Details of protein turnover calculation algorithm

In order to determine protein degradation rates, we fit the experimental data of all the proteins and Lys using the matrix exponential function. The ODE for Lys (Eq. 11) and individual proteins (Eq. 9) can be written in matrix form $\frac{d}{dt}\boldsymbol{\theta}(t) = \mathbf{G}\boldsymbol{\theta}(t)$; where the matrix $\mathbf{G}$ has degradation parameters (γ_a , γ_i and γ_U) to be determined. We rearrange the equation (11) and (9) in two different forms as follows:

$$\frac{d\theta_{A_L}}{dt} = \gamma_a\left(\theta_{F_L} - \theta_{A_L}\right) + \sum_{i=1}^{n} \gamma_i \frac{[\eta_i P_i]}{[A]}\left(\theta_{P_{iL}} - \theta_{A_L}\right) + \gamma_U \frac{[\eta_U P_U]}{[A]}\left(\theta_{UL} - \theta_{A_L}\right) \qquad (S1)$$

$$\frac{d\theta_{A_L}}{dt} = \sum_{i=1}^{n} \gamma_i \frac{[\eta_i P_i]}{[A]}\theta_{P_{iL}} - \left(\gamma_a + \sum_{i=1}^{n} \gamma_i \frac{[\eta_i P_i]}{[A]} + \frac{\gamma_U[\eta_U P_U]}{[A]}\right)\theta_{A_L} + \gamma_a\theta_{F_L} + \gamma_U \frac{[\eta_U P_U]}{[A]}\theta_{UL} \qquad (S2)$$

$$\frac{d\theta_{P_{iL}}}{dt} = \gamma_i\left(\theta_{A_L} - \theta_{P_{iL}}\right) \qquad (S3)$$

$$\frac{d\theta_{P_{iL}}}{dt} = \gamma_i\theta_{A_L} - \gamma_i\,\theta_{P_{iL}} \qquad (S4)$$

We construct our optimization problem with

- the function $\boldsymbol{\theta} : \mathbb{R} \mapsto \mathbb{R}^{n+3}$ which is built out of the n functions $\theta_{P_{iL}}$ along with $\theta_{A_L}$and θ_{FL}, θ_{UL}. Without loss of generality, we set $\boldsymbol{\theta}_i$= $\theta_{P_{iL}}$for $1 \le i \le n$ and $\boldsymbol{\theta}_{n+1}$= $\theta_{A_L}$and $\boldsymbol{\theta}_{n+2}$= θ_{FL}, $\boldsymbol{\theta}_{n+3}$= θ_{UL}. The $(n+3)$rd entry is meant to represent the lumped unidentified proteome (U).
- The vector $\boldsymbol{\gamma} \in \mathbb{R}^{n+2}$, defined similarly to $\boldsymbol{\theta}$: $\boldsymbol{\gamma}_i$= γ_i for $1 \le i \le n$ and $\boldsymbol{\gamma}_{n+1} = \gamma_a$ and $\boldsymbol{\gamma}_{n+2} = \gamma_U$.
- The mapping $\mathrm{G} : \mathbb{R}^{n+2} \mapsto \mathbb{R}^{(n+3)\times(n+3)}$ which constructs a matrix from a vector $\boldsymbol{\gamma}$ as,

$$\mathbf{G}(\boldsymbol{\gamma})_{ij} = \begin{cases} -\gamma_i & \text{if } i = j \leq n \\ -\gamma_{\mathrm{a}} - \sum_{i=1}^{n} \frac{[\eta_i P_i]}{[A]}\gamma_i - \frac{[\eta_{\mathrm{U}} P_U]}{[A]}\gamma_U & \text{if } i = j = n+1 \\ \gamma_i & \text{if } i \leq n \text{ and } j = n+1 \\ \frac{[\eta_j P_j]}{[A]}\gamma_j & \text{if } i = n+1 \text{ and } j \leq n \\ \frac{[\eta_{\mathrm{U}} P_{\mathrm{U}}]}{[A]}\gamma_U & \text{if } i = n+1 \text{ and } j = n+3 \\ \gamma_{\mathrm{a}} & \text{if } i = n+1 \text{ and } j = n+2 \\ -\gamma_U & \text{if } i = j = n+3 \\ \gamma_U & \text{if } i = n+3 \text{ and } j = n+1 \\ 0 & \text{otherwise} \end{cases} \tag{S5}$$

When the argument $\boldsymbol{\gamma}$ is unambiguous, we simply write $\mathbf{G}$ instead of $\mathbf{G}(\boldsymbol{\gamma})$.

Now we have $\mathbf{G}\,\boldsymbol{\theta}(t) = \frac{d}{dt}\boldsymbol{\theta}(t)$, and therefore $\boldsymbol{\theta}(t) = e^{t\mathbf{G}}\boldsymbol{\theta}(0)$, where $e^{\mathbf{G}}$ is the matrix exponential operator of a matrix $\mathbf{G}$, and (equation (9) and (11) in the main text) have been aggregated into matrix form. We formalize our minimization problem as

$$\boldsymbol{\gamma}_* = \underset{\boldsymbol{\gamma}\in\mathbb{R}^{n+2}}{\operatorname{argmin}} \sum_{i=1}^{m} \| e^{t_i\mathbf{G}}\boldsymbol{\theta}(0) - \boldsymbol{\theta}(t_i)\|_{\mathbf{W}}^2,$$

where the t_i are the time points for which we have protein data and $\boldsymbol{\theta}(0) = 1$ is our initial condition. The matrix $\mathbf{W} \in \mathbb{R}^{(n+3)\times(n+3)}$, defined as

$$\mathbf{W}_{ij} = \begin{cases} 1 & \text{if } i = j \leq n+2 \\ 0 & \text{otherwise} \end{cases}$$

exists solely to exclude the dimensions of $\boldsymbol{\theta}(t)$ that are constructed from θ_{A_F} and θ_U from calculation of error with the norm $\|\cdot\|_{\mathbf{W}}^2 = \langle \cdot, \mathbf{W}\,\cdot \rangle$.

Though the optimal degradation parameters can be found with a gradient-based search of the n+2 dimensional positive reals $\mathbb{R}^{n+2}$. Due to the mapping $\mathbf{G}$ and the matrix exponential, a closed-form expression for the gradient $\nabla\boldsymbol{\gamma}$ is not available, and thus we avoid gradient-based search to

find the optimal $\boldsymbol{\gamma}_*$. Instead, we propose derivative fitting approach and the details of the algorithm are given below:

1. Fit $\boldsymbol{\theta}(t)$ with a surrogate model $\boldsymbol{\vartheta}(t)$ so that $\|\boldsymbol{\theta}(t_i) - \boldsymbol{\vartheta}(t_i)\|_{\mathbf{W}}$ is minimized over all t_i and both $\boldsymbol{\vartheta}(t)$ and $\frac{d}{dt}\boldsymbol{\vartheta}(t)$ are cheaply available for any $t \in \mathbf{P}$.

   We fit each protein ($\boldsymbol{\theta}_{\boldsymbol{P_{iL}}}$) individually together with Lys ($\boldsymbol{\theta}_{\boldsymbol{AL}}$) data to get both $\boldsymbol{\vartheta}(t)$ and $\frac{d}{dt}\boldsymbol{\vartheta}(t)$.

2. Pick a uniform grid of k points $\widehat{t_i}$ over $[t_1, t_k]$

3. Set $j \coloneqq 1$

4. Set the $(n+3) \times (n+2)$ matrix **M** as

$$\mathbf{M}(t)_{ij} \coloneqq \begin{cases} \boldsymbol{\vartheta}_{\boldsymbol{n+1}}(t) - \boldsymbol{\vartheta}_{\boldsymbol{i}}(t) & \text{if } i = j \leq n \\ \frac{[\eta_i p_i]}{[A]}\left(\boldsymbol{\vartheta}_{\boldsymbol{i}}(t) - \boldsymbol{\vartheta}_{\boldsymbol{n+1}}(t)\right) & \text{if } i = n+1 \text{ and } j \leq n \\ \boldsymbol{\vartheta}_{\boldsymbol{n+2}}(t) - \boldsymbol{\vartheta}_{\boldsymbol{n+1}}(t) & \text{if } i = n+1 \text{ and } j = n+1 \\ 0 & \text{otherwise} \end{cases}$$

5. Let

$$h(x) = \begin{cases} x^2 & \text{if } x < 0 \\ 0 & \text{otherwise} \end{cases}$$

6. set

$$\boldsymbol{\gamma}_0, u \coloneqq \underset{\substack{\boldsymbol{\gamma} \in \mathbb{R}^{n+2} \\ u \in \mathbb{R}^k}}{\operatorname{argmin}} \sum_{i=1}^{k} \left\| \mathbf{M}(\widehat{t_i})\boldsymbol{\gamma} - \frac{d}{dt}\boldsymbol{\vartheta}(\widehat{t_i}) + u_i \frac{[A]}{[\eta_{\mathrm{U}}]} \right\|_{\mathbf{W}}^2$$

   where $\boldsymbol{\gamma}_0, u$ are solved approximately with an iterative method.

7. Let $\boldsymbol{\delta}_{n+3}$ be the Kronecker delta vector which is zero everywhere except at entry n+3

8. Set the (n+3)rd entry of $\boldsymbol{\gamma}_0$ to zero. Let $\boldsymbol{S} \coloneqq \{\psi(t) = e^{\widehat{t_\iota}\mathbf{G}(\boldsymbol{\gamma}_0+\boldsymbol{\delta}_{n+3}\,g)}\boldsymbol{\theta}(0) : g \geq 0\}$ be the set of protein dynamics obtained by perturbing γ_{UIP} and find the θ_U that maximizes correlation with $\mathbf{u}$

$$\theta_{\mathrm{UIP}} \coloneqq \underset{\psi\in\boldsymbol{S}}{\operatorname{argmax}} \cos\angle(\mathbf{u},\psi) = \underset{\psi\in\boldsymbol{S}}{\operatorname{argmax}} \frac{\sum_{i=1}^{k} u_i\psi(\widehat{t_\iota})}{(\sum_{i=1}^{k} u_i^2)(\sum_{i=1}^{k} \psi(\widehat{t_\iota})^2)}$$

9. set $\boldsymbol{\gamma}_0 \coloneqq \boldsymbol{\gamma}_0 + \boldsymbol{\delta}_{n+3}\,\gamma_{\mathrm{UIP}}$ where γ_{UIP} is the perturbation that produced θ_{UIP} and set

   $\epsilon_j \coloneqq \sum_{i=1}^{m} \left\|\boldsymbol{\theta}(t_i) - e^{t_i\mathbf{G}(\boldsymbol{\gamma}_0)}\boldsymbol{\theta}(0)\right\|_{\mathbf{W}}^2$

10. if $j = 1$ or $\epsilon_j < \epsilon_{j-1}$, set $j \coloneqq j + 1$, and goto 6, using $\boldsymbol{\gamma}_0$ as an initial guess for the iterative solver used on line 6.

11. return $\boldsymbol{\gamma}_0$

There are some helpful clarifying remarks to be made about Algorithm 1:

- We use a grid spacing of 10^{-1} for the $\widehat{t_\iota}$ in line 2.
- The matrix $\mathbf{M}(t)$ on line 4 is a rearrangement of $\mathbf{G}\,\boldsymbol{\theta}(t) = \frac{d}{dt}\boldsymbol{\theta}(t)$, so it satisfies

$$\mathbf{M}(t)\boldsymbol{\gamma} = \frac{d}{dt}\boldsymbol{\theta}(t)$$

- The vector u_i on line 6 is meant to represent a combination of noise and the derivative of the unidentified protein for which no dynamics history is available except for at $t = 0$. T
- Solving the minimization problem on line 6 is done with constrained iterative optimization to enforce $\boldsymbol{\gamma} \in \mathbb{R}^{n+2}$. The trust-region method we used occasionally encountered difficulty satisfying all constraints without taking many hundreds of iterations.

- The penalty term $\tau\left(h(u_i)\right)^2$ in the minimization problem on line 6 is intended to penalize $u(t)$ that have infeasible dynamics. The parameter τ is chosen to be the largest possible $\tau \in \mathbb{R}$ such that the constraints in the previous remark are not violated.

- Steps 7--9 find the gamma that induces protein dynamics that most closely fit u_i.

The parameters exploits the derivative relationship $\mathbf{G\theta}(t) = \frac{d}{dt}\boldsymbol{\theta}(t)$**,** where the observed protein dynamics $\boldsymbol{\theta}(t)$ are replaced with a surrogate model $\boldsymbol{\vartheta}(t)$ that is fit to the protein dynamics. The matrix $\mathbf{M}(t)$ simply rearranges $\mathbf{G\theta}(t) = \frac{d}{dt}\boldsymbol{\theta}(t)$ so that the vector $\boldsymbol{\gamma}$ is on the right-hand side of a matrix with known values. We find the $\boldsymbol{\gamma}_0$ that minimizes the L2-norm of the residual over all synthetic points $\widehat{t_i}$.