Methods

Fitness landscape inference

Our landscape inference method consists of three steps, followed by a final validation procedure as outlined below. The input for the inference method is a matrix Y, with elements y_{it} which represent the number of times the i^{th} karyotype appears in the longitudinal sample at time t. It will also be useful to also define here the matrix U as the normalized version of Y where each column is divided by its sum.

Frequent karyotypes

"Frequent karyotypes" are operationally defined throughout the manuscript as those observed in at least N cells across all sampled time points, where N is a user supplied hyperparameter. We consider that Y is generated via sampling from a latent population growing exponentially such that:

$$v_{it} = v_0 e^{f_i t}, \tag{1}$$

where v_{it} is the number of cells in the latent population with karyotype i at time t, and f_i is their fitness. These are normalized to form our prediction matrix $\hat{U}$, with elements:

$$\hat{u}_{it} = v_{it} / \sum_i v_i. \tag{2}$$

If $\hat{U}$ is sampled according to a binomial process to generate Y, then the probability of generating each y_{it} is:

$$l_{it} = P(y_{it}; \hat{u}_{it}, \sum_i y_{it}). \tag{3}$$

Defining S the set of indices belonging to frequent karyotypes, we find values of $f_{i \in S}$ and v_0 which maximise $\sum_{i \in S} \sum_t log\ l_{it}$, i.e. the log likelihood, using a multi-start procedure and standard gradient descent methods.

Nearest neighbours

This step of the inference pipeline estimates the fitness of all the Von Neumann neighbours of the frequent clones estimated in the prior step. The growth of these neigbhbour karyotypes is governed by:

$$\frac{dv_i}{dt} = f_i v_i + \sum_j P(\alpha_i | \alpha_j) f_j v_j, \tag{4}$$

where the second term represents influx due to missegregations from the frequent karyotypes. $P(\alpha_i|\alpha_j)$ is the probability that the karyotype α_j missegregates to generate a cell with karyotype α_i (see [**?**, **?**] for details). 4 is solved numerically for the sampled timepoints, then normalized to find $\hat{u}_{it}$. The likelihood is computed similarly to the frequent clones, using:

$$l_{it} = P(y_{it}; \hat{u}_{it}, \sum_i y_{it}) + \sum_j Q(f_i - f_j), \tag{5}$$

where $Q(f_i - f_j)$ is a prior distribution for the fitness gradient between karyotypes (taken to be normal). The parameters of the prior are estimated from the previously estimated frequent karyotyopes. Thus we find values of $f_{i \notin S}$ and v_0 which maximise $\sum_{i \notin S} \sum_t log\ l_{it}$.

All other karyotypes

Kriging (gaussian process regression) was used to infer fitness of all other karyotypes. Kriging was implemented using the Krig function in the R package fields. Karyotypes of all frequent clones and their Von Neumann neighbours were used as the input matrix of independent variables and their accompanying fitness estimates were used as the vector of dependent variables. The model was fit with no drift component (function parameter m=1), otherwise all arguments to the Krig function were set to their defaults.

Cross validation procedure

.

We developed a leave-one-out cross validation procedure as a metric to judge the performance of our inference pipeline. For this procedure the fitness of the frequent karyotypes is estimated as normal. Since these fitness estimates $f_{i \in S}$ are relatively robust we treat them as a ground truth for the cross validation procedure. Then, steps 2 and 3 of the pipeline are repeatedly applied while leaving out each of the frequent karyotypes in turn. Each pipeline trained on the reduced data set is then used to predict the fitness of the omitted clone, resulting in a second set of fitness estimates for the frequent karyotypes $g_{i \in S}$. Finally the R^2 value between $f_{i \in S}$ and $g_{i \in S}$ is calculated and used as the metric to evaluate the inference pipeline performance.

Agent Based Model

We developed a stochastic, agent-based model of karyotype evolution. The model treats cells as individual agents with their own karyotype and fitness value. The model advances in fixed time steps δt. At each timestep, cells may divide at a rate determined by their karyotype specific fitness value (which is treated directly as division rate). Dividing cells may mis-segregate, with the probability of each chromosome of a dividing cell mis-segregating to a specified as specified previously [**?**, **?**]. The ABM features exponential growth and no cell death. To maintain a finite size population, we implemented a 'bottleneck' procedure. We measured population size at each timestep and if the population size exceeded a fixed threshold, we would select a fixed fraction of cells at random (typically 90%) and immediately remove them from the simulation. This mimics cell growth in culture conditions, where cells grow exponentially in ideal conditions then are passaged to remove a majority of cells before confluence is reached. The model was developed in C++ using standard library headers.

Gaussian random fields

Gaussian random fields (GRF) were used to generate artificial fitness landscapes of varying complexity. Each GRF is a continuous field resulting from mapping each point in K dimensional karyotype space to a fitness value. GRF are defined here by a matrix R with rows r_i corresponding to points in the K dimensional space. Each r_i acts as a point source for a wave of wavelength λ, the interference pattern of which creates the GRF. Thus for a karyotype α in the K dimensional space, fitness is evaluated according to:

$$\sum_i sin(\frac{||r_i - \alpha||}{\lambda}). \tag{6}$$