and backward direction of the sequence as

$$\tilde{x}_{\rightarrow}[r] = \max_{s \in t_i:\, s \le r} x[s] + \int_{t=s}^{r} f_\theta(x[t])dt$$

$$\tilde{x}_{\leftarrow}[r] = \min_{s \in t_i:\, s \ge r} x[s] + \int_{t=s}^{r} g_\phi(x[t])dt.$$

In the case where $\widetilde{x_{\rightarrow}}$ or $\widetilde{x_{\leftarrow}}$ requires extrapolation (i.e., there is no such s to satisfy the constraint), we set $\widetilde{x_{\rightarrow}}[r] = \tilde{x}_{\leftarrow}[r]$. In order to empirically compute the integrals we used a step size of $\Delta_t = 0.1$ and the python `torchdiffeq` package.[28,30] We parameterized $f_\theta(.)$ and $g_\phi(.)$ using a mutli-layer perceptron (MLP) with two hidden layers of the same size as the input. Finally, we combine the forward and backward results to produce the final estimate

$$\hat{x}[r] = \frac{1}{2}(\tilde{x}_{\rightarrow}[r] + \tilde{x}_{\leftarrow}[r]).$$

We trained the model using the Adam optimizer[84] and a learning rate of 1e-3.

RNN: We learn a set of single-sequence bidirectional gated recurrent unit (GRU)[27] models to learn the dynamics for a single (x_i, y_i, t_i) sequence, again reducing the problem input to (x, t). We defined a time step $\Delta_t = 1$ between observations of interest. At each time point, we computed $z_r = q_\phi(x[r])$ for an MLP $q_\phi(.)$ as the embedding for each observation in the time series, and computed

$\hat{z}_t^{\rightarrow} = q_\phi\left(p_\theta(h_t^{\rightarrow})\right)$ $\quad$ $h_{t+1}^{\rightarrow} = g_\xi^{(gru\rightarrow)}(z_t, h_t^{\rightarrow})$ if $t \in t_i$; $h_{t+1}^{\rightarrow} = g_\xi^{(gru\rightarrow)}(\hat{z}_t^{\rightarrow}, h_t^{\rightarrow})$ otherwise,

where $h_0^{\rightarrow} = 0$. Similarly, we define the backward GRU as

$\hat{z}_t^{\leftarrow} = q_{\phi'}\left(p_{\theta'}(h_t^{\leftarrow})\right)$ $\quad$ $h_{t+1}^{\leftarrow} = g_{\xi'}^{(gru\leftarrow)}(z_t, h_t^{\leftarrow})$ if $t \in t_i$; $h_{t+1}^{\leftarrow} = g_{\xi'}^{(gru\leftarrow)}(\hat{z}_t^{\leftarrow}, h_t^{\leftarrow})$ otherwise,

with $h_T^{\leftarrow} = 0$. Finally, we combine the forward- and backward directions to produce the simulated gene expression profile

$$\hat{x}[r] = p_\theta\left(\frac{1}{2}(h_r^{\rightarrow} + h_r^{\leftarrow})\right)$$

for an MLP $p_\theta(.)$.

We used a embedding dimension of $z_r^{\rightarrow}, z_r^{\leftarrow}, h_t^{\rightarrow}, h_t^{\leftarrow} \in \mathbb{R}^{32}$, and used two hidden layers, each with 1024 hidden neurons, for $q_\phi(.)$ and $p_\theta(.)$. We trained the model end-to-end with the Adam optimizer[84] and a learning rate of 1e-3.

mTAN: We trained a discretized multi-time attention network (mTAN)[29] using the Adam optimizer[84] and a learning rate of 1e-3. As the mTAN module does not handle experimental variables, for each time series (x_i, y_i, t_i) the model received the reduced input (x_i, t_i). We used a latent embedding dimension of 32, a default temporal embedding dimension of 16, 8 attention heads, and 4 temporal reference points. The model learned the temporal embedding in the transformer's encoder, and fixed the temporal embedding in the transformer's decoder.

cVAE: We trained a conditional variational autoencoder (cVAE)[25] to learn $p(x_i[r] \mid y_i, t_i[r])$ for the Evo-devo dataset. We trained the model using the Adam optimizer[84] and a learning rate of 1e-3. We used a batch size of 128 gene expression profiles, since the model takes individual measurements as input rather than full time series. We used a model latent dimension of 32 with symmetric MLPs for the encoder and decoder. We tried both 2- and 3-hidden-layer networks, each hidden layer with 1024 hidden units. We also varied the β weight for the KL-divergence loss term with $\beta \in \{0.7, 1.0\}$. After a hyperparameter search, we selected the 2-hidden-layer encoder and decoder networks and set $\beta = 1.0$.

CPA: We trained a compositional perturbation autoencoder (CPA)[26] using an embedding dimension of 32 and batch size of 128. In the model, we considered the time to be independent of the organ label (the covariate) and dependent on the species label (the perturbation). We used a

patience of 5, autoencoder and temporal learning rate of 1e-3 and weight decay of 1e-7, and an adversary learning rate of 1e-5 and weight decay of 1e-10. We used an autoencoder width of 1024 units and tried an autoencoder depth of both 2- and 3 hidden layers. We used an adversary width of 16 and depth of 2, with 16 adversary steps. We also tried using both an MLP and a logarithmic sigmoid to represent the temporal curve. After hyperparameter search, we chose the 2-hidden-layer autoencoder and logarithmic sigmoid temporal curve.

Sagittarius: We used a latent space of dimension $d = 32$, a temporal range of interest $\left(\theta^{(0)}, \theta^{(1)}\right) = (0, 25)$, and a reference time series length $S + 1 = 4$. We chose $H = 8$ attention heads for our transformer layers, where the temporal embedding is 8-dimensional in both the encoder and decoder. We set the key- and query dimension $d_k = 32$ for the transformer. We used a batch size of 8 time series, and the Adam optimizer[84] with a learning rate of 1e-3. Finally, we used our batch size to set $\beta = 0.1667$ for our empirical loss $\mathcal{L}_{Sag}(.)$.

We used symmetric MLPs to learn $q_\xi(x_i, y_i)$ and $p_\theta\left(\hat{z}_j\left(x_i, y_i, t_i, y_j, t_j\right)\right)$ in Sagittarius's encoder and decoder respectively. We considered both 2- and 3-hidden-layer MLP architectures, with 1024 hidden units in each layer. We embedded the species and organ values for each time series into compact representations as an initial step in both the $q_\xi(.)$ and $p_\theta(.)$ networks; we considered either 2- or 8-dimensional embeddings for each of the species and organ labels. We also embedded the species and organ labels in the transformer encoder's key and decoder's query representations, and tried both 4- and 8-dimensional embeddings for each of the species and organ labels in the transformer. Using the validation set, we selected a 3-layer MLP for both $q_\xi(.)$ and $p_\theta(.)$, a 2-dimensional embedding for both the species and the organ labels in $q_\xi(.)$ and $p_\theta(.)$, and a 4-dimensional embedding for both the species and organ labels in the transformer modules.

During training, we used the reconstruction objective for each available time series, setting $j = i$ for $i = 1, \ldots, 48$ in $\mathcal{L}_{Sag}(.)$. We also included the following 4 simulation objectives during training.

1. Temporal generation: we randomly selected 12 time series from our training dataset. For each of these time series, we constructed a new training input $\bar{x_{i'}}$ where we masked out an additional three time points from the sequence. The masked time points were added as a partner training sequence $\widetilde{x_{j'}}$.
2. Same-species generation: we randomly selected 12 time series from our training dataset. For each of these, we appended them as new training points $\bar{x_{i'}}$ and randomly selected another training sequence that had the same species label but different organ label, which we added to our training data as the partner sequence $\widetilde{x_{j'}}$.

3. Same-organ generation: equivalent to the second, we randomly selected 12 time series $\bar{x_{i'}}$ from the training data and paired them each with a time series $\widetilde{x_{j'}}$ that shares the organ label but has a different species label.
4. Random generation: we randomly selected 12 time series $\bar{x_{i'}}$ and partner time series $\tilde{x}_{j'}$ from the dataset.

Sagittarius was then also trained a generation objective, formulated according to the empirical version of the loss term $\mathcal{L}_{Sag}(.)$, with $i = i'$ and $j = j'$.

Evo-devo quantitative extrapolation experiment

For the quantitative extrapolation experiments, we masked the latest four time points available for each time series in the Evo-devo dataset.[12] We then trained all models on the unmasked portion of the dataset. This resulted in 471 measurements to use for training or validation and 192 test measurements. At evaluation time, we used the models to predict the expression vectors on the masked time points and compared the simulated results from each model with the measurements in the dataset.

As an initial pre-processing step, we restricted the gene expression vector of the 5,037 orthologous genes in the dataset using the Augmented Dickey-Fuller (ADF) test, which tests for stationarity. We randomly selected one species and organ time series, which was the rabbit heart time series. Based on that combination, we retained the genes for which the ADF test failed to reject the null hypothesis that the gene was non-stationary over time and discarded all of the others, using a significance threshold of $p < 0.05$. This resulted in 4,533 retained genes.

Evo-devo dataset evaluation

To evaluate the simulated gene expression vectors, we considered three metrics: root mean squared error (RMSE), average Pearson correlation comparing genes, and average Pearson correlation comparing time points. Using T_i to denote the number of measurements for the ith time series, $x_i[t]$ to denote the Evo-Devo dataset's measurement for the ith time series at time point t, and $\hat{x}_i[t]$ to denote the model's simulated measurement for the ith time series at time point t, we defined the model's test RMSE per sequence as

$$RMSE_i = \sqrt{\frac{1}{4}\sum_{t=T_i-3}^{T_i}(\hat{x_i}[t] - x_i[t])^2},$$

with an overall model average test RMSE of

$$RMSE = \frac{1}{48}\sum_{i=1}^{48} RMSE_i.$$

To determine whether Sagittarius was statistically better than the comparison approaches in terms of RMSE, we used the one-sided paired t-test between Sagittarius's RMSE per sequence and the per-sequence RMSE of the comparison approach that performed best on average.

For the Pearson correlation (comparing genes), and using $\rho_{pearson}$ to denote the Pearson correlation computation, we defined the model's test correlation (comparing genes) per sequence as

$$\rho_i^{(genes)} = \frac{1}{4}\sum_{t=T_i-3}^{T_i} \rho_{pearson}(\hat{x}_i[t], x_i[t]),$$

with an overall model average test Pearson correlation (comparing genes) of

$$\rho^{(genes)} = \frac{1}{48}\sum_{i=1}^{48} \rho_i^{(genes)}.$$

To define the Pearson correlation (comparing time points), with $x_i[t,g]$ and $\hat{x}_i[t,g]$ used to denote expression of gene g at time point t from time series i from the Evo-Devo-measured and model-simulated gene expression respectively, we first defined

$$x_i[T_i-3{:}T_i, g] = [x_i[T_i-3, g], x_i[T_i-2, g], x_i[T_i-1, g], x_i[T_i, g]]$$

and

$$\hat{x}_i[T_i-3{:}T_i, g] = [\hat{x}_i[T_i-3, g], \hat{x}_i[T_i-2, g], \hat{x}_i[T_i-1, g], \hat{x}_i[T_i, g]].$$

We then defined the model's test correlation (comparing time points) per sequence as

$$\rho_i^{(times)} = \frac{1}{4533}\sum_{g=1}^{4533} \rho_{pearson}(\hat{x}_i[T_i-3{:}T_i, g], x_i[T_i-3{:}T_i, g]),$$

With an overall model average test Pearson correlation (ranked by time points) of

$$\rho^{(times)} = \frac{1}{48}\sum_{i=1}^{48} \rho_i^{(times)}.$$

To assess whether Sagittarius statistically outperformed the comparison approaches in terms of $\rho^{(genes)}$ and $\rho^{(times)}$, we first computed the Fisher z-transformation[38] of the correlation values, defined as

$$z(\rho) = \frac{1}{2} ln\left(\frac{1+\rho}{1-\rho}\right)$$

for some correlation ρ. Then, we used the one-sided paired t-test between Sagittarius's Fisher-transformed correlation per sequence and the per-sequence Fisher-transformed correlation of the comparison approach that performed best on average.

Mouse developmental analysis

We trained Sagittarius using the complete Evo-devo dataset $\mathcal{D} = \{(x_i, y_i, t_i)\}_{i=1}^{48}$. After training completed, we selected $\mathcal{D}_{mouse} = \{(x_j, y_j, t_j)\}_{j=1}^{7}$ where $y_{species,j} = mouse$. For each of the 7 organs in the dataset, we then used Sagittarius to simulate 10 gene expression time series. We generated the observations at ranked timepoints ranging from -5 to 13 with a granularity of 0.1,

resulting in $t^* = [-5.0, -4.9, \ldots, -0.1, 0.0, 0.1, \ldots, 12.9, 13.0\,]$. For each source sequence $(x_j\,, y_j\,, t_j)$ we then simulated the target sequence for $(y_j\,, t^\star)$ with

$$x_j^* = \hat{x}_j(x_i\,, y_i\,, t_i\,, y_j\,, t^*).$$

To further smooth the results, we then computed the moving average of x_j^* to produce

$$\widetilde{x_j^*}[t] = \frac{1}{5+10(5+t)} \sum_{r=0, r\in t^*}^{t+0.5} x_j^*[r] \;\text{ if } t < -4.5$$

$$\widetilde{x_j^*}[t] = \frac{1}{10} \sum_{r=t-0.5, r\in t^*}^{t+0.5} x_j^*[r] \;\text{ if } t \in [-4.5, 12.5]$$

$$\widetilde{x_j^*}[t] = \frac{1}{5+10(13-t)} \sum_{r=t-0.5, r\in t^*}^{13} x_j^*[r] \;\text{ if } t > 12.5,$$

for all $t \in t^*$. This resulted in 10 smoothed mouse gene expression time series samples for each organ in the dataset.

Transcriptomic velocity: Given the smoothed samples, we next computed the UMAP[41] embedding $\widetilde{z_j^*}[t] = UMAP\big(\widetilde{x_j^*}[t]\big)$ at each generated time point in each sample. We also computed the developmental velocity in the UMAP space as

$$v_j^* = \widetilde{x_j^*}[t + 0.1] \;-\; \widetilde{x_j^*}[t] \text{ if } t < 13.0 \qquad\qquad v_j^* = 0 \text{ if } t = 13$$

for all $t \in t^*$, and then further smoothed the results using moving average with a window size of 1, defined as

$$\widetilde{v_j^*}[-5] = v_j^*[-5]$$

$$\widetilde{v_j^*}[t] \;=\; \frac{1}{2}\big[v_j^*[t - 0.1] \;+\; v_j^*[t]\big] \text{ if } t \;\in [-4.9, 13].$$

We took the average (mean) of the velocities of each of our 10 samples to produce the unified organ velocity vector $\tilde{v}_j^*$. Finally, we normalized the velocity embeddings and, to decrease clutter in the plot, restricted our final result to integer time indices, such that

$$\hat{v}_j^*[t] = \frac{\tilde{v}_j^*[t]}{\big|\tilde{v}_j^*[t]\big|}$$

for all $t \in t^* \cap \mathbb{Z}$.

To produce the organ development plot, we projected the $\widetilde{z_j^*}$ to a grid, and defined the velocity at each grid point to be the average of the 100 velocity vectors $\hat{v}_j^?$ nearest to that grid point using `sklearn.neighbors.NearestNeighbors`,[85] weighted by their distance from the grid point. Finally, we discard the velocities with the 5% smallest magnitudes to simplify the plot. For our PCA[42] analysis of the same data (**Supplementary Fig. 2**), we repeated these steps, using $\widetilde{z_j^*}[t] = PCA\big(\widetilde{x_j^*}[t], nPCs = 2\big)$.

To identify genes that had a very similar expression at early developmental stages but differing expression levels in different organs at later developmental stages, we took

$$x_i^{(early)} = \widetilde{x_i^*}[t \in [-5.0, -2.5]\,] \qquad\qquad x_i^{(late)} = \widetilde{x_i^*}[t \in [11.5, 13.0]\,]$$

for each simulated, smoothed time series $\widetilde{x_i^*}$. Considering all 10 simulated sequences, this resulted in 250 early time points and 250 late time points per mouse organ. We then computed two ANOVA statistical tests with Bonferroni multiple testing correction, first comparing $x^{(early)}[gene\ m]$ from each organ to measure statistical similarity at early developmental stages and then comparing $x^{(late)}[gene\ m]$ to measure similarity at later stages.

Tabula muris gene evaluation: We first generated 10 mouse gene expression vectors for each organ at time points ranging from 11 to 25, with a granularity of 0.1. Given $t^{*} = [11.0, 11.1, \ldots, 24.9, 25.0]$, we simulated gene expression profiles and smoothed the results to produce $\tilde{x}_j^{*}$ as when producing $\tilde{x}_j^{*}$. We then computed the Spearman correlation over time for each of the genes based on the simulated data. We also computed the Spearman correlation over time for each gene based on the measured data in the Evo-devo dataset.[12] Finally, we took the heart and aorta, kidney, and liver tissue data from the Tabula Muris Senis droplet dataset,[16] which were the three tissues that aligned with the Evo-devo organs. For each cell type in the tissue data, we computed the average expression of that cell type at each of the measured timepoints, and then took the Spearman correlation of the average cell type expression over time.

LINCS dataset processing

We used the LINCS L1000 Platform level 3 pharmacogenomic dataset.[15] We restricted the data to drug and cell line combinations where the doses were measured in μM and then further restricted measurements to doses no more than 20 μM. After this processing step, we again restricted the dataset to include only the drug and cell line experiments, which we interpreted as "time series", that had more than 15 dose and time measurements. This resulted in 2,687 total time series for our dataset, each with between 16 and 78 measurements (**Supplementary Fig. 4**), where over 73% of the treatment combinations retained in the dataset had fewer than 25 measurements. Each measurement contains 978 genes. We represented this dataset as $y_i = [drug_i, cell\ line_i]$ and $t_i = [dose_i, time_i]$.

Existing models and LINCS training

We restricted our LINCS comparisons to the cVAE model,[25] which was the only existing model that could be applied to multiple continuous variables out-of-the-box. For Sagittarius and the cVAE model, we randomly partitioned the data into an 80% training, 10% validation, and 10% test split. We terminated model training when the validation loss had not decreased for at least 100 epochs and returned the model with lowest validation loss.

cVAE: We trained a cVAE[25] using the Adam optimizer[84] with a learning rate of 1e-3. We used a symmetric MLP encoder and decoder architecture, both with 2 hidden layers of 128 units each, a latent embedding dimension of 16, a KL-divergence weight $\beta = 1.0$, and a batch size of *1024*.

Sagittarius: We trained Sagittarius using the Adam optimizer[84] with a learning rate of 1e-3 and a batch size of 1024. We used an 8-dimensional vector to embed both the drug and the cell line as an initial input to Sagittarius's expression encoder and decoder, and two hidden layers with 128 neurons each to learn $q_\xi(.)$ and $p_\theta(.)$. We used a latent embedding dimension $d = 16$, 8 attention heads, and 16 temporal reference points (for both time and dose). We embedded the dose into an 8-dimensional vector and time into a 4-dimensional vector using the high-frequencing embeddings. For the transformer keys and queries, we used an 8-dimensional embedding for both drug and cell line. Finally, we used $\beta = 0.25$ for the KL-divergence weight in $\mathcal{L}_{Sag}(.)$.

During training, we used the reconstruction objective for each available time series, setting $j = i$in $\mathcal{L}_{Sag}(.)$. We also included the following 3 simulation objectives during training.

1. Generate drug: we randomly selected 32 drugs from the training dataset. For each drug, we identified two measured cell line combinations, and labeled one as $\bar{x_{i'}}$ and the other as its partner training sequence $\widetilde{x_{j'}}$.
2. Generate cell line: we randomly selected 32 cell lines from the training dataset. For each cell line, we identified two measured drug combinations, and labeled one as $\bar{x_{i'}}$ and the other as its partner training sequence $\widetilde{x_{j'}}$.
3. Random generation: we randomly selected 16 pairs of measured combinations from the training dataset, and labeled one as $\bar{x_{i'}}$ and the other as its partner training sequence $\widetilde{x_{j'}}$.

Sagittarius was then also trained a generation objective, formulated according to the empirical version of the loss term $\mathcal{L}_{Sag}(.)$, with $i = i'$ and $j = j'$.

LINCS quantitative simulation experiment

For the three different generation tasks we set for the LINCS dataset, we masked different combinations of experimental and continuous variables to create our test sets. For each, we first randomly selected 5 drug and cell line experimental combinations to remove from the training data, requiring that both the drug and the cell line appeared at least once somewhere else in the dataset.

1. **Complete generation:** For each of these experimental combinations, we also selected 3 non-zero doses and 1 non-zero time at random from each combination's measured time series to remove from all time series in the training dataset.

2. **Combination & Dose:** For each of these experimental combinations, we also randomly selected 3 non-zero doses from those time series to remove from all time series in the training dataset.
3. **Combination & Time:** For each of these experimental combinations we also randomly selected 1 non-zero time in the time series to remove from all time series in the training dataset.

At evaluation time, we aimed to simulate the time series for the masked experimental combinations, doses, and times. For the complete generation task, this resulted in 2144 training sequences with 7651 total measurements, 269 validation sequences with 924 total measurements, and 7441 test sequences with 15,068 total measurements; for the combination & dose task, this resulted in 2144 training sequences with 27,242 total measurements, 269 validation sequences with 3326 total measurements, and 7377 test sequences with 14,905 total measurements; and for the combination & time task, this resulted in 2144 training sequences with 10,417 total measurements, 269 validation sequences with 1202 total measurements, and 7395 test sequences with 14,966 total measurements. To evaluate the models' performance, we computed the Spearman correlation between the measured gene expression vectors that we had removed from the training data and the models' simulated gene expression vectors.

LINCS quantitative extrapolation experiment

We used Spearman correlation to assess model performance for the three LINCS generation tasks. Formally, let $\mathcal{Y}^h$ be the set of drug and cell line treatment combinations that are masked during training for each generation task; let $\mathcal{T}^h_{dose}$ be the set of doses that are masked during training (note that for the combination & time task, $\mathcal{T}^h_{dose} = \emptyset$); let $\mathcal{T}^h_{time}$ be the set of treatment times that are masked during training (so $\mathcal{T}^h_{time} = \emptyset$ for the combination & dose task). Then, define the measurement in the LINCS dataset for treatment combination i at dose t_{dose} and time t_{time} as $x_i[t_{dose}, t_{time}]$, and the model's simulated measurement for the same combination, dose, and time as $\hat{x}_i[t_{dose}, t_{time}]$. For the complete generation task, we computed the model's overall Spearman correlation as

$$\rho_{model} = \sum_{y \in \mathcal{Y}^h} \sum_{t_{dose} \in \mathcal{T}^h_{dose}} \sum_{t_{time} \in \mathcal{T}^h_{time}} \rho_{spearman}(\hat{x}_i[t_{dose}, t_{time}], x_i[t_{dose}, t_{time}]),$$

with $\rho_{spearman}$ denoting the Spearman correlation.

For the combination & dose and combination & time tasks, we computed the model's overall Spearman correlation as

$$\rho_{model} = \sum_{i=1}^{N} \sum_{t_{dose} \in \overrightarrow{t_i^{dose}}} \left[\sum_{t_{time} \in \overrightarrow{t_i^{time}}} 1\{(y_i \in \mathcal{Y}^h) \vee (t_{dose} \in \mathcal{T}_{dose}^h) \vee (t_{time} \in \mathcal{T}_{time}^h)\} \rho_{spearman}(\hat{x}_i[t_{dose}, t_{time}], x_i[t_{dose}, t_{time}]) \right],$$

where $1\{\cdot\}$ denotes an indicator variable.

Drug dosage similarity network

We first trained Sagittarius on the complete LINCS dataset,[15] not masking any datapoints. We then selected 78 random distinct doses d_{gen} at random from the dataset, sorting them from smallest to largest, and set the corresponding treatment time to be $t_{gen} = [6.0]^{78}$. The randomly-selected doses ranged from 8.33e-5 to 19.9998. For each drug and cell line experimental combination in the dataset we then computed

$$\hat{x}_i^{(gen)} = p_\theta\left(\hat{z}_i(x_i, y_i, t_i, y_i, [d_{gen}, t_{gen}]\right),$$

thereby producing samples from each of the i combinations at our desired dose and time, even when these are unmeasured in the dataset. To remove the strong cell-type-specific clustering of the generated expression vectors, we then computed the differential expression by taking

$$\hat{x}_i^{\Delta}[r] = \hat{x}_i^{(gen)}[r] - \hat{x}_i^{(gen)}[0].$$

We then computed the average differential expression for each of the 2,687 experimental combinations as

$$\bar{x}_i^{\Delta} = \frac{1}{78}\sum_{r=1}^{78} \hat{x}_i^{\Delta}[r] \in \mathbb{R}^{978}.$$

Given the average differential expression vectors, we computed a similarity score between combinations i and j as

$$\sigma_{i,j} = 1 - \left|\bar{x}_i^{\Delta} - \bar{x}_j^{\Delta}\right|_2$$

$$\Sigma_{i,j} = arg \min_{i',j'} \left(arg \max_{i'',j''} \left(\frac{\sigma_{i,j} - \sigma_{i',j'}}{\sigma_{i'',j''} - \sigma_{i',j'}} \right) \right)$$

to normalize the similarities scores to $\Sigma_{i,j} \in [0,1]$.

To construct an average differential expression k-nearest-neighbors (KNN) network G_{KNN}, we defined the hyperparameters $k = 50, m = 30, \theta = 0.95$. For each experimental combination i in our simulated dataset, we considered all edges (i,j) for all combinations j, weighted by $w(i,j) = \Sigma_{i,j}$. We followed the following procedure for constructing G from this fully-connected weighted graph, where each vertex represents an experimental drug and cell line combination.

1. Remove all edges (i,j) where $w(i,j) < \theta$.

2. For all remaining nodes i, if $degree(i) > k$ then remove the $degree(i) - k$ edges with lowest weights, resulting in a vertex with degree k.
3. Remove all nodes i where $degree(i) < m$.
4. Reduce G to its largest connected subgraph.

To further analyze G_{KNN}, we then used the Louvain community detection algorithm[86] as implemented in the Python `community` package[87] to identify communities in G_{KNN}. To reduce the complexity of the analysis, we then combined neighboring communities until 4 communities remained, $\{C_1, C_2, C_3, C_4\}$. We calculated the average community IC_{50} by taking the average of the IC_{50} doses in the GDSC dataset[33] for every vertex in the community that had a GDSC measurement.

We plotted G_{KNN} using Cytoscape.[88] We used the edge-weight spring embedded layout with minimum, maximum, and default edge weights of 0, 1, and 0.5 respectively. We ran 200 average iterations for each node. The spring strength parameter was set to 15, spring rest length to 45, the disconnected spring strength to 0.05, and the disconnected spring rest length to 2000. We did not add any spring strength to avoid collisions, and used 2 layout passes. Finally, we randomized the graph before computing the layout.

Drug sensitivity prediction dataset

For the drug IC_{50} prediction task, we randomly selected 78 different dose and time points, $[d^*, t^*]$, that had been measured in the LINCS dataset. Then, given a fully-trained Sagittarius, we could compute the transformer encoder's average key representation

$$k_{dr,cl} = \frac{1}{78}\sum_{t=1}^{78} \oplus_{h=1}^{H} k^{enc}_{h,(dr,cl,d_t^*,t_t^*)} \in \mathbb{R}^{224},$$

where $\oplus$ represents vector concatenation and H is the number of attention heads.

GDSC experiment: For the GDSC-based prediction, we computed $k_{dr,cl}$ for each GDSC[33]-measured combination of drug and cell line, provided that both the drug and cell line appeared somewhere in the LINCS dataset (although not necessarily together).

We then considered two models: one used the $k_{dr,cl}$ dataset produced by Sagittarius to predict the GDSC IC_{50} values for that experimental combination, and had 271 datapoints; the other was trained on the measured LINCS experimental combinations available that also appeared in the GDSC dataset, and had 151 datapoints. We then divided the measured LINCS-GDSC dataset into 3 splits and ran 3-fold cross validation, where for each fold the test set made up $\frac{2}{3}$ of the data. We similarly divided Sagittarius's simulated dataset such that the test set for each split matched the LINCS-GDSC dataset test split, and the rest of the data was available for training.

CTRP experiment: For this experiment we computed $k_{dr,cl}$ for each CTRP[34]-measured experimental combination. We compared this dataset from Sagittarius, which had 2,929 datapoints, to the measured LINCS-CTRP intersecting dataset, which had 625 datapoints. As before, we used 3-fold validation based on the LINCS-CTRP dataset, where $\frac{2}{3}$ of the dataset was used as test for each fold, and defined folds for the Sagittarius dataset to match the LINCS-CTRP test fold.

Drug sensitivity prediction model and hyperparameter selection

We held out 10% of the training data for both the LINCS-based and the Sagittarius datasets to determine the best regression model for the drug sensitivity prediction task. For both datasets, we tried a Support Vector Regression (SVR) model with linear, polynomial, and radial-basis-function (RBF) kernels, and an MLP regression model with regularizing weight $\alpha \in \{1e-4, 1e-2, 1, 10\}$, with all other hyperparameters maintained as the defaults in `sklearn`.[85] We evaluated the model's validation performance using the average Spearman correlation between the measured IC_{50} labels (either from GDSC or CTRP) and the model's predicted IC_{50} labels. When comparing the network performance from the LINCS-based and Sagittarius datasets, we restricted our analysis to cell lines where at least one of the two models had a statistically significant correlation (Spearman rank-order p-value < 0.05).

GDSC hyperparameters: The LINCS-GDSC dataset model achieved best validation performance with the SVR with RBF kernel; the Sagittarius dataset model achieved best validation performance with the MLP regressor with $\alpha = 10$.

CTRP hyperparameters: The LINCS-CTRP dataset model achieved best validation performance with the SVR with polynomial kernel; the Sagittarius dataset model achieved best validation performance with the MLP regressor with $\alpha = 0.01$.

Evaluation: To evaluate the model's performance on the drug sensitivity prediction task, we looked at the Spearman correlation per drug. For the test drugs and cell lines with a measured drug sensitivity $s(d, c)$ for drug d and cell line c, and a corresponding predicted drug sensitivity $\hat{s}(d, c)$, we defined the model's Spearman performance for drug d as

$$\rho(d) = \rho_{spearman}\left(\left[\hat{s}(d, c_1), \hat{s}(d, c_2), \ldots, \hat{s}(d, c_{N_d})\right], \left[s(d, c_1), s(d, c_2), \ldots, s(d, c_{N_d})\right]\right),$$

where $\left[c_1, c_2, \cdots, c_{N_d}\right]$ are the N_d cell lines that were treated with drug d in the GDSC or CTRP datasets.

Cancer gene essentiality prediction: dataset construction

For each tested gene g and cell line cl combination in the DepMap dataset, independently considering the DEMETER[36] and CERES[35] versions, we searched for all drugs dr in the LINCS dataset[15] that listed gene g as the drug's target. Given 78 randomly-selected doses $d^{\cdot}$ and times $t^{\cdot}$ from the set of all doses and times that had been measured in the LINCS dataset, we computed the transformer encoder's average key representation

$$k_{dr,cl} = \frac{1}{78}\sum_{t=1}^{78} \oplus_{h=1}^{H} k^{enc}_{h,(dr,cl,d_t^{\cdot},t_t^{\cdot})} \in \mathbb{R}^{224},$$

where $\oplus$ indicates concatenation and $H = 8$. This resulted in 4,216 datapoints for the DEMETER version and 1,666 for the CERES versions.

We also constructed a LINCS-DepMap dataset. For every gene g and cell line cl in the DepMap dataset, we searched for a drug dr in the LINCS dataset that listed g as its gene target. If such a drug existed in the dataset, we added the corresponding average measured post-treatment expression across all tested doses and times from LINCS to the LINCS-DepMap dataset. This resulted in 765 datapoints for the DEMETER version and 353 datapoints for the CERES version.

Gene essentiality model hyperparameters and evaluation

For both the LINCS-DepMap dataset model and the Sagittarius dataset model, we trained an MLP regressor using two hidden layers with 200- and 100 hidden nodes respectively, ReLU activation functions, mean-squared-error loss, and the Adam optimizer[84] with a learning rate of 1e-3. We used 5-fold cross validation, where 20% of the LINCS-DepMap dataset was used as the test set, and we aligned the Sagittarius dataset's test set to match the LINCS-DepMap test set. We further held out 10% of the resulting training set for each of the 5 splits to use as a validation set for early model training termination.

To evaluate the model, we computed the Spearman correlation for each cell line. Denoting each tested cell line as c, with the measured tested target genes essentialities $[e(c,g_1), e(c,g_2), \ldots, e(c,g_{N_c})]$ and corresponding predicted essentiality $[\hat{e}(c,g_1), \hat{e}(c,g_2), \cdots, \hat{e}(c,g_{N_c})]$, we computed the Spearman correlation as

$$\rho(c) = \rho_{spearman}\left(\left[\hat{e}(c,g_1), \hat{e}(c,g_2), \ldots, \hat{e}(c,g_{N_c})\right], \left[e(c,g_1), e(c,g_2), \ldots, e(c,g_{N_c})\right]\right).$$

TCGA dataset processing

We used the TCGA Firehose legacy dataset[37] mutation data. Each patient's mutation profile is mapped to somatic mutations from 20,501 total genes. The mutation profiles are binary vectors,

where 1 indicates that the gene contained a mutation. We first removed all patients with nan survival times and then restricted the dataset to the 1,000-most frequently mutated genes across all cancer types. We then removed patients from the dataset if they had no profiled mutations across any of the remaining 1,000 genes. If there were fewer than 12 patients remaining in a given cancer type, we also excluded this cancer type. We constructed a time series for each cancer type by ordering the remaining patients according to their labeled survival times, sorted from shortest survival to longest survival. We then constructed the dataset of (x_i, y_i, t_i) where x_i are the sequence of mutation profiles for patients of cancer type y_i, each with the corresponding survival times t_i.

Time series patient filtering

In order to apply Sagittarius's time series framework to the TCGA mutation profiles, we needed the cancer type time series to accurately reflect cancer survival times. In particular, inclusion of patients with a censored survival time, meaning they lost contact with the study before an observed death event, might lead to incorrect overall survival times and relative ordering of patients in the time series. However, excluding all patients with censored survival times would greatly decrease the size of the dataset and limit the statistical power of the model (**Supplementary Fig. 8**).

We hypothesized that censored patients could be divided approximately into two categories. First, some patients who lost contact with the study might die shortly afterwards, meaning their censored survival time (the time at which they lost contact with the study) would closely reflect their overall survival time, were it to have been observed. Second, some patients who lost contact with the study would survive well beyond the censoring time, and therefore the censored survival time would be substantially different from the overall survival time. The first of these two categories could therefore be included in the time series formulation, with censored survival time used as a proxy for overall survival time; the second category should be excluded to maintain the integrity of the input time series.

In order to identify censored patients belonging to the first category, where censored survival times closely reflected overall survival times, we trained a neural network on each cancer type individually to predict $t_i[r]$ from $x_i[r]$. In this step, we also included each patient's binary censoring label $c[r]$, where $c[r] = 0$ indicates that the rth patient had an observed death event. We defined the individual patient loss as

$$\mathcal{L}_{individual}(x[r], t[r], \hat{t}[r], c[r]) = \mathbb{1}[c[r] = 0]\,|\hat{t} - t|_1 + \mathbb{1}[c[r] = 1]\,max(t - \hat{t}, 0),$$

thereby not penalizing the model for overestimating the survival time of a censored patient. We further defined the empirical model loss as

$$\mathcal{L}_{\phi}(x, t, \hat{t}, c) = \frac{1}{T}\sum_{r=1}^{T} \mathcal{L}_{individual}(x[r], t[r], \hat{t}[r], c[r]) + \lambda|\phi|_2^2.$$

Using a single hidden layer with 32 neurons, we trained the regressor on each cancer type individually with the regularizing weight $\lambda = 0.3$, stochastic gradient descent (SGD) optimizer, and learning rate of 1e-1. We trained for up to 2,500 epochs, and selected the model epoch with the maximal concordance index for the patients with an observed death event.

We then leveraged techniques from Learning with Noisy Labels (LNL), where each patient's survival time, either observed or censored, represented a potentially noisy label for their actual survival time. For each cancer type's selected model, we computed the absolute error per patient for both the observed ($|\epsilon_{obs}|$) and censored ($|\epsilon_{cens}|$) groups. Using the scipy[89] python package, we fit a beta distribution β_{obs} to $|\epsilon_{obs}|$ and a beta distribution β_{cens} to $|\epsilon_{cens}|$, and then computed the probability that each patient in $|\epsilon_{obs}|$ and $|\epsilon_{cens}|$ could have been generated by either β_{obs}or β_{cens}. Following previous work,[61,62] if the probability that a censored patient's absolute error was generated by β_{obs} was larger than the probability it was generated by β_{cens}, we switched their label to observed. Similarly, if a censored patient's absolute error was smaller than the absolute error for at least one observed patient, we also switched their label to observed. We discarded all other censored patients, and retained only the observed patients and the censored patients with a swapped label as the resulting cancer type time series. After filtering, our dataset contained 2297 cancer patients. For the SARC time series, we included 31 patients with a censored death event to include 115 total patients; for the THCA time series, we included 2 patients with a censored death event to include 15 total patients.

Model hyperparameters

As in the Evo-devo dataset, the mean and linear comparison approaches did not require any hyperparameters. For Sagittarius, we used 20% of the available data as a validation set for hyperparameter selection and training termination. We used $S + 1 = 4$ temporal reference points, a latent dimension $d = 16$, $H = 8$ attention heads, an 8-dimensional cancer type embedding for the encoder and decoder, a 2-dimensional temporal embedding for the transformer, and a 4-dimensional cancer type embedding for the transformer. We set $\beta = 1$ and used a batch size of 2. We tried both 1-, 2-, and 3 hidden layer symmetric MLPs for the encoder and decoder, each with 256 hidden neurons per layer. For the SARC cancer type, we selected the 2-layer MLPs based on validation performance; for the THCA cancer type, we selected 3-layer MLPs.

In addition to the reconstruction task where $i = j$ in $\mathcal{L}_{Sag}(.)$, we trained Sagittarius with 2 simulation objectives.

1. Time generation: we randomly selected 12 cancer type time series and, for each, randomly masked out 3 patients in the time series to produce $\bar{x_{i'}}$. The time series made up of the 3 masked patients was its partner training sequence $\widetilde{x_{j'}}$.
2. Cancer type generation: we randomly selected 12 pairs of cancer type time series. We considered one of each pair to be $\bar{x_{i'}}$ and the other as its partner sequence $\widetilde{x_{j'}}$.

Sagittarius was then also trained a generation objective, formulated according to the empirical version of the loss term $\mathcal{L}_{Sag}(.)$, with $i = i'$ and $j = j'$.

Quantitative cancer patient extrapolation experiment

We defined the extrapolation task as follows. For the cancer type with N observed patients, we defined a training split of the k observed patients with shortest survival time, as well as all censored patients with a survival time shorter than the longest of the k observed survival times, and used the remaining $N - k$ patients as the test split. We then varied $k = 1, \ldots, N - 1$. For each of the $N - 2$ different test splits, we evaluated model performance with the AUROC of the simulated mutation profiles. For this evaluation, we restricted the genes that we evaluated to those that had a pattern with respect to survival time. Specifically, we used the augmented Dickey-Fuller statistical test to identify the mutations for each cancer type for which we did not reject the null hypothesis, indicating non-stationarity, with a significance threshold of $p < 0.05$. We then further took the union of the genes that appeared in the non-stationary gene sets for at least δ cancer types to create the evaluation gene set $\gamma(\delta)$. We excluded any test patients that did not have a measured mutation in any of the genes in $\gamma(\delta)$. Based on the resulting number of test patients, we used $\delta = 2$ for THCA, resulting in 9 usable test splits, and $\delta = 4$ for SARC, resulting in 61 usable test splits. Then, we evaluate the model performance on cancer type y_i for a test patient with survival time $t > \tau$, for some threshold test set threshold τ, as

$$AUROC_i(\tau, t) = AUROC(\hat{x}_i[t, \gamma(\delta)], x_i[t, \gamma(\delta)]),$$

where $x_i[t, \gamma(\delta)]$ is the measured mutation profile for a patient with cancer type i and survival time t, restricted to the genes in set $\gamma(\delta)$, and $AUROC(\cdot)$ is the AUROC computation between two vectors. Then we computed the model's overall performance on the test split beginning with test patient survival time τ as

$$AUROC_i(\tau) = \frac{1}{|\{t_j \in T_i : t_i \geq \tau\}|} \sum_{t \in T_i} 1\{(t \geq \tau) \wedge (x_i[t, \gamma(\delta)] \neq \vec{0})\} AUROC_i(\tau, t),$$

where $1\{\cdot\}$ represents an indicator variable. Then, to compare the model performance across test splits for cancer type i, we defined the overall model performance as the average of the test split AUROCs for that cancer type, or

$$AUROC_i = \frac{1}{T_i - 1} \sum_{\tau \in \vec{t_i}[2:]} AUROC_i(\tau).$$

Early cancer patient mutation simulation

To simulate the early-stage sarcoma patient mutation profiles, we trained Sagittarius on all available TCGA data and then simulated mutation probability profiles at 27 survival time points, ranging from 203-283 months. Specifically, we selected the longest 27 survival times that appeared somewhere in the initial TCGA dataset, with

$$t \in \{203.12, 204.01, 260.70, 208.23, 209.43, 210.51, 210.81, 211.01, 211.73, 212.09, 216.59, 216.75, 225.43, 229.04, 230.72, 232.00, 232.62, 233.44, 234.10, 238.11, 244.32, 244.91, 255.49, 263.07, 275.66, 281.08, 282.69\}$$

Months (**Supplementary Fig. 10**). We then averaged the mutation profile predictions of the 27 time points and identified the 10 genes the model predicted as most likely to be mutated.

Figures

Figures were created with BioRender.

Data availability

The datasets used for this project are available at https://figshare.com/projects/Sagittarius/144771.

Code availability

A python repository including the Sagittarius implementation and code to reproduce the results in this paper is available at https://github.com/addiewc/Sagittarius.

1. Gulati, G. S. *et al.* Single-cell transcriptional diversity is a hallmark of developmental potential. *Science* **367**, 405–411 (2020).

2. Arbeitman, M. N. *et al.* Gene expression during the life cycle of Drosophila melanogaster. *Science* **297**, 2270–2275 (2002).

3. Zheng, L. *et al.* Pan-cancer single-cell landscape of tumor-infiltrating T cells. *Science* **374**, abe6474 (2021).

4. Klein, A. M. *et al.* Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell* **161**, 1187–1201 (2015).

5. Lee, J. S. *et al.* Single-cell transcriptome of bronchoalveolar lavage fluid reveals sequential change of macrophages during SARS-CoV-2 infection in ferrets. *Nat. Commun.* **12**, 4567 (2021).

6. Vento-Tormo, R. *et al.* Single-cell reconstruction of the early maternal-fetal interface in humans. *Nature* **563**, 347–353 (2018).

7. Douglass, E. F., Jr *et al.* A community challenge for a pancancer drug mechanism of action inference from perturbational profile data. *Cell Rep Med* **3**, 100492 (2022).

8. Kohonen, P. *et al.* A transcriptomics data-driven gene space accurately predicts liver cytopathology and drug-induced liver injury. *Nat. Commun.* **8**, 15932 (2017).

9. Almogy, G. *et al.* Cost-efficient whole genome-sequencing using novel mostly natural sequencing-by-synthesis chemistry and open fluidics platform. *bioRxiv* 2022.05.29.493900 (2022) doi:10.1101/2022.05.29.493900.

10. Tang, F. *et al.* mRNA-Seq whole-transcriptome analysis of a single cell. *Nat. Methods* **6**, 377–382 (2009).

11. Ramsköld, D. *et al.* Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. *Nat. Biotechnol.* **30**, 777–782 (2012).

12. Cardoso-Moreira, M. *et al.* Gene expression across mammalian organ development. *Nature* **571**, 505–509 (2019).

13. Cao, J. *et al.* The single-cell transcriptional landscape of mammalian organogenesis. *Nature* **566**, 496–502 (2019).

14. Cao, J., Zhou, W., Steemers, F., Trapnell, C. & Shendure, J. Sci-fate characterizes the dynamics of gene expression in single cells. *Nat. Biotechnol.* **38**, 980–988 (2020).

15. Subramanian, A. *et al.* A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles. *Cell* **171**, 1437–1452.e17 (2017).

16. Tabula Muris Consortium. A single-cell transcriptomic atlas characterizes ageing tissues in the mouse. *Nature* **583**, 590–595 (2020).

17. Schaum, N. *et al.* Ageing hallmarks exhibit organ-specific temporal signatures. *Nature* **583**, 596–602 (2020).

18. Wang, W. *et al.* Single-cell transcriptomic atlas of the human endometrium during the menstrual cycle. *Nat. Med.* **26**, 1644–1653 (2020).

19. Sunkin, S. M. *et al.* Allen Brain Atlas: an integrated spatio-temporal portal for exploring the central nervous system. *Nucleic Acids Res.* **41**, D996–D1008 (2013).

20. Radovic, A., He, J., Ramanan, J., Brubaker, M. A. & Lehrmann, A. M. Agent Forecasting at Flexible Horizons using ODE Flows. *ICML* (2021).

21. Street, K. *et al.* Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics* **19**, 477 (2018).

22. Setty, M. *et al.* Characterization of cell fate probabilities in single-cell data with Palantir. *Nat. Biotechnol.* **37**, 451–460 (2019).

23. Peng, G., Cui, G., Ke, J. & Jing, N. Using Single-Cell and Spatial Transcriptomes to Understand Stem Cell Lineage Specification During Early Embryo Development. *Annu. Rev. Genomics Hum. Genet.* **21**, 163–181 (2020).

24. Haniffa, M. *et al.* A roadmap for the Human Developmental Cell Atlas. *Nature* **597**, 196–205

(2021).

25. Sohn, K., Lee, H. & Yan, X. Learning Structured Output Representation using Deep Conditional Generative Models. in *Advances in Neural Information Processing Systems* vol. 28 (Curran Associates, Inc., 2015).

26. Lotfollahi, M. *et al.* Compositional perturbation autoencoder for single-cell response modeling. *bioRxiv* 2021.04.14.439903 (2021) doi:10.1101/2021.04.14.439903.

27. Cho, K. *et al.* Learning Phrase Representations using RNN Encoder--Decoder for Statistical Machine Translation. in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 1724–1734 (Association for Computational Linguistics, 2014).

28. Chen, R. T. Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D. K. Neural Ordinary Differential Equations. in *Advances in Neural Information Processing Systems* vol. 31 (Curran Associates, Inc., 2018).

29. Shukla, S. N. & Marlin, B. Multi-Time Attention Networks for Irregularly Sampled Time Series. in *International Conference on Learning Representations* (2021).

30. Chen, R. T. Q., Amos, B. & Nickel, M. Learning Neural Event Functions for Ordinary Differential Equations. in (2022).

31. Vaswani, Shazeer & Parmar. Attention is all you need. *Adv. Neural Inf. Process. Syst.* (2017).

32. Rahaman, N. *et al.* On the Spectral Bias of Neural Networks. in *Proceedings of the 36th International Conference on Machine Learning* (eds. Chaudhuri, K. & Salakhutdinov, R.) vol. 97 5301–5310 (PMLR, 09--15 Jun 2019).

33. Iorio, F. *et al.* A Landscape of Pharmacogenomic Interactions in Cancer. *Cell* **166**, 740–754 (2016).

34. Seashore-Ludlow, B. *et al.* Harnessing Connectivity in a Large-Scale Small-Molecule Sensitivity Dataset. *Cancer Discov.* **5**, 1210–1223 (2015).

35. Meyers, R. M. *et al.* Computational correction of copy number effect improves specificity of CRISPR–Cas9 essentiality screens in cancer cells. *Nat. Genet.* **49**, 1779–1784 (2017).

36. Tsherniak, A. *et al.* Defining a Cancer Dependency Map. *Cell* **170**, 564–576.e16 (2017).

37. Cancer Genome Atlas Research Network *et al.* The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.* **45**, 1113–1120 (2013).

38. Fisher, R. A. 014: On the' Probable Error' of a Coefficient of Correlation Deduced from a Small Sample. (1921).

39. Tam, P. P. & Behringer, R. R. Mouse gastrulation: the formation of a mammalian body plan. *Mech. Dev.* **68**, 3–25 (1997).

40. Pijuan-Sala, B. *et al.* A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature* **566**, 490–495 (2019).

41. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv [stat.ML]* (2018).

42. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **24**, 417–441 (1933).

43. Qiu, C. *et al.* Systematic reconstruction of cellular trajectories across mouse embryogenesis. *Nat. Genet.* **54**, 328–341 (2022).

44. Briggs, J. A. *et al.* The dynamics of gene expression in vertebrate embryogenesis at single-cell resolution. *Science* **360**, (2018).

45. Qiao, G.-J., Chen, L., Wu, J.-C. & Li, Z.-R. Identification of an eight-gene signature for

survival prediction for patients with hepatocellular carcinoma based on integrated bioinformatics analysis. *PeerJ* **7**, e6548 (2019).

46. Takada, H. & Kurisaki, A. Emerging roles of nucleolar and ribosomal proteins in cancer, development, and aging. *Cell. Mol. Life Sci.* **72**, 4015–4025 (2015).

47. Loganathan, T., Ramachandran, S., Shankaran, P., Nagarajan, D. & Mohan S, S. Host transcriptome-guided drug repurposing for COVID-19 treatment: a meta-analysis based approach. *PeerJ* **8**, e9357 (2020).

48. Belyaeva, A. *et al.* Causal network models of SARS-CoV-2 expression and aging to identify candidates for drug repurposing. *Nat. Commun.* **12**, 1024 (2021).

49. Minamiyama, M. *et al.* Naratriptan mitigates CGRP1-associated motor neuron degeneration caused by an expanded polyglutamine repeat tract. *Nat. Med.* **18**, 1531–1538 (2012).

50. Yang, C. *et al.* A survey of optimal strategy for signature-based drug repositioning and an application to liver cancer. *Elife* **11**, (2022).

51. Cheng, X. *et al.* Drug repurposing for cancer treatment through global propagation with a greedy algorithm in a multilayer network. *Cancer Biol Med* (2021) doi:10.20892/j.issn.2095-3941.2020.0218.

52. Folkes, A. J. *et al.* The identification of 2-(1H-indazol-4-yl)-6-(4-methanesulfonyl-piperazin-1-ylmethyl)-4-morpholin-4-yl-thieno[3,2-d]pyrimidine (GDC-0941) as a potent, selective, orally bioavailable inhibitor of class I PI3 kinase for the treatment of cancer. *J. Med. Chem.* **51**, 5522–5532 (2008).

53. Suzuki, N., Nakagawa, F., Matsuoka, K. & Takechi, T. Effect of a novel oral chemotherapeutic agent containing a combination of trifluridine, tipiracil and the novel

triple angiokinase inhibitor nintedanib, on human colorectal cancer xenografts. *Oncol. Rep.* **36**, 3123–3130 (2016).

54. Roth, G. J. *et al.* Nintedanib: from discovery to the clinic. *J. Med. Chem.* **58**, 1053–1063 (2015).

55. Menden, M. P. *et al.* Community assessment to advance computational prediction of cancer drug combinations in a pharmacogenomic screen. *Nat. Commun.* **10**, 2674 (2019).

56. Corsello, S. M. *et al.* The Drug Repurposing Hub: a next-generation drug library and information resource. *Nat. Med.* **23**, 405–408 (2017).

57. Bozic, I. *et al.* Accumulation of driver and passenger mutations during tumor progression. *Proc. Natl. Acad. Sci. U. S. A.* **107**, 18545–18550 (2010).

58. Chen, X. *et al.* Non-invasive early detection of cancer four years before conventional diagnosis using a blood test. *Nat. Commun.* **11**, 3475 (2020).

59. International Cancer Genome Consortium *et al.* International network of cancer genome projects. *Nature* **464**, 993–998 (2010).

60. Zehir, A. *et al.* Mutational landscape of metastatic cancer revealed from prospective clinical sequencing of 10,000 patients. *Nat. Med.* **23**, 703–713 (2017).

61. Arazo, E., Ortego, D., Paul, A., O'Connor, N. E. & McGuinness, K. Unsupervised Label Noise Modeling and Loss Correction. *ICML* (2019).

62. Li, J., Socher, R. & Hoi, S. C. H. DivideMix: Learning with noisy labels as semi-supervised learning. *arXiv [cs.CV]* (2020).

63. Brown, L. C. *et al.* LRP1B mutations are associated with favorable outcomes to immune checkpoint inhibitors across multiple cancer types. *J Immunother Cancer* **9**, (2021).

64. Arang, N. & Gutkind, J. S. G Protein-Coupled receptors and heterotrimeric G proteins as

cancer drivers. *FEBS Lett.* **594**, 4201–4232 (2020).

65. Ichikawa, D. *et al.* Integrated diagnosis based on transcriptome analysis in suspected pediatric sarcomas. *NPJ Genom Med* **6**, 49 (2021).

66. Pietrobono, S., Gagliardi, S. & Stecca, B. Non-canonical Hedgehog Signaling Pathway in Cancer: Activation of GLI Transcription Factors Beyond Smoothened. *Front. Genet.* **10**, 556 (2019).

67. Banerjee, S. *et al.* Loss of the PTCH1 tumor suppressor defines a new subset of plexiform fibromyxoma. *J. Transl. Med.* **17**, 246 (2019).

68. Martinez, M. F. *et al.* Nevoid Basal Cell Carcinoma Syndrome: PTCH1 Mutation Profile and Expression of Genes Involved in the Hedgehog Pathway in Argentinian Patients. *Cells* **8**, (2019).

69. Lo, W. W., Pinnaduwage, D., Gokgoz, N., Wunder, J. S. & Andrulis, I. L. Aberrant hedgehog signaling and clinical outcome in osteosarcoma. *Sarcoma* **2014**, 261804 (2014).

70. Yoon, J. W. *et al.* Noncanonical regulation of the Hedgehog mediator GLI1 by c-MYC in Burkitt lymphoma. *Mol. Cancer Res.* **11**, 604–615 (2013).

71. Vatapalli, R. *et al.* Histone methyltransferase DOT1L coordinates AR and MYC stability in prostate cancer. *Nat. Commun.* **11**, 4153 (2020).

72. Ge, Z. *et al.* Clinical significance of high c-MYC and low MYCBP2 expression and their association with Ikaros dysfunction in adult acute lymphoblastic leukemia. *Oncotarget* **6**, 42300–42311 (2015).

73. Wang, X., Haswell, J. R. & Roberts, C. W. M. Molecular pathways: SWI/SNF (BAF) complexes are frequently mutated in cancer--mechanisms and potential therapeutic

insights. *Clin. Cancer Res.* **20**, 21–27 (2014).

74. Tazzari, M. *et al.* Molecular Determinants of Soft Tissue Sarcoma Immunity: Targets for Immune Intervention. *Int. J. Mol. Sci.* **22**, (2021).

75. Stecca, B. & Ruiz i Altaba, A. Context-dependent regulation of the GLI code in cancer by HEDGEHOG and non-HEDGEHOG signals. *J. Mol. Cell Biol.* **2**, 84–95 (2010).

76. Brechbiel, J., Miller-Moslin, K. & Adjei, A. A. Crosstalk between hedgehog and other signaling pathways as a basis for combination therapies in cancer. *Cancer Treat. Rev.* **40**, 750–759 (2014).

77. Fan, X. *et al.* The association between methylation patterns of DNAH17 and clinicopathological factors in hepatocellular carcinoma. *Cancer Med.* **8**, 337–350 (2019).

78. Hassounah, N. B., Bunch, T. A. & McDermott, K. M. Molecular pathways: the role of primary cilia in cancer progression and therapeutics with a focus on Hedgehog signaling. *Clin. Cancer Res.* **18**, 2429–2435 (2012).

79. Chen, J., Zhang, J., Hong, L. & Zhou, Y. EGFLAM correlates with cell proliferation, migration, invasion and poor prognosis in glioblastoma. *Cancer Biomark.* **24**, 343–350 (2019).

80. Yu, Q. *et al.* Upregulated NLGN1 predicts poor survival in colorectal cancer. *BMC Cancer* **21**, 884 (2021).

81. Yeo, G. H. T., Saksena, S. D. & Gifford, D. K. Generative modeling of single-cell time series with PRESCIENT enables prediction of cell trajectories with interventions. *Nat. Commun.* **12**, 3222 (2021).

82. Fischer, D. S. *et al.* Inferring population dynamics from single-cell RNA-sequencing time series data. *Nat. Biotechnol.* **37**, 461–468 (2019).

83. de Ruiter, J. *pybiomart: A simple pythonic interface to biomart*. (Github).

84. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *ICLR* (2015).

85. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

86. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech: Theory Exp.* **2008**, P10008 (2008).

87. Aynaud, T. python-louvain 0.15: Louvain algorithm for community detection. Preprint at https://github.com/taynaud/python-louvain\texttthttps://github.com/taynaud/python-louvain (2020).

88. Shannon, P. *et al.* Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Res.* **13**, 2498–2504 (2003).

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

Fig.1

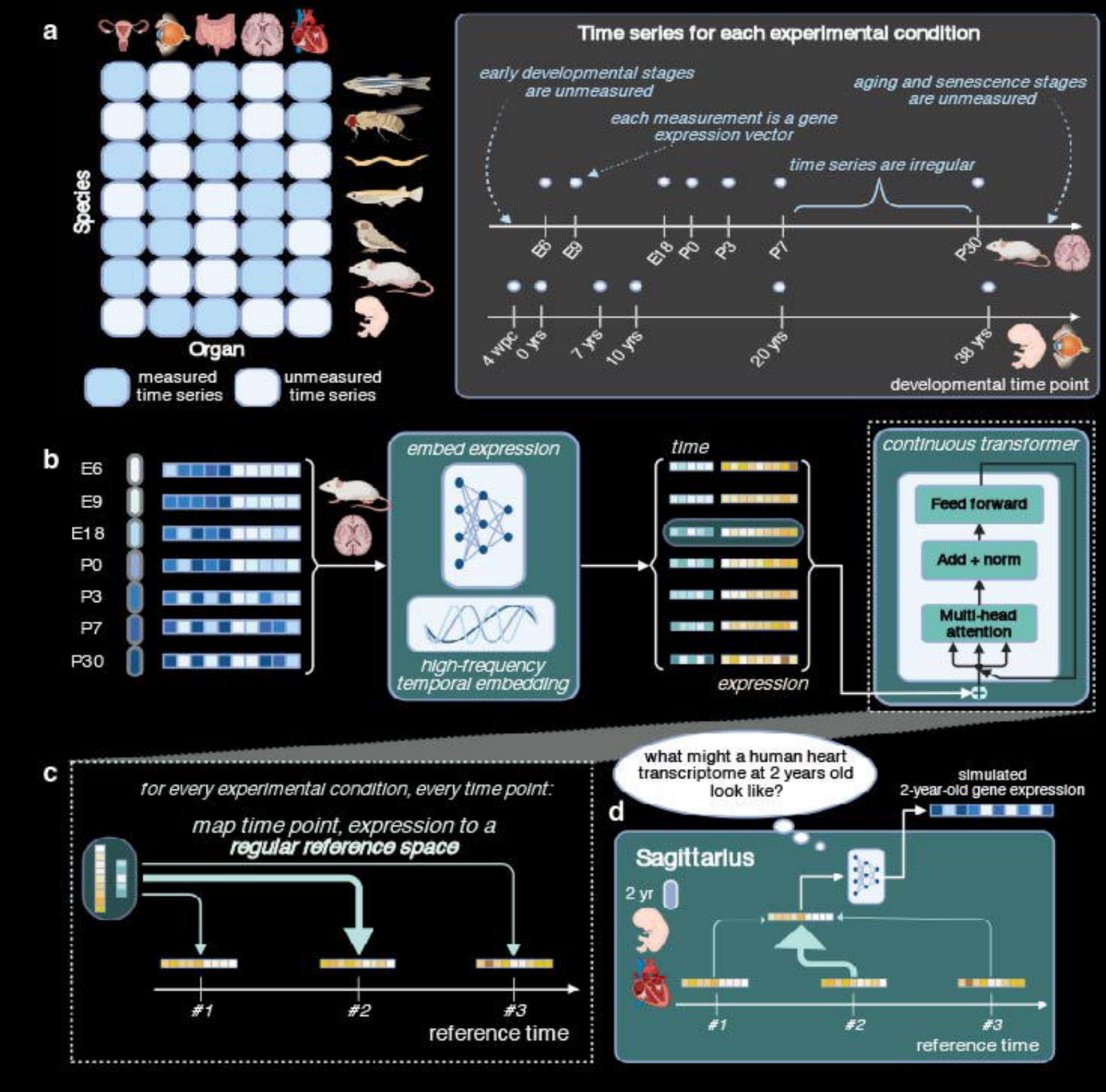
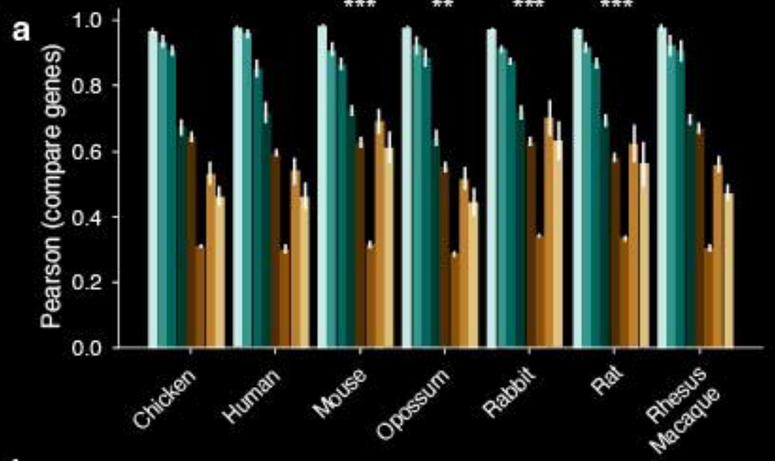
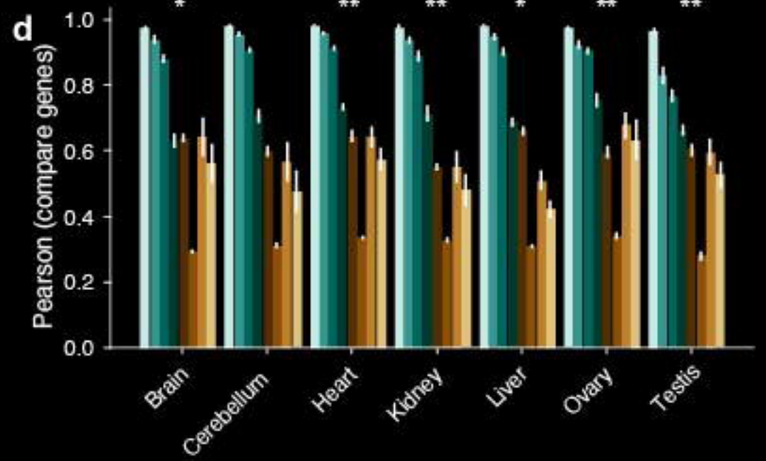
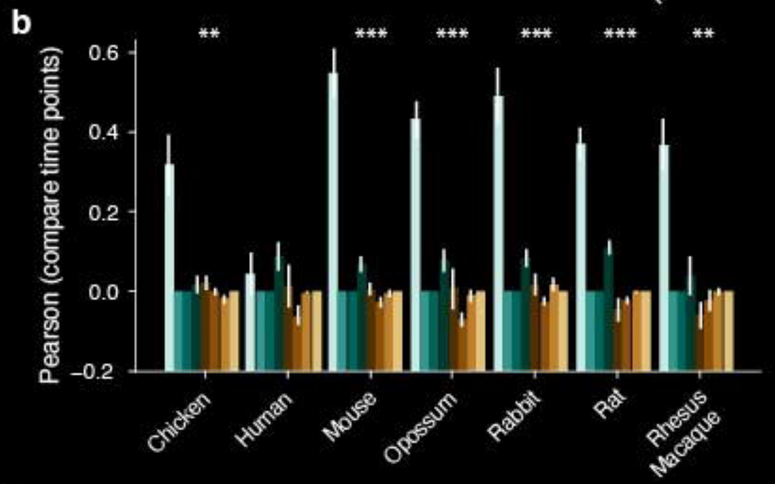
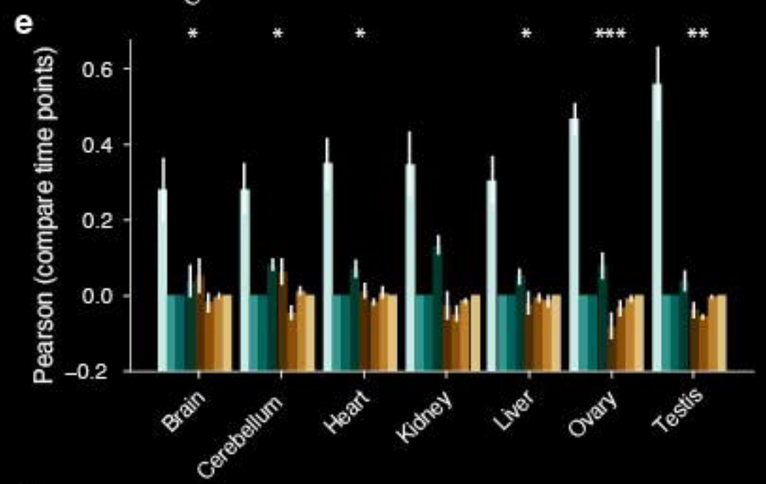


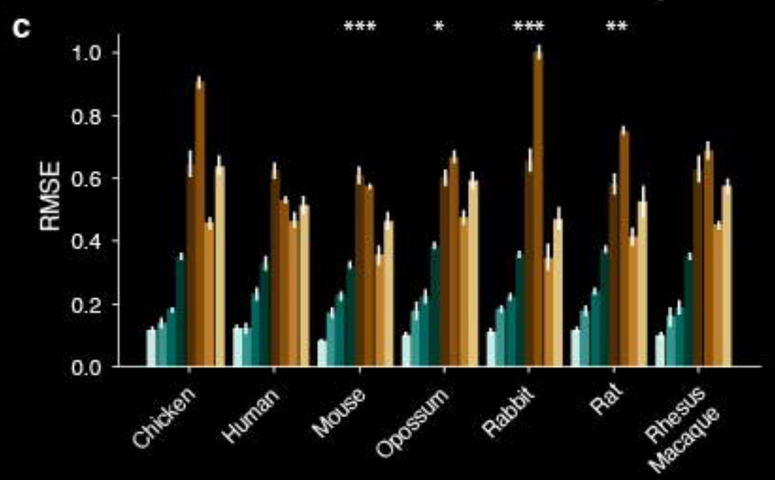
Fig. 2











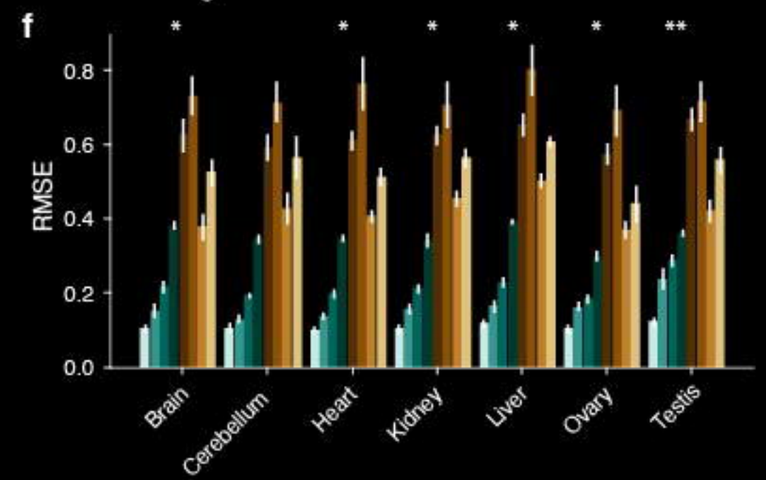


Fig. 3

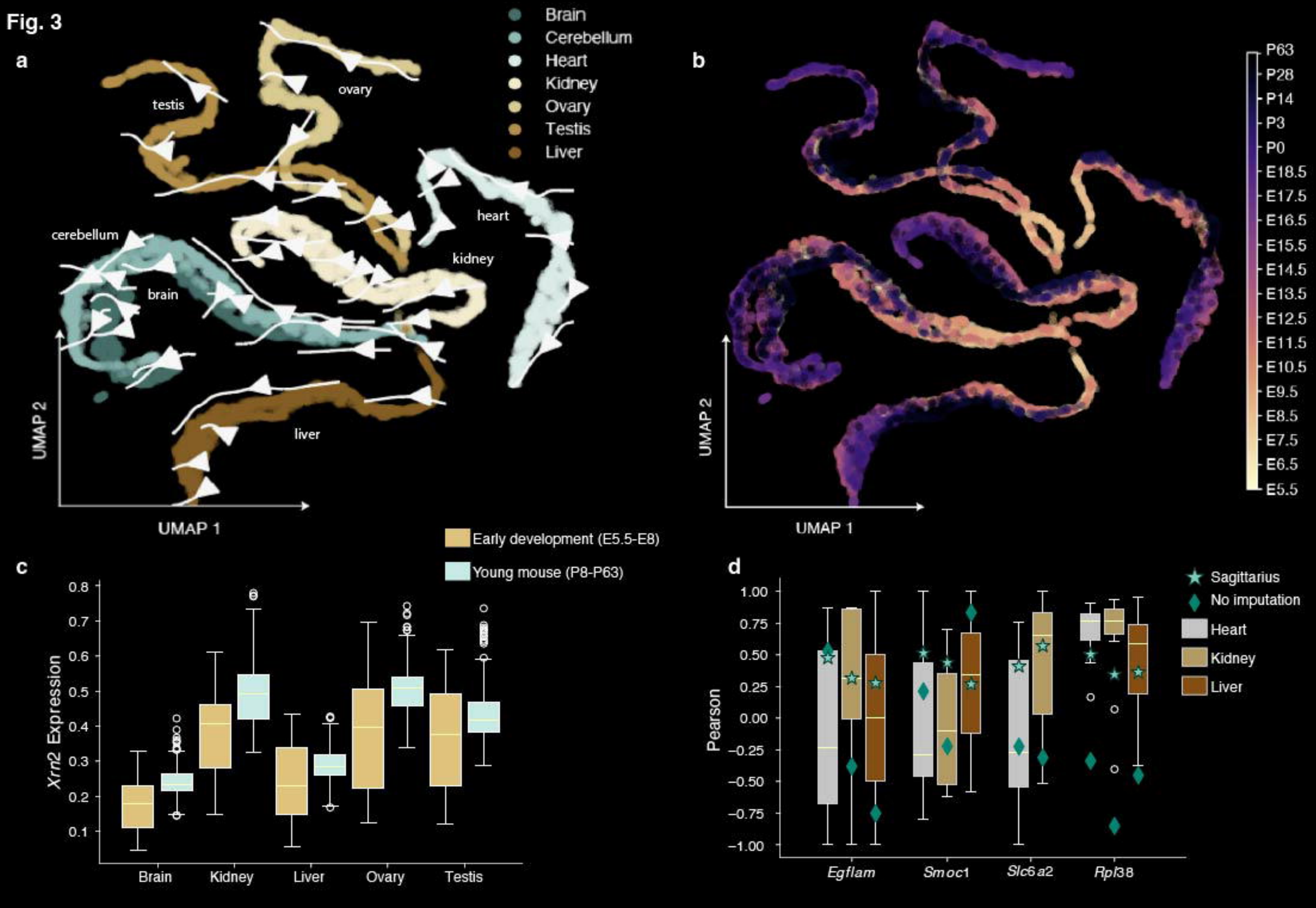
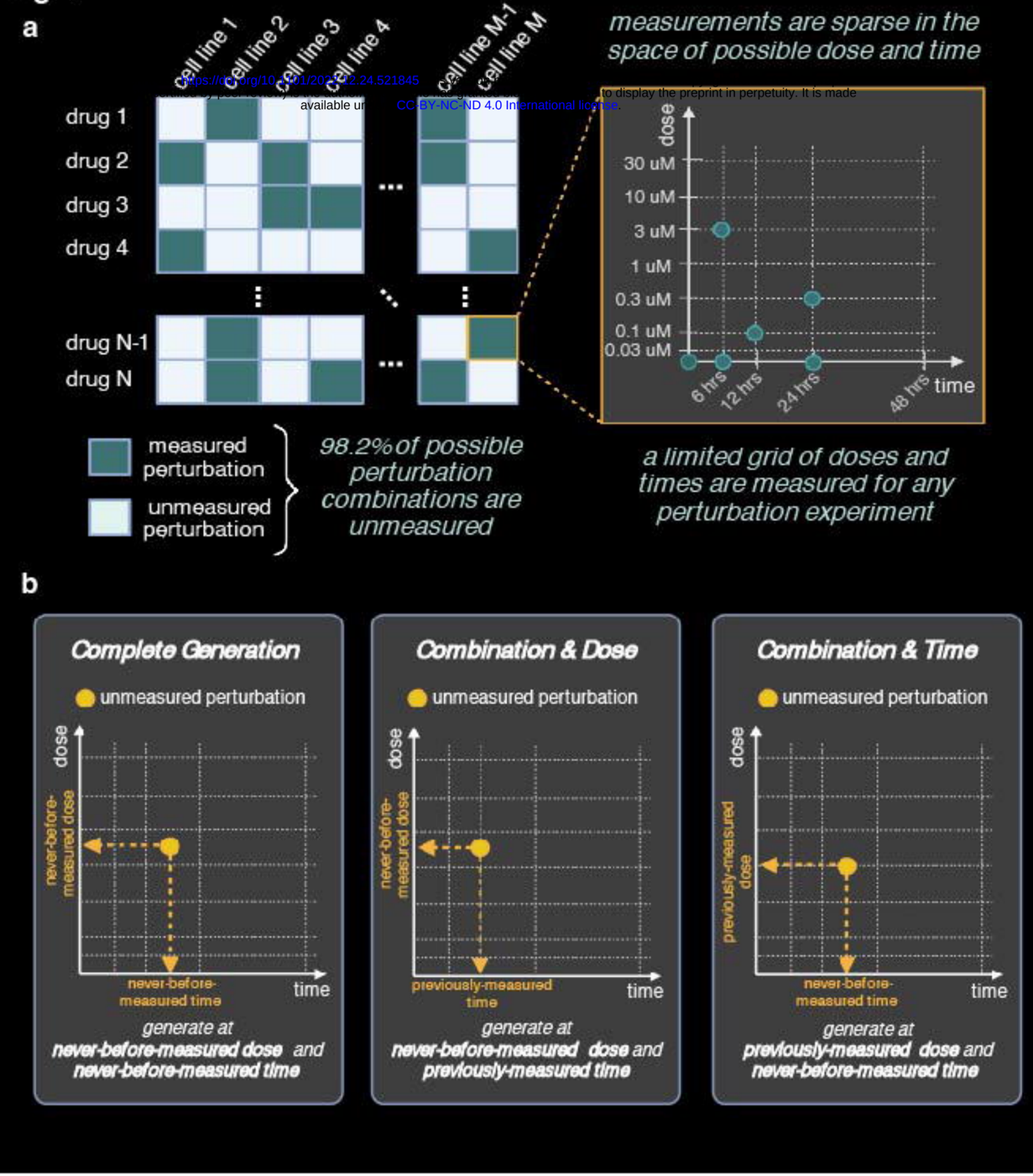
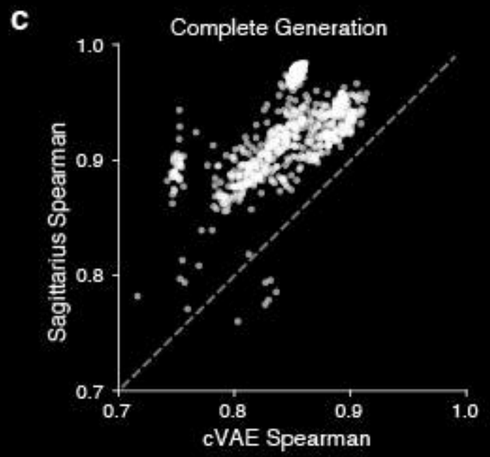
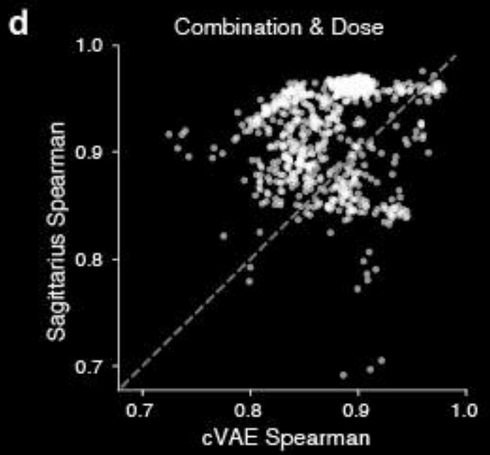


Fig. 4







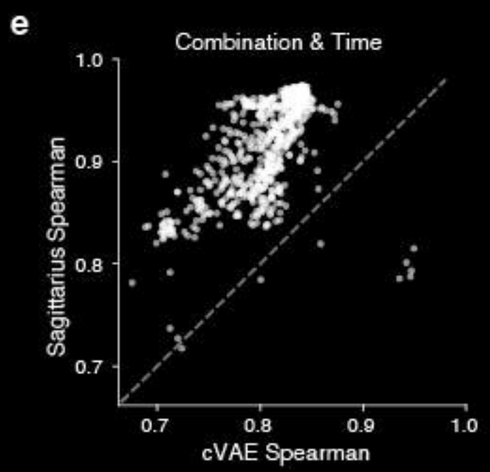
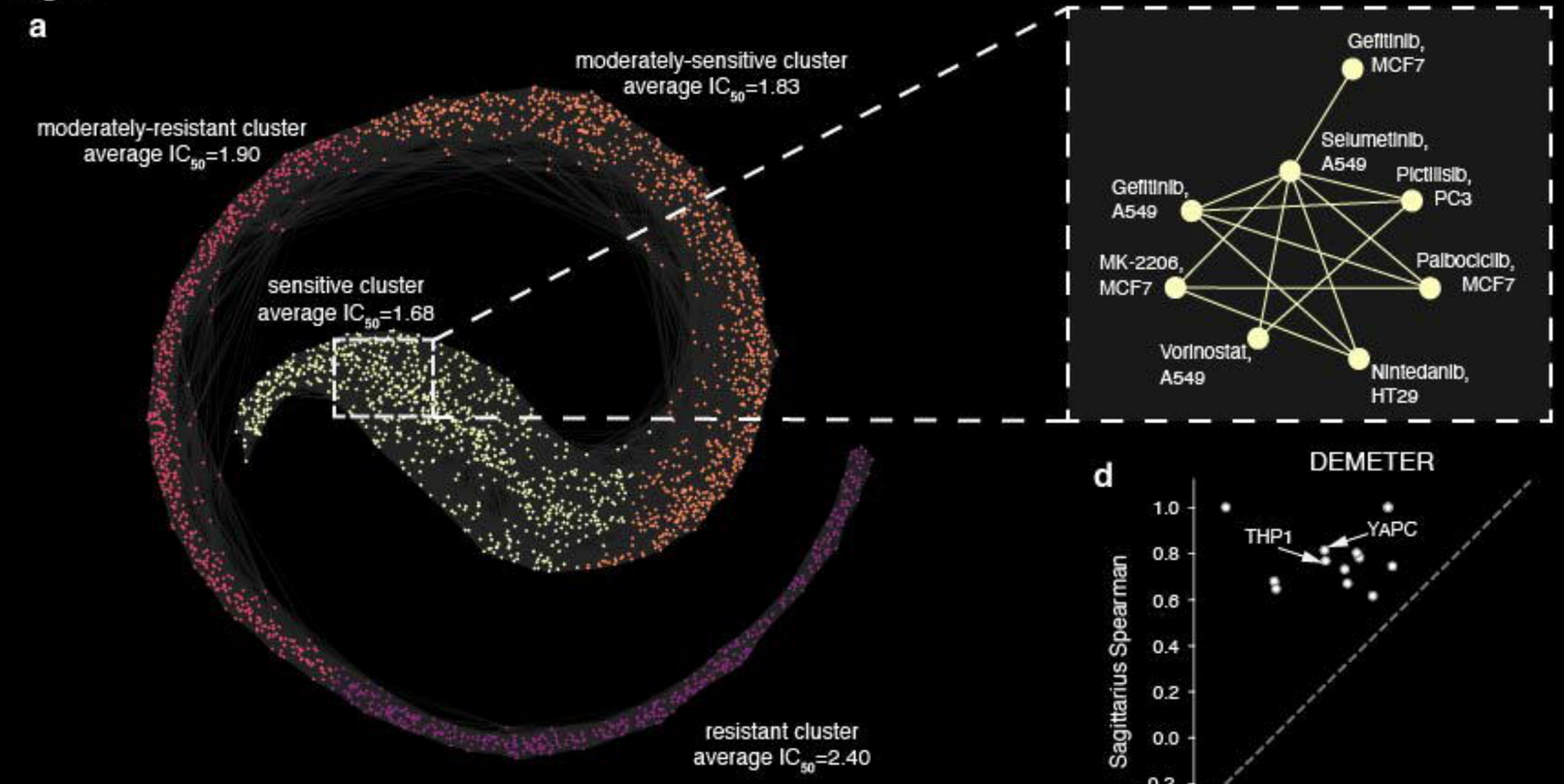
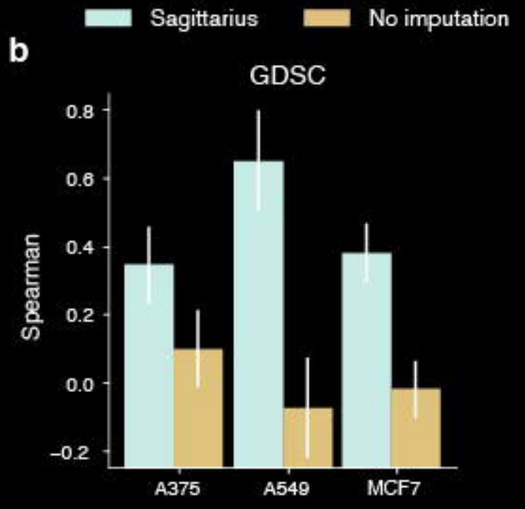
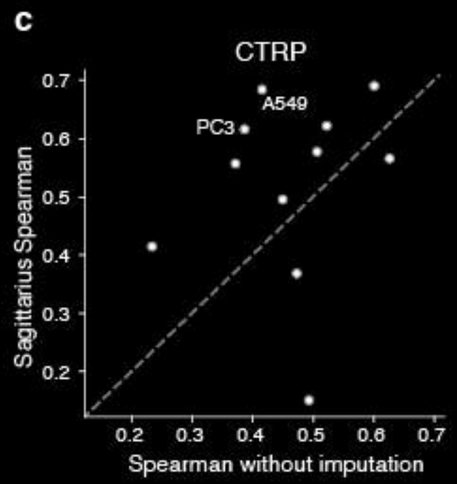
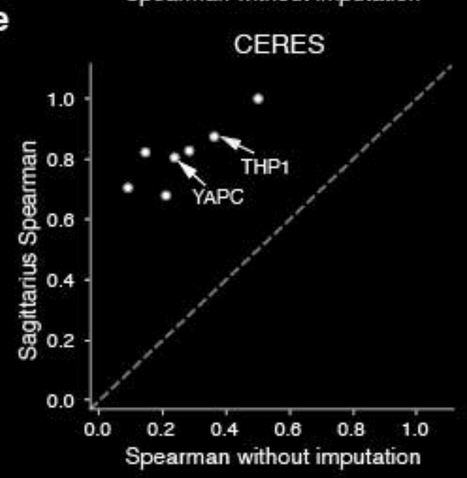


Fig. 5

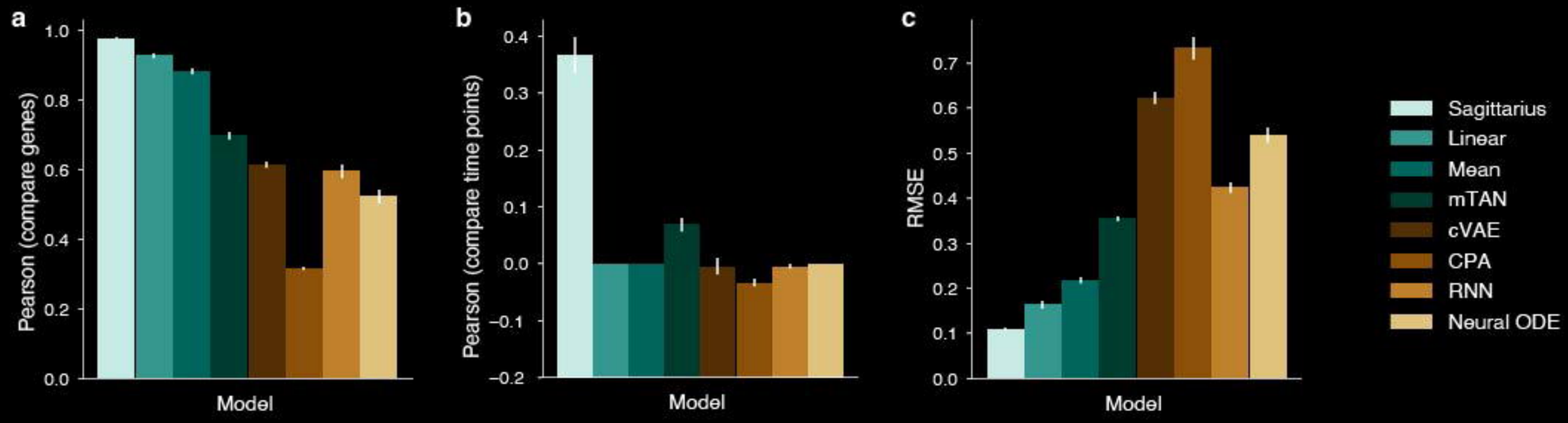




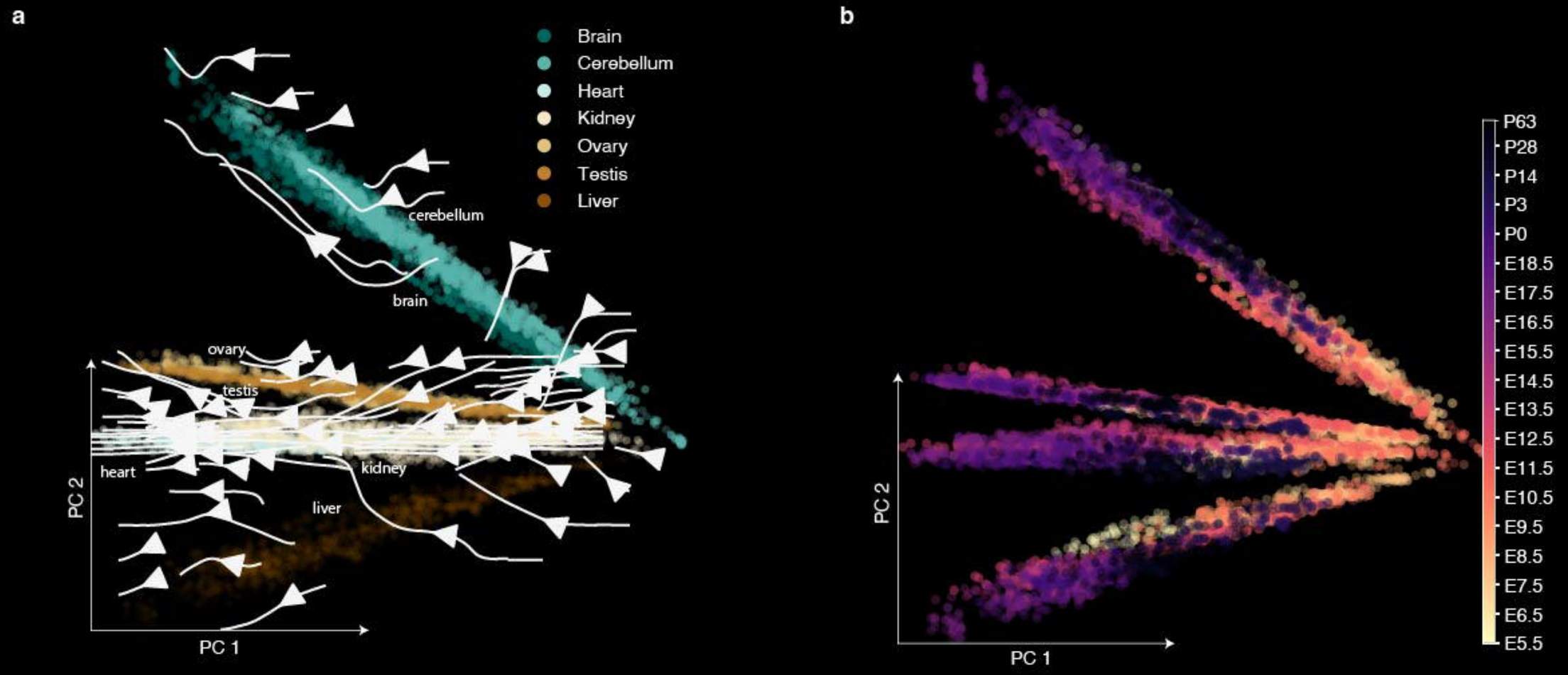




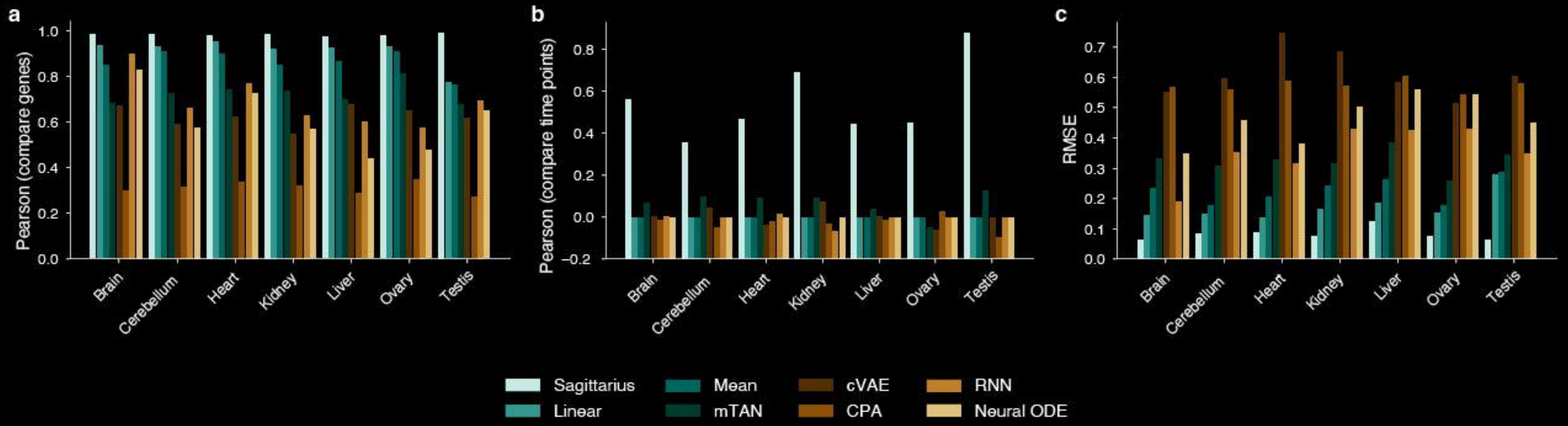
Supplementary Fig. 1



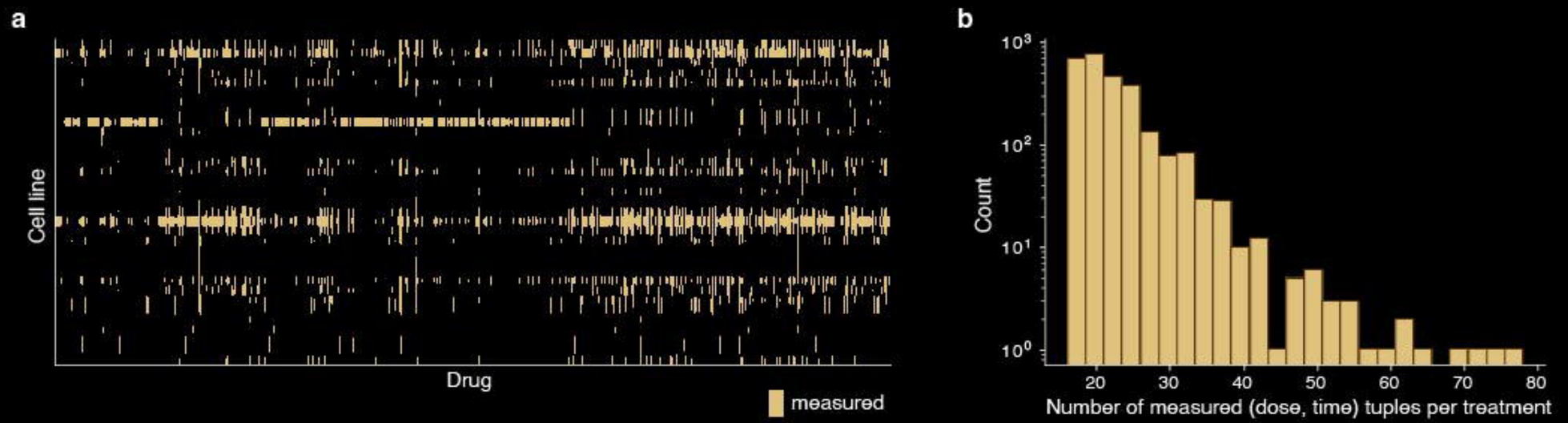
Supplementary Fig. 2



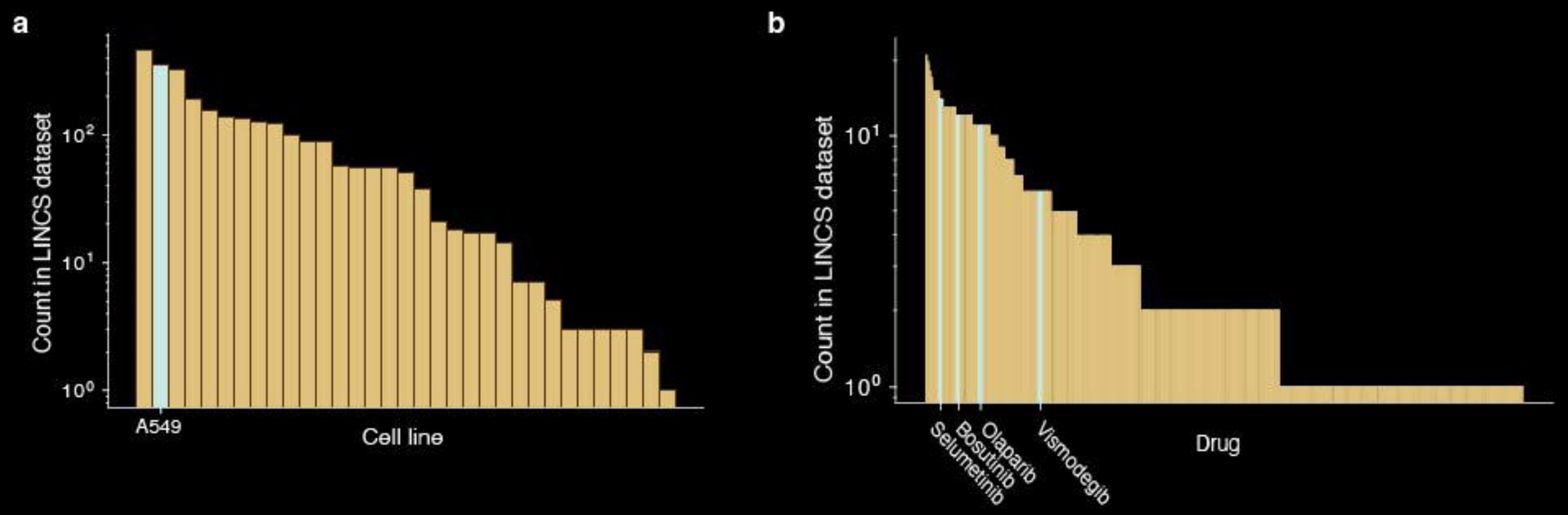
Supplementary Fig. 3



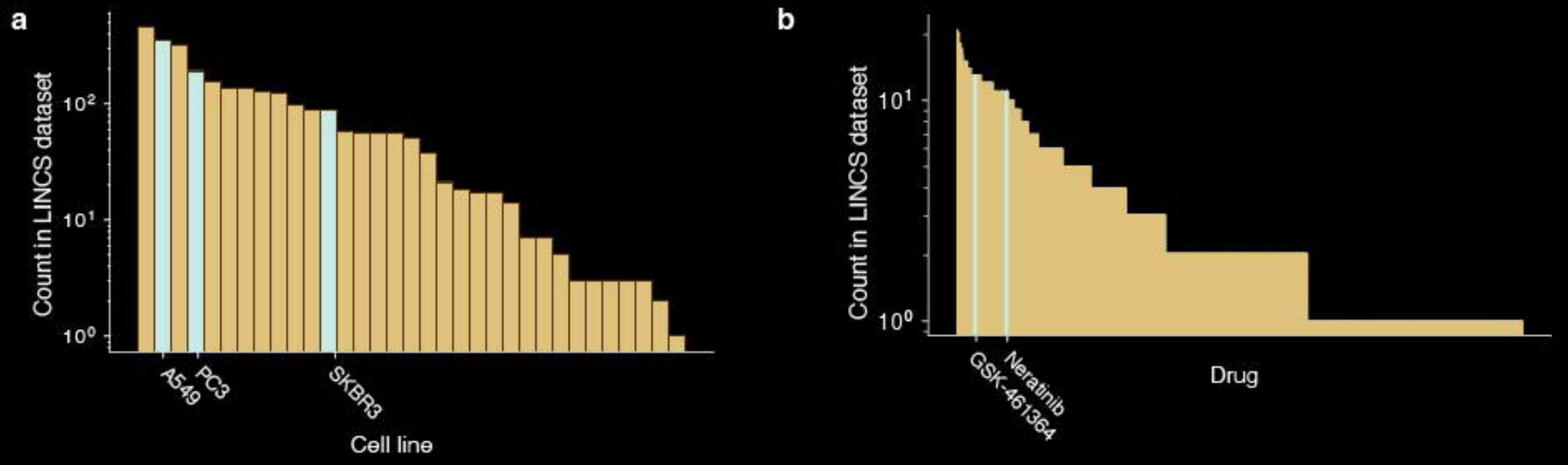
Supplementary Fig. 4



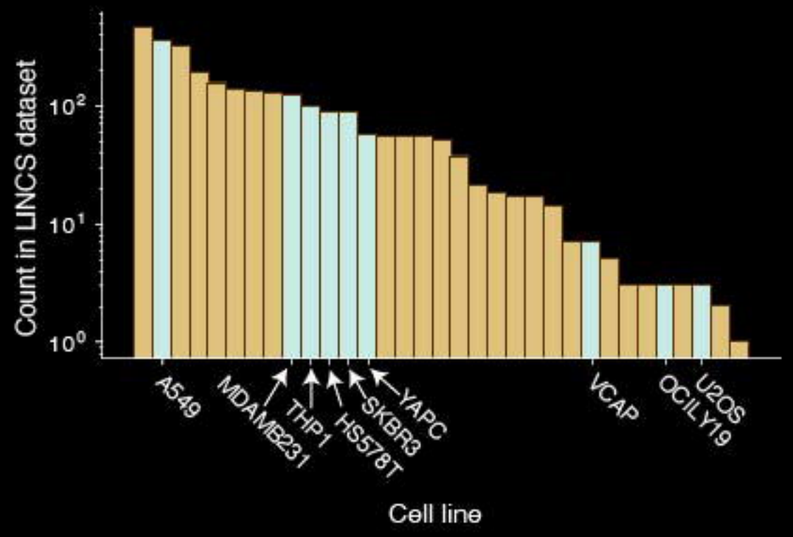
Supplementary Fig. 5



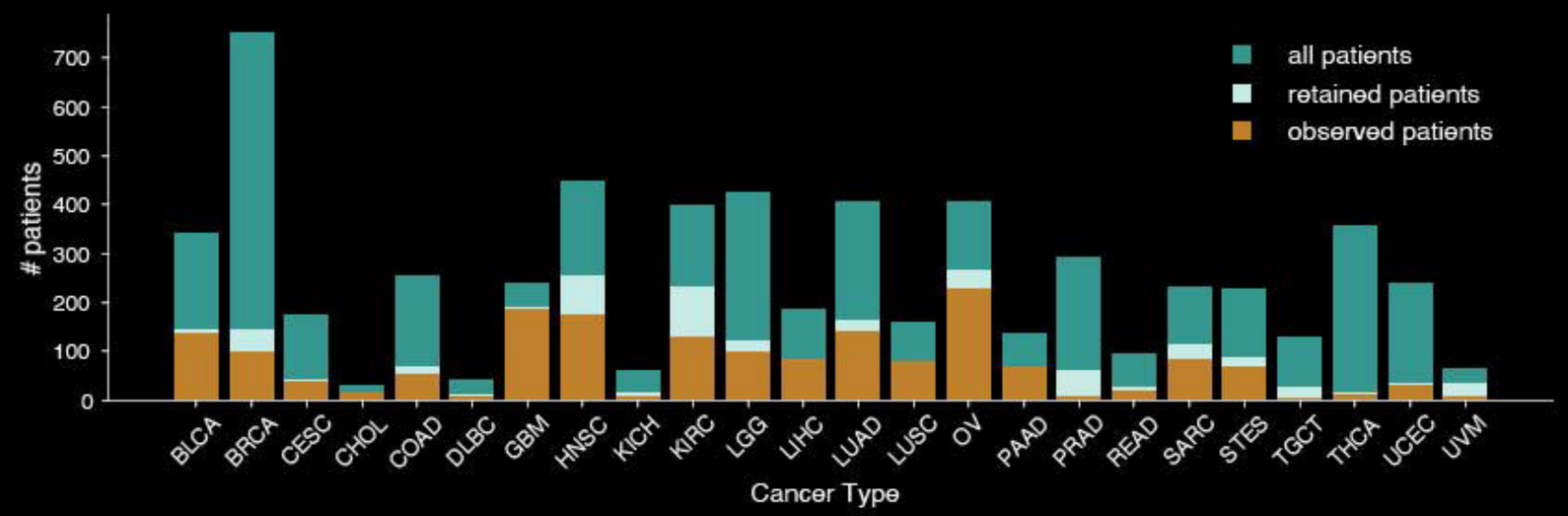
Supplementary Fig. 6



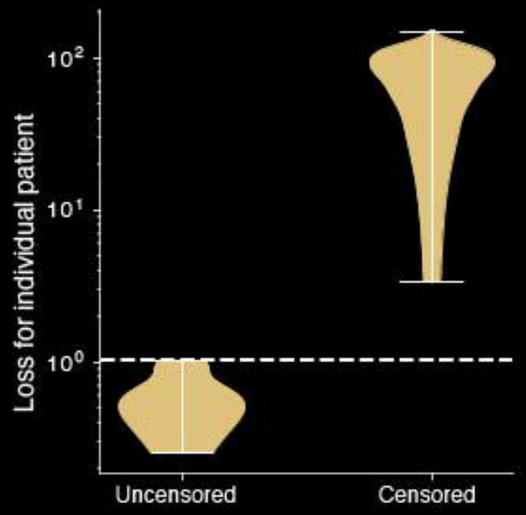
Supplementary Fig. 7



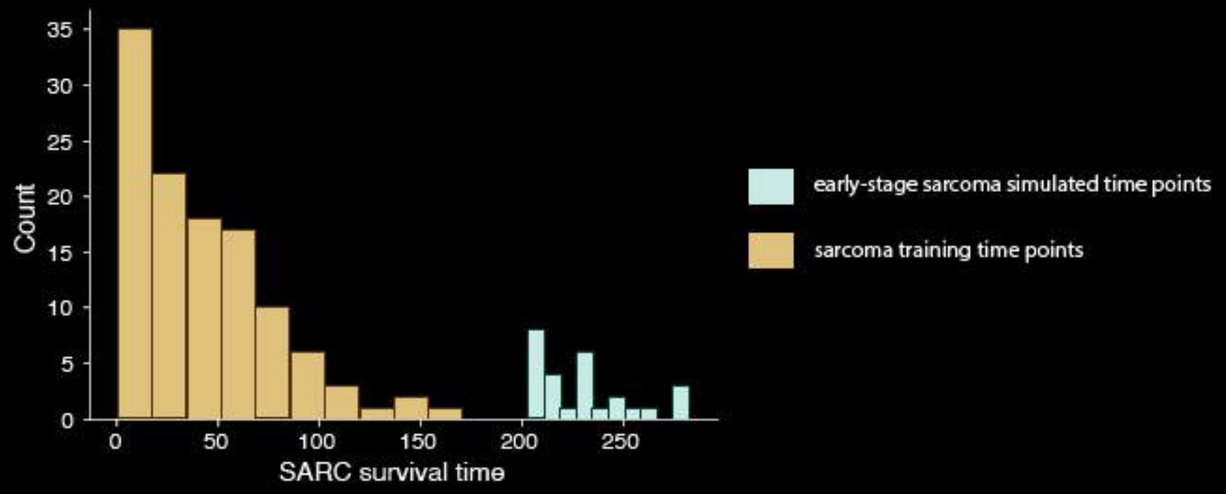
Supplementary Fig. 8



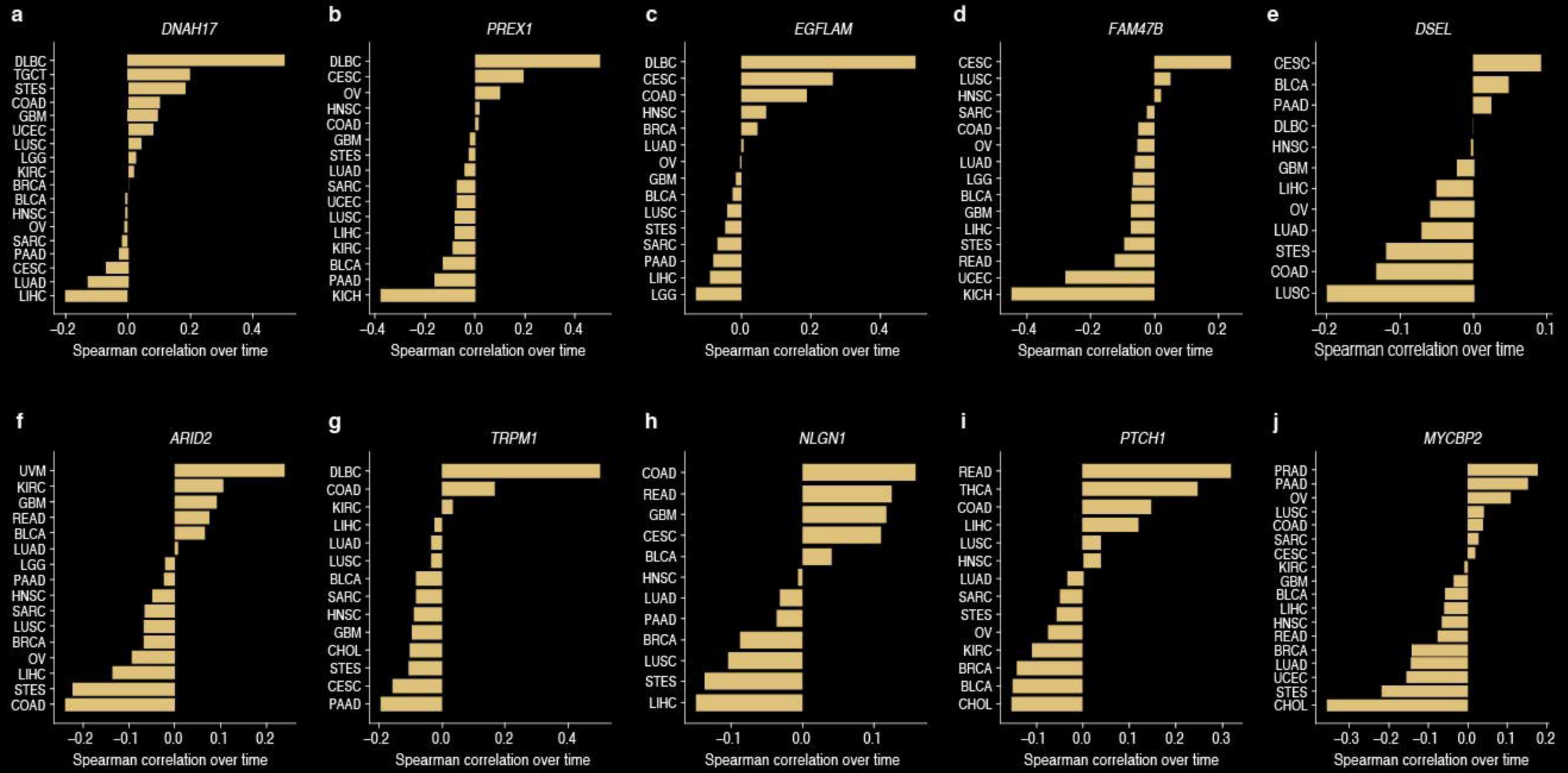
Supplementary Fig. 9



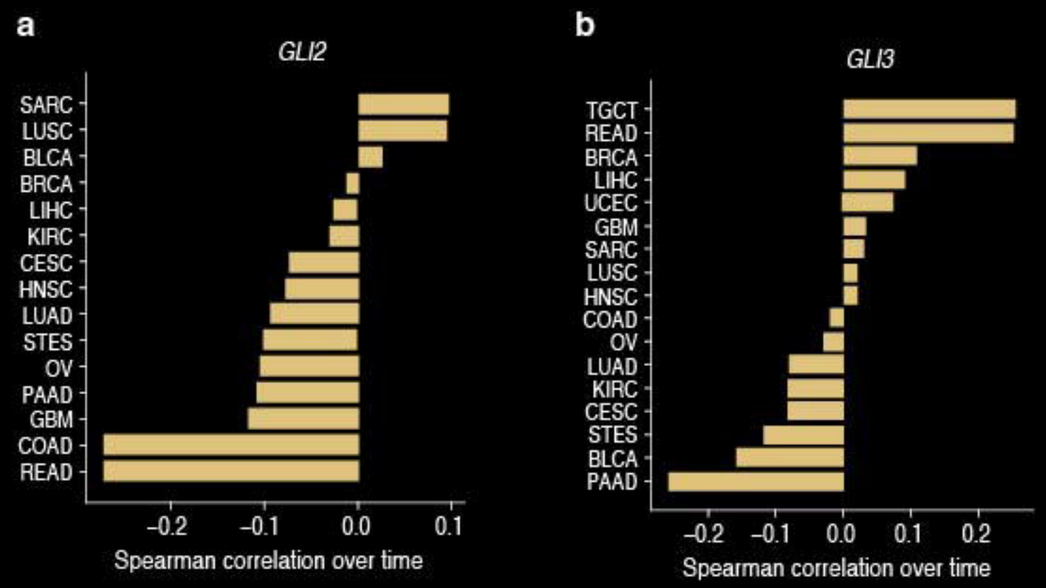
Supplementary Fig. 10



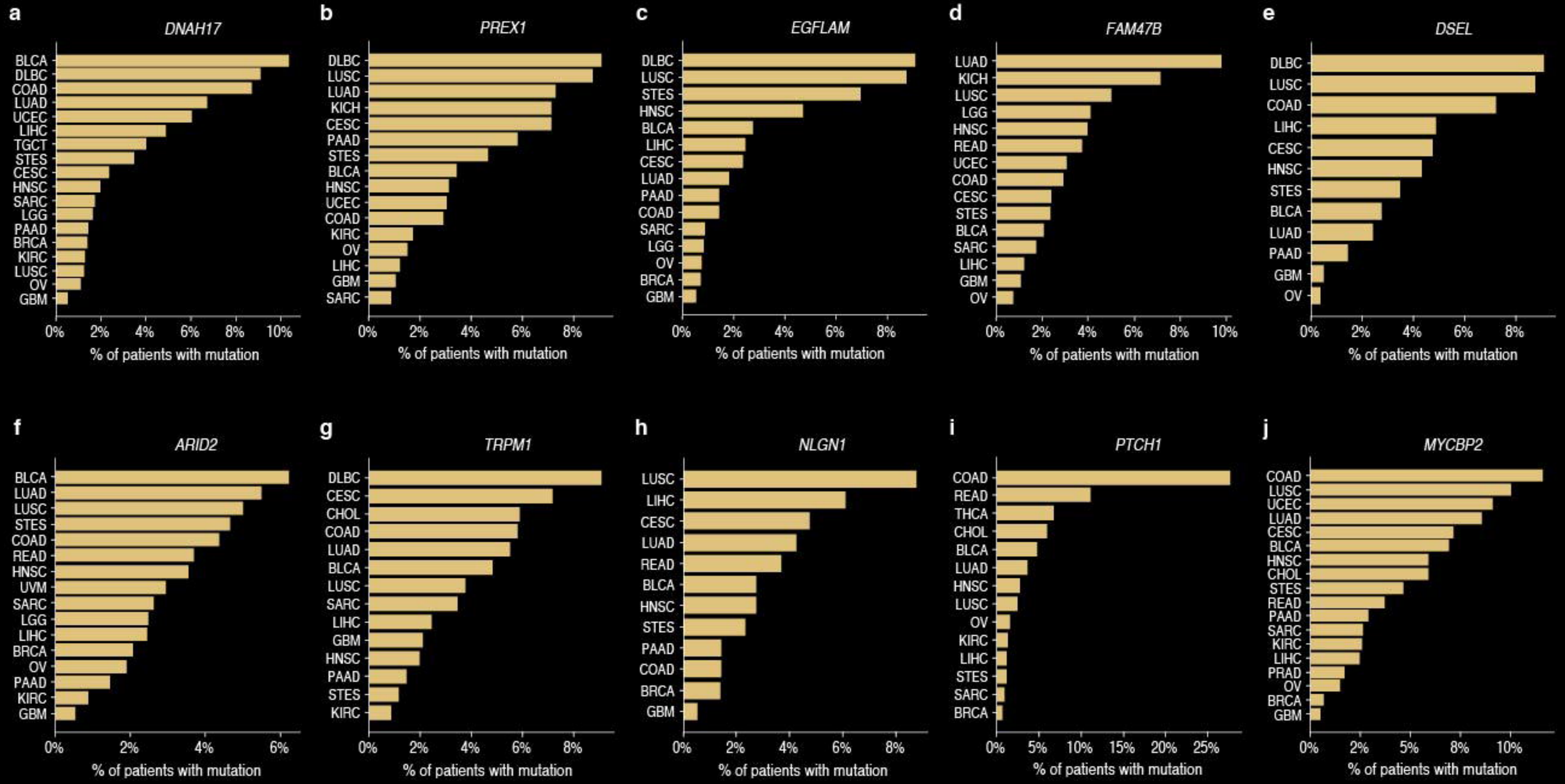
Supplementary Fig. 11



Supplementary Fig. 12



Supplementary Fig. 13



Supplementary Fig. 14