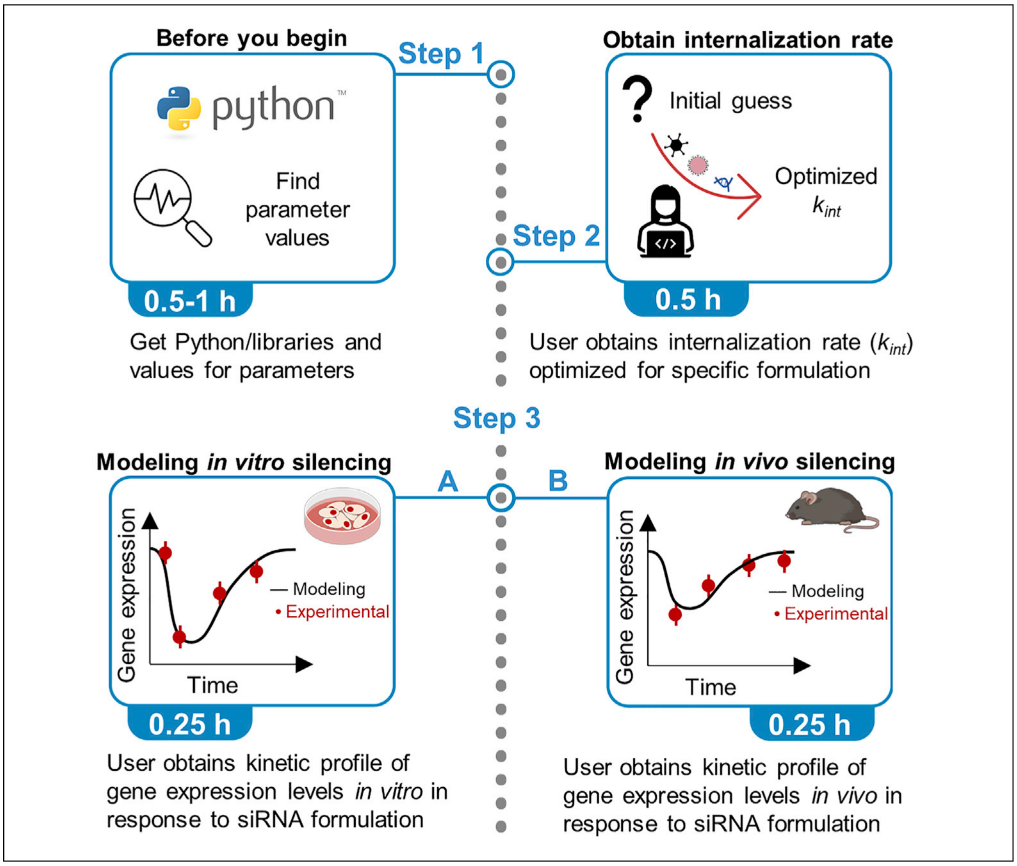
Please cite this article in press as: Roh et al., A kinetic modeling platform for predicting the efficacy of siRNA formulations *in vitro* and *in vivo*, STAR Protocols (2022), https://doi.org/10.1016/j.xpro.2022.101723

STAR Protocols

CellPress OPEN ACCESS

Protocol

A kinetic modeling platform for predicting the efficacy of siRNA formulations *in vitro* and *in vivo*


Before you begin
python™
Find parameter values
0.5-1 h
Get Python/libraries and values for parameters
Step 1
Obtain internalization rate
Initial guess
Optimized k_{int}
Step 2
0.5 h
User obtains internalization rate (k_{int}) optimized for specific formulation
Step 3
A
B
Modeling *in vitro* silencing
Gene expression
Time
— Modeling
• Experimental
0.25 h
User obtains kinetic profile of gene expression levels *in vitro* in response to siRNA formulation
Modeling *in vivo* silencing
Gene expression
Time
— Modeling
• Experimental
0.25 h
User obtains kinetic profile of gene expression levels *in vivo* in response to siRNA formulation

Esther H. Roh, Millicent O. Sullivan, Thomas H. Epps, III

msullivan@udel.edu (M.O.S.)
thepps@udel.edu (T.H.E.)

Highlights

- Protocol to predict siRNA formulation efficacy using data from a single time point
- Applicable to various cell types and nanocarriers
- Translation of *in vitro* experimental silencing data to predict *in vivo* results

We present a computational modeling protocol that can accurately predict changes in both *in vitro* and *in vivo* gene expression levels in response to the application of various siRNA formulations. We describe how to use this Python-based pipeline to obtain crucial information, namely maximum silencing level and duration of silencing, toward the design of therapeutically relevant dosing regimens. The protocol details the steps for running internalization rate fitting to produce predictions based on experimental measurements from a single time point.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Roh et al., STAR Protocols 3, 101723
December 16, 2022 © 2022 The Author(s).
https://doi.org/10.1016/j.xpro.2022.101723

Protocol

A kinetic modeling platform for predicting the efficacy of siRNA formulations *in vitro* and *in vivo*

Esther H. Roh,[1,5] Millicent O. Sullivan,[1,2,*] and Thomas H. Epps, III[1,3,4,6,*]

[1]Department of Chemical and Biomolecular Engineering, University of Delaware, Newark, DE 19716, USA

[2]Department of Biomedical Engineering, University of Delaware, Newark, DE 19716, USA

[3]Center for Research in Soft matter and Polymers (CRiSP), University of Delaware, Newark, DE 19716, USA

[4]Department of Materials Science and Engineering, University of Delaware, Newark, DE 19716, USA

[5]Technical contact: estherhr@udel.edu

[6]Lead contact

*Correspondence: msullivan@udel.edu (M.O.S.), thepps@udel.edu (T.H.E.)
https://doi.org/10.1016/j.xpro.2022.101723

SUMMARY

We present a computational modeling protocol that can accurately predict changes in both *in vitro* and *in vivo* gene expression levels in response to the application of various siRNA formulations. We describe how to use this Python-based pipeline to obtain crucial information, namely maximum silencing level and duration of silencing, toward the design of therapeutically relevant dosing regimens. The protocol details the steps for running internalization rate fitting to produce predictions based on experimental measurements from a single time point.

For complete details on the use and execution of this protocol, please refer to Roh et al., 2021.

BEFORE YOU BEGIN

The protocol below describes the specific steps and values necessary to predict green fluorescent protein (GFP) silencing in B16F10 cells using electrotransfection *in vitro* and apolipoprotein B (apoB) silencing in mouse models using siRNA-lipid nanoparticles. However, we have also used this protocol for various scenarios with different vectors (e.g., naked siRNA, jetPRIME®, Lipofectamine™, Oligofectamine™, chitosan, polymer conjugates, photo-responsive polymers, multifunctional envelope-type nanodevices, calcium phosphate nanoparticles), different target genes (e.g., Luciferase, GFP, glyceraldehyde 3-phosphate dehydrogenase, interleukin 1 beta, voltage dependent anion channel 1, PTEN), and different cell types (e.g., B16F10, Neuro2A, HeLa, LNCaP, H1299, A549).

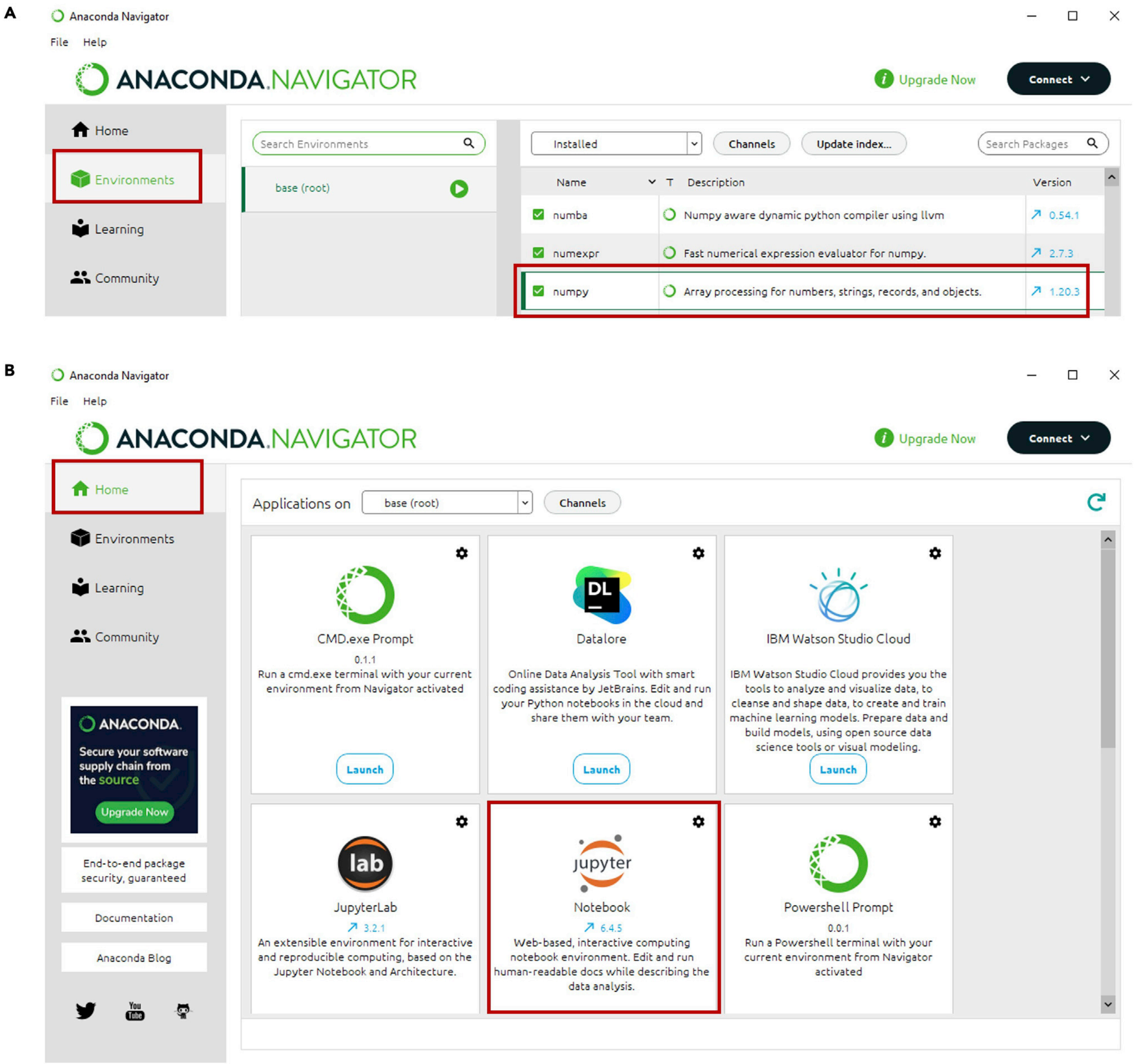
The ability to simulate numerous experimental gene silencing scenarios, as described in this protocol, has not been demonstrated previously with other RNA interference (RNAi) models.

Downloading and launching Python platform

Timing: 0.5–1 h

This section includes the minimal hardware requirements, software requirements, and input parameters needed to run the model.

1. **Hardware:** Personal computer with a minimum of 5 GB free disk space and 32 GB random-access memory.



Name	T	Description	Version
☑ numba		Numpy aware dynamic python compiler using llvm	0.54.1
☑ numexpr		Fast numerical expression evaluator for numpy.	2.7.3
☑ numpy		Array processing for numbers, strings, records, and objects.	1.20.3

CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated

Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

JupyterLab
3.2.1
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.4.5
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated

Figure 1. Launching Anaconda and preparing required Python packages

(A) After launching Anaconda, make sure that the appropriate Python packages are available in the 'Environments' tab.

(B) Once all packages are available, launch Jupyter Notebook in the 'Home' tab.

2. **Software requirements:** A convenient method to run Python on any operating system is to use Anaconda, an open-source distribution of Python for scientific computing.
   a. Anaconda can be downloaded from https://www.anaconda.com/ according to individual computer specifications.
   b. Once Anaconda Distribution - Python v.3.9 is installed, open the Anaconda Navigator, click on the 'Environments' tab to the left, and make sure the following Anaconda packages are installed: notebook v.6.0.3, numpy v.1.18.1 (Harris et al., 2020), scipy v.1.4.1 (Virtanen et al., 2020), scikit-learn v.0.22.1 (Pedregosa et al., 2011), and matplotlib v.3.1.3 (Hunter, 2007). All of the packages mentioned should be pre-installed in the base environment (Figure 1A).

c. Click on the ‘Home’ tab to the left and launch Jupyter Notebook (Figure 1B).

Preparing input parameters

⏱ Timing: 0.5–72 h

3. To employ the kinetic model provided in this protocol, users are required to provide the values for the following parameters.
   a. siRNA half-life
   b. mRNA half-life
   c. protein half-life
   d. cell doubling time
   e. transfection volume
   f. transfection time
   g. transfection concentration
   h. *in vitro* experimental measurements of gene expression levels at one or more time points, preferably between 48–72 h

Note: There are numerous protocols available from either commercial websites (e.g., Thermo Fisher Scientific, Dharmacon) or peer-reviewed literature (Sakurai et al., 2010) that detail steps for obtaining this type of experimental measurement.

KEY RESOURCES TABLE

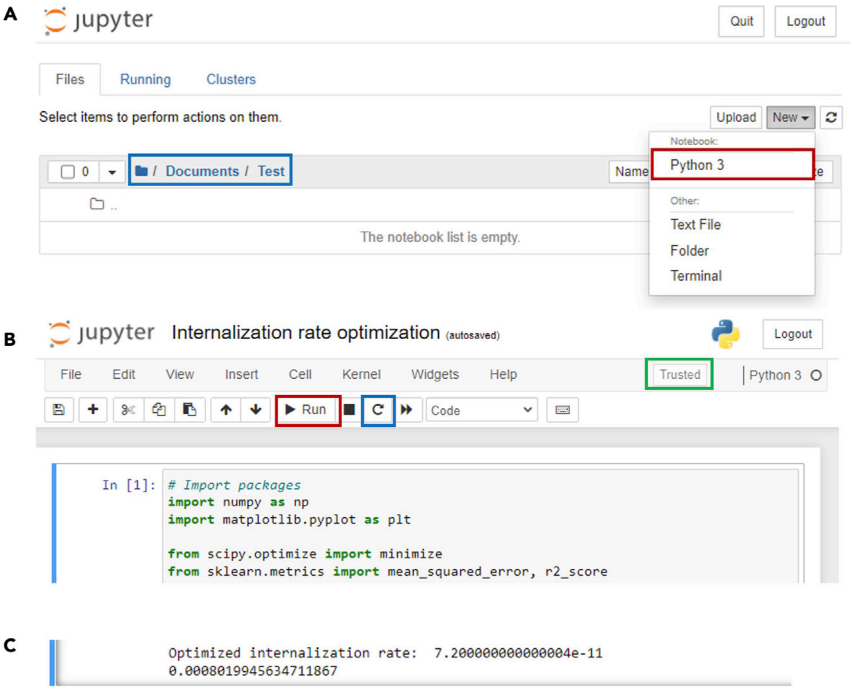
REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
Anaconda	Anaconda Inc.	https://www.anaconda.com/
Python	Anaconda Inc.	https://www.anaconda.com/
Fitting internalization rate	This paper	N/A
In vitro model	This paper	N/A
In vivo model	This paper	N/A
Other		
Internalization rate of external siRNA (k_{int})	This paper	N/A
Transfection volume (V_{ex})	Specified experimentally	N/A
Intracellular volume (V_{in})	(Wittrup et al., 2015; Bartlett and Davis, 2006)	Vi
Dilution rate of siRNA due to cell division ($k_{s,dil}$)	Calculated from cell doubling time	N/A
Intracellular degradation rate of siRNA ($k_{s,deg}$)	(Bartlett and Davis, 2006)	kdeginna
Intracellular RISC concentration ($RISC_{tot}$)	(Bartlett and Davis, 2006)	rtot
Formation rate of active RISC (k_{RISC})	(Bartlett and Davis, 2006)	kformRISC
Intracellular RISC degradation rate ($k_{r,deg}$)	(Bartlett and Davis, 2006)	kdegRISC
Intracellular mRNA production rate (k_{mRNA})	Calculated from mRNA half-life	N/A
Intracellular mRNA degradation rate ($k_{m,deg}$)	Calculated from mRNA half-life	N/A
Cleavage rate of mRNA (k_{cleav})	(Roh et al., 2021)	k_{cleav}
Intracellular protein production rate (k_{prot})	Calculated from protein half-life	N/A
Intracellular protein degradation rate ($k_{p,deg}$)	Calculated from protein half-life	N/A

STEP-BY-STEP METHOD DETAILS

Fitting for internalization rate

⏱ Timing: 15–30 min

⚠ CRITICAL: To proceed with the following steps, make sure to first install all software and packages listed in ‘Software requirements.’



```
In [1]: # Import packages
        import numpy as np
        import matplotlib.pyplot as plt

        from scipy.optimize import minimize
        from sklearn.metrics import mean_squared_error, r2_score
```

```
Optimized internalization rate:  7.20000000000000004e-11
0.0008019945634711867
```

Figure 2. Fitting for an optimized internalization rate

(A) After opening Jupyter Notebook, choose the correct directory in which to save your notebook (blue box). Click on the 'New' box and create a new Python 3 notebook (red box).

(B) After copy and pasting all of the code provided in the 'fitting for internalization rate' section, run code (red box). If there is an error and users must re-run the code for any reason, make sure to reset the kernel (blue box) before re-running. Make sure that the kernel is trusted (green box).

(C) Once the code is finished, the optimized internalization rate and root-mean-square error (RMSE) for that rate are displayed.

One of the core aims of our protocol is to provide users with the ability to easily predict changes in gene expression levels both *in vitro* and *in vivo* upon application of any siRNA formulation of interest. Traditionally, modeling the efficacy of various siRNA formulations is challenging because the RNAi process consists of multiple steps that are difficult to characterize experimentally (Haley and Zamore, 2004; Cuccato et al., 2011; Bartlett and Davis, 2006). In particular, cellular uptake and necessary post-uptake processing, including endosomal escape, comprises multiple steps, and the parameters that describe these steps are unique to each delivery system. In this mathematical framework, we define the internalization rate as a single aggregated parameter so that this rate is the only unknown variable, and thus, can be fit from a single experimental time point. By fitting for an aggregated internalization rate, users can reduce the number of parameters needed to implement the model in various experimental scenarios. This fitting process is performed as follows:

1. To call the packages that will be used for this protocol, copy and paste the following code into Jupyter Notebook (Figure 2A).

   Note: Full code is also available in the ipynb file, 'Fitting for internalization rate.ipynb,' that is included in the electronic information associated with this publication (Data S1).

```
# Import packages

import numpy as np

import matplotlib.pyplot as plt
```

```
from scipy.optimize import minimize

from sklearn.metrics import mean_squared_error, r2_score
```

2. Copy and paste the following code, and modify the values for the seven variables accordingly.

   Note: Cell-doubling rates are needed to correctly account for the dilution of intracellular siRNA over time. The next three parameters are required to compute the component degradation rates and the production rates of mRNA and protein assuming steady-state values in the absence of siRNA. The last three parameters are needed to correctly scale and account for the initial siRNA concentration available to the cells.

```
# Inputs

cell_doubling = 17.0 # cell doubling time (h)

siRNA_half_life = 24.0 # siRNA half-life (h)

mRNA_half_life = 7 # mRNA half-life (h)

prot_half_life = 26 # protein half-life (h)

siRNA0 = 1500 # transfection concentration (nM)

transfection_time = 0.08 # transfection time (h)

V_ex = 100 * 10 ** -6 # transfection volume (L)
```

3. Copy and paste the following code, and define the final time point of interest as the variable "tf_transfection".

   Note: The accuracy of model predictions is enhanced as more steps are taken. However, increasing the number of steps results in an escalation of computation cost (e.g., time). A minimum value of 0.01 is recommended based on characteristic time calculations of the fastest rate (cleavage rate of mRNA = 500 nM^{-1}h^{-1}). (Roh et al., 2021; Wittrup et al., 2015).

```
# Time and steps

t0 = 0

tf_transfection = 200.0

dt = 0.01

t = np.linspace(t0,tf_transfection,int(tf_transfection/dt + dt))

n = len(t)
```

4. Copy and paste the following code.
   a. Input the time points of experimental data as parameters of variable "x".
   b. Input the respective gene expression levels at each time point as parameters of variable "y".

   Note: Data for y-array are defined as protein expression levels post-siRNA exposure that are scaled between the values of 0 and 1.

```
# Experimental measurements

x = [48] # time of measurement (h)

y = [0.49] # gene expression levels
```

5. Copy and paste the following code, and input an initial guess for internalization rate.

   Note: The recommended starting value is 10^{-10} h^{-1}.

```
# Initial guess for internalization rate

k_int_init = 1 * 10 ** -10 # (h^-1)
```

6. Copy and paste the following code, run code, and obtain an optimized internalization rate (Figures 2B and 2C). Troubleshooting 1–4.

   Note: Do not change any values within the code provided in step 6. At the end of the run, the code will automatically report the optimized internalization rate along with the RMSE associated with that rate.

```
# Parameters for volume scaling

V_in = 2000 * 10 ** -15 # Intracellular volume (L)

# Unit conversions

mol_M = 6.022 * 10 ** 23 # (number/mole)

rtot = 1.9 * 10 ** 15 # (number/L)

r_tot = rtot / mol_M * (10 ** 9) # (nM)

# Initial concentrations

z0 = [siRNA0,0,0,100,100] # External siRNA, intracellular siRNA, RISC, mRNA, prot

DNA = 100

# Rates

k_sdeg = 0.69315 / siRNA_half_life # (h^-1)

k_sdil = 0.69315 / cell_doubling # (h^-1)

k_mRNA = 0.69315 / mRNA_half_life # (h^-1)

k_mdeg = 0.69315 / mRNA_half_life # (h^-1)

k_risc_orig = 2 * 10 ** ^-19 # (L/h/number)

k_risc_M = k_risc_orig * mol_M # (M^-1h^-1)

k_risc = k_risc_M * 10 ** -9 # (nM^-1h^-1)

k_rdeg = 0.077 # (h^-1)

k_mRNA = 0.69315 / mRNA_half_life # (h^-1)

k_mdeg = 0.69315 / mRNA_half_life # (h^-1)

k_cleav = 5 * 10 ** 2 # (nM^-1h^-1)

k_prot = 0.69315 / prot_half_life # (h^-1)
```

```
k_pdeg = 0.69315 / prot_half_life # (h^-1)
# Unpopulated lists
siRNA_ex = np.ones(n) * z0[0]
siRNA = np.ones(n) * z0[1]
RISC = np.ones(n) * z0[2]
mRNA = np.ones(n) * z0[3]
prot = np.ones(n) * z0[4]
# Changing time points to array for optimization code
x_array = np.array(x)
x_opt_array = x_array * 100
x_indices_to_access = x_opt_array.tolist()
print(x_indices_to_access)
# Optimization code
def kint_rmse(k_int):
  for i in range(1,n):
    siRNA_ex[i] = (-k_int * siRNA_ex[i-1]) * dt + siRNA_ex[i-1]
    if i > (transfection_time / dt):
       siRNA_ex[i] = 0
    siRNA[i] = (k_int * (siRNA_ex[i-1] * (V_ex/V_in)) - k_sdeg * siRNA[i-1] - k_sdil * siRNA[i-1]) * dt + siRNA[i-1]
    RISC[i] = (k_risc * siRNA[i-1] * (r_tot - RISC[i-1]) - k_rdeg * RISC[i-1]) * dt + RISC[i-1]
    if RISC[i] > r_tot:
       RISC[i] = r_tot
    mRNA[i] = (k_mRNA * DNA - k_mdeg * mRNA[i-1] - k_cleav * RISC[i-1] * mRNA[i-1]) * dt + mRNA[i-1]
    if mRNA[i] < 0:
       mRNA[i] = 0
    prot[i] = (k_prot * mRNA[i-1] - k_pdeg * prot[i-1]) * dt + prot[i-1]
    if prot[i] < 0:
       prot[i] = 0
    prot_norm = prot / z0[4]
    prot_norm_array = np.array(prot_norm)
    y_pred_array = prot_norm_array[x_indices_to_access]
    y_pred = list(y_pred_array)
  rmse = np.sqrt(mean_squared_error(y,y_pred))
  return rmse
# Define empty array for internalization rate
optimized_kint = 0
i = kint_rmse(k_int_init)
```

```
  updated_k_int = k_int_init + (0.1 * 10 ** -11)
j = kint_rmse(updated_k_int)
while i > j:
  print(i)
  k_int_init = k_int_init + (0.1 * 10 ** -11)
  i = kint_rmse(k_int_init)
  updated_k_int = k_int_init + (0.1 * 10 ** -11)
  j = kint_rmse(updated_k_int)
  if i <= j:
     print(k_int_init)
     optimized_kint = k_int_init
     break
print("Optimized internalization rate: ", optimized_kint)
print(kint_rmse(optimized_kint))
```

Modeling siRNA-induced changes in gene expression levels *in vitro*

⏲ Timing: 15 min

⚠ CRITICAL: To proceed with the following steps, make sure to first obtain an optimized internalization rate using the procedure listed in 'fitting for internalization rate.'

To expand the applicability of the model described in this protocol, the number of parameters required to accurately predict gene silencing *in vitro* following the delivery of an siRNA formulation is further reduced by including only the most important parameters within the mathematical framework to get quantitative agreement with experimental data. Major rate-limiting kinetic steps for the four main components of the RNAi process (i.e., siRNA, RNA-induced silencing complex (RISC), mRNA, and protein) are identified from literature and incorporated into the model. The parameters that are considered for each component and the values of these parameters are listed in Tables 1 and 2. In the paragraph below, the rationale for choosing those specific kinetic rates for *in vitro* modeling of siRNA-induced gene silencing is detailed.

$$\frac{d[siRNA_{in}]}{dt} = k_{int} * \frac{V_{ex}}{V_{in}} * [siRNA_{ex}] - k_{s,dil}[siRNA_{in}] - k_{s,deg}[siRNA_{in}] \quad \text{(Equation 1)}$$

$$\frac{d[RISC]}{dt} = k_{RISC}[siRNA_{in}][RISC_{tot} - RISC] - k_{r,deg}[RISC] \quad \text{(Equation 2)}$$

$$\frac{d[mRNA]}{dt} = k_{mRNA}[DNA] - k_{m,deg}[mRNA] - k_{cleav}[mRNA][RISC] \quad \text{(Equation 3)}$$

$$\frac{d[prot]}{dt} = k_{prot}[mRNA] - k_{p,deg}[prot] \quad \text{(Equation 4)}$$

As noted in the previous section, the increase in intracellular siRNA concentration is defined to depend solely on a single aggregated internalization rate (k_{int}, Equation 1). Additionally, the amount of intracellular siRNA is affected by *(i)* dilution of the local intracellular siRNA concentration as cells divide ($k_{s,dil}$, Equation 1); and *(ii)* siRNA degradation ($k_{s,deg}$, Equation 1). To model the RISC concentration, we assume that the concentration of active RISC, e.g., the species that can recognize and

Table 1. Model variables

Variable (name)	Description (Units)
$siRNA_{ex}$	Extracellular siRNA (nM)
$siRNA_{in}$	Intracellular siRNA (nM)
RISC	Activated RISC (RISC + siRNA) (nM)
mRNA	Intracellular mRNA (nM)
DNA	Intracellular DNA (nM)
prot	Intracellular protein (nM)

Reprinted with permission from (Roh et al., 2021). Copyright 2021 American Chemical Society.

cleave mRNA, depends solely on the binding of free siRNA to RISC (k_{RISC}, Equation 2) and RISC degradation ($k_{r,deg}$, Equation 2). Although previous studies have shown that RISC has multiple intermediates, (Pratt and MacRae, 2009) our assumption is reasonable because the formation of the siRNA-RISC intermediate has the slowest kinetic rate of all steps in active RISC formation (Pratt and MacRae, 2009). Meanwhile, the mRNA concentration is modeled to depend on innate mRNA production (k_{mRNA}, Equation 3), degradation ($k_{m,deg}$, Equation 3), and cleavage (k_{cleav}, Equation 3) rates. Finally, protein concentration depends directly on protein production from the mRNA template (k_{prot}, Equation 4), along with protein degradation ($k_{p,deg}$, Equation 4).

7. To call the packages that will be used for this protocol, copy and paste the code provided below into Jupyter Notebook (Figure 3A).

   Note: Full code is also available in the ipynb file, 'Modeling siRNA-induced changes in gene expression levels in vitro.ipynb,' that is included in the electronic information associated with this publication (Data S2).

```
# Import packages

import numpy as np

import matplotlib.pyplot as plt
```

8. Copy and paste the following code, and modify the values for the seven variables accordingly.

```
# Inputs

cell_doubling = 17.0 # cell doubling time (h)

siRNA_half_life = 24.0 # siRNA half-life (h)

mRNA_half_life = 7 # mRNA half-life (h)

prot_half_life = 26 # protein half-life (h)

siRNA0 = 1500 # transfection concentration (nM)

transfection_time = 0.08 # transfection time (h)

V_ex = 100 * 10 ** -6 # transfection volume (L)
```

9. Copy and paste the following code, and define the final time point of interest as the variable "tf_transfection".

```
# Time and steps

t0 = 0

tf_transfection = 200.0

dt = 0.01

t = np.linspace(t0,tf_transfection,int(tf_transfection/dt + dt))

n = len(t)
```

10. Copy and paste the following code, and input the internalization rate fitted using the previous procedure.

```
# Internalization rate

k_int = 7.2 * 10 ** -11 # (h^-1)
```

11. Copy and paste the following code, run code, and obtain changes in mRNA and protein concentration levels (Figure 3B). Troubleshooting 5.

```
# Parameters for volume scaling

V_in = 2000 * 10 ** -15 # Intracellular volume (L)

# Unit conversions

mol_M = 6.022 * 10 ** 23 # (number/mole)

rtot = 1.9 * 10 ** 15 # (number/L)

r_tot = rtot / mol_M * (10 ** 9) # (nM)

# Initial concentrations

z0 = [siRNA0,0,0,100,100] # External siRNA, intracellular siRNA, RISC, mRNA, prot

DNA = 100

# Rates

k_sdeg = 0.69315 / siRNA_half_life # (h^-1)

k_sdil = 0.69315 / cell_doubling # (h^-1)

k_mRNA = 0.69315 / mRNA_half_life # (h^-1)

k_mdeg = 0.69315 / mRNA_half_life # (h^-1)

k_risc_orig = 2 * 10 ** -19 # (L/h/number)

k_risc_M = k_risc_orig * mol_M # (M^-1h^-1)

k_risc = k_risc_M * 10 ** -9 # (nM^-1h^-1)

k_rdeg = 0.077 # (h^-1)

k_mRNA = 0.69315 / mRNA_half_life # (h^-1)

k_mdeg = 0.69315 / mRNA_half_life # (h^-1)

k_cleav = 5 * 10 ** 2 # (nM^-1h^-1)

k_prot = 0.69315 / prot_half_life # (h^-1)

k_pdeg = 0.69315 / prot_half_life # (h^-1)
```

```
# Number of cells
rtot = 1.9 * 10 ** 15 # (number/L)
mol_M = 6.022 * 10 ** 23 # (number/mole)
intracellular_vol = 2000 * 10 ** -15 # (L/cell)
r_tot = rtot / mol_M * (10 ** 9) # (nM)
# Unpopulated lists
siRNA_ex = np.ones(n) * z0[0]
siRNA = np.ones(n) * z0[1]
RISC = np.ones(n) * z0[2]
mRNA = np.ones(n) * z0[3]
prot = np.ones(n) * z0[4]
# ODEs
for i in range(1,n):
  siRNA_ex[i] = (-k_int * siRNA_ex[i-1]) * dt + siRNA_ex[i-1]
  if i > (transfection_time / dt):
     siRNA_ex[i] = 0
  siRNA[i] = (k_int * (siRNA_ex[i-1] * (V_ex/V_in)) - k_sdeg * siRNA[i-1] - k_sdil * siRNA[i-
1]) * dt + siRNA[i-1]
  RISC[i] = (k_risc * siRNA[i-1] * (r_tot - RISC[i-1]) - k_rdeg * RISC[i-1])* dt + RISC[i-1]
  if RISC[i] > r_tot:
     RISC[i] = r_tot
  mRNA[i] = (k_mRNA * DNA - k_mdeg * mRNA[i-1] - k_cleav * RISC[i-1] * mRNA[i-1]) * dt + mRNA[i-1]
  if mRNA[i] < 0:
     mRNA[i] = 0
  prot[i] = (k_prot * mRNA[i-1] - k_pdeg * prot[i-1]) * dt + prot[i-1]
  if prot[i] < 0:
     prot[i] = 0
# Normalized values
siRNA_norm = siRNA / z0[0]
mRNA_norm = mRNA / z0[3]
prot_norm = prot / z0[4]
```

12. Copy and paste the following code, and plot desired data accordingly (Figure 3C).

Note: In this example code, experimental data regarding GFP expression levels over time are incorporated into the code. Users can change data within the x-, y-, and err1-arrays for time points, gene expression levels, and associated error, respectively. The '# Experimental data' section of the code can be omitted if users do not wish to plot experimental and modeling data together. Also, plot details including plot type, x-axis and y-axis ticks, and x-axis and y-axis limits can be customized to meet the users' needs. Detailed explanations and examples can be found at matplotlib.org (Hunter, 2007).

```
# Experimental data

x = [0,24,48,72,96,168] # time of measurement (h)

y = [1.00,0.84,0.49,0.43,0.50,0.95] # gene expression levels

err1 = [0.00,0.06,0.04,0.03,0.04,0.00] # standard deviation associated with each
measurement

# Plot figure

plt.errorbar(x,y,yerr=err1,fmt = 'bo')

plt.plot(t,prot_norm,'k')

plt.ylim(0,1.05)

plt.yticks(np.arange(0,1.1,0.2))

plt.xlim(0,192)

plt.xticks(np.arange(0,193,24))

plt.xlabel('Time(h)')

plt.ylabel('Relative GFP expression')

plt.show()
```

Modeling siRNA-induced changes in gene expression levels *in vivo*

Timing: 15 min

⚠ CRITICAL: To proceed with the following steps, make sure to first obtain an optimized internalization rate using the procedure listed in 'fitting for internalization rate.'

A major difference between biological parameters that directly affect intracellular siRNA concentrations *in vitro vs. in vivo* is that the clearance of nucleic acids at the target tissue *in vivo* results in siRNA dilution rates that are significantly different from the dilution rates *in vitro* (Oraiopoulou et al., 2017). More specifically, the dominating siRNA dilution factor *in vitro* is cell division. However, siRNA dilution *in vivo* is governed by additional extracellular delivery barriers. Additional terms were added to the mathematical framework to account for this difference in extracellular siRNA concentration prior to cellular processing. Again, parameters were lumped into two process-related terms – accumulation and clearance from the tissue of interest – to eliminate the need to characterize a series of hard-to-measure transport steps (Blanco et al., 2015). The accumulation of siRNA in the tissue of interest was defined as a constant and approximated from fractional biodistribution values characterized in the literature, (Hoshyar et al., 2016; Jasinski et al., 2018; He et al., 2010; Pérez-Campaña et al., 2013) because transport steps before accumulation into the target tissue, such as clearance from blood, occur rapidly and can be assumed to be instantaneous in comparison to the rates of later steps (Alexis et al., 2008). Particle clearance rates from specific tissues are not as widely characterized as accumulation percentages, and users are required to vary the clearance rate to obtain an accurate kinetic profile for *in vivo* gene silencing.

13. To call the packages that will be used for this protocol, copy and paste the code provided below into Jupyter Notebook (Figure 4A).

Note: Full code is also available in the ipynb file, 'Modeling siRNA-induced changes in gene expression levels in vivo.ipynb,' that is included in the electronic information associated with this publication (Data S3).

Table 2. Model parameters and values

Parameter (name)	Description	Determination	Value (Units)
k_{int}	Internalization rate of external siRNA	Fit to each system	(h^{-1})
V_{ex}	Transfection volume	Specified experimentally	(L)
V_{in}	Intracellular volume	(Wittrup et al., 2015; Bartlett and Davis, 2006)	$2 * 10^{-12}$ (L)
$k_{s,dil}$	Dilution rate of siRNA due to cell division	Calculated from cell doubling time	(h^{-1})
$k_{s,deg}$	Intracellular degradation rate of siRNA	Calculated from siRNA half-life	(h^{-1})
$RISC_{tot}$	Intracellular RISC concentration	(Bartlett and Davis, 2006)	3.2 (nM)
k_{RISC}	Formation rate of active RISC	(Bartlett and Davis, 2006)	$1.2*10^{-4}$ ($nM^{-1}h^{-1}$)
$k_{r,deg}$	Intracellular RISC degradation rate	(Bartlett and Davis, 2006)	$7.7*10^{-2}$ (h^{-1})
k_{mRNA}	Intracellular mRNA production rate	Steady-state approximation, calculated from mRNA half-life	(h^{-1})
$k_{m,deg}$	Intracellular mRNA degradation rate	Calculated from mRNA half-life	(h^{-1})
k_{cleav}	Cleavage rate of mRNA	(Roh et al., 2021)	$5*10^{2}$ ($nM^{-1}h^{-1}$)
k_{prot}	Intracellular protein production rate	Steady-state approximation, calculated from protein half-life	(h^{-1})
$k_{p,deg}$	Intracellular protein degradation rate	Calculated from protein half-life	(h^{-1})

Reprinted with permission from (Roh et al., 2021). Copyright 2021 American Chemical Society.

```
# Import packages
import numpy as np
import matplotlib.pyplot as plt
```

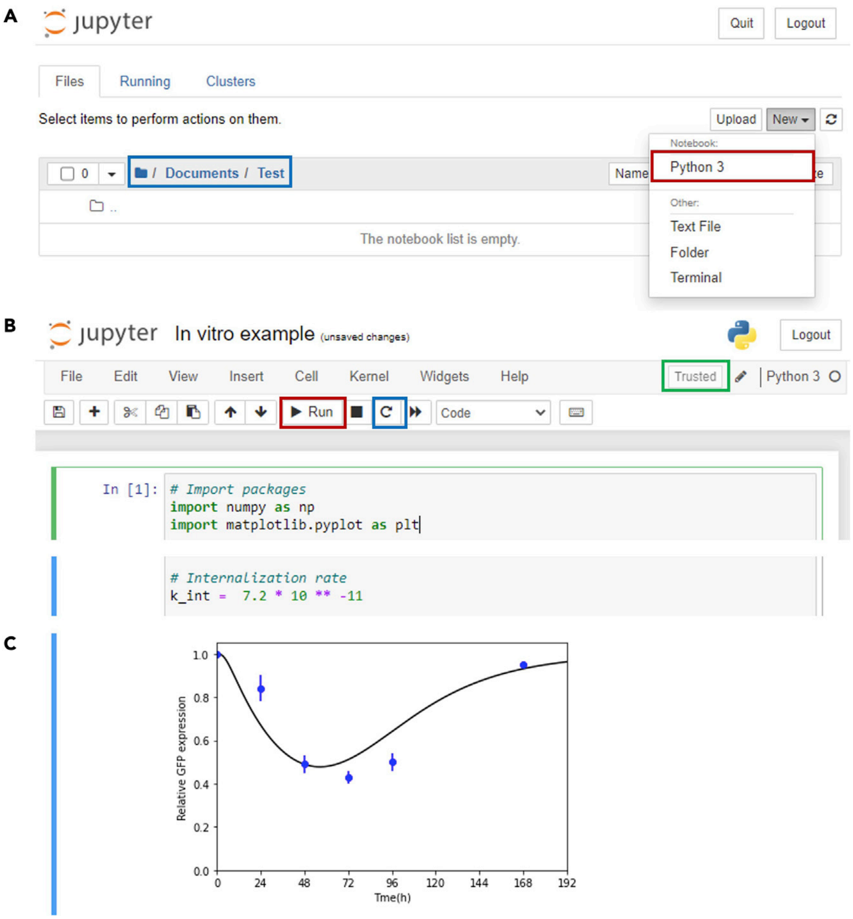
14. Copy and paste the code below.
    a. Modify the values for the eight parameters accordingly.

Note: Users can directly input the seven parameters used in the 'modeling siRNA-induced changes in gene expression levels in vitro' section IF the *in vitro* data set uses the same siRNA formulation to target the same cell type. The example code given here includes different parameter values from the previous section because the *in vivo* example is modeling a different experimental scenario.

    b. Estimate the percentage of particles that arrive at the tissue of interest (defined as the variable "pen") from existing literature.

Note: Accumulation ("pen") is defined as the fractional biodistribution within the tissue of interest, ranging between values of 0 and 1, and is needed to accurately account for the initial external siRNA concentration that is available for cellular internalization. Users are encouraged to estimate this value from literature on the basis of particle size and material.

```
# Inputs
cell_doubling = 27.0 # cell doubling time (h)
siRNA_half_life = 24.0 # siRNA half-life (h)
mRNA_half_life = 16 # mRNA half-life (h)
prot_half_life = 3.3 # protein half-life (h)
siRNA0 = 3760 # transfection concentration (nM)
transfection_time = 530 # transfection time (h)
V_ex = 0.4 * 10 ** -3 # transfection volume (L)
pen = 0.106/100
```



The notebook list is empty.

```
# Import packages
import numpy as np
import matplotlib.pyplot as plt
```

```
# Internalization rate
k_int =  7.2 * 10 ** -11
```

Figure 3. Modeling changes in gene expression levels *in vitro* using the fitted internalization rate

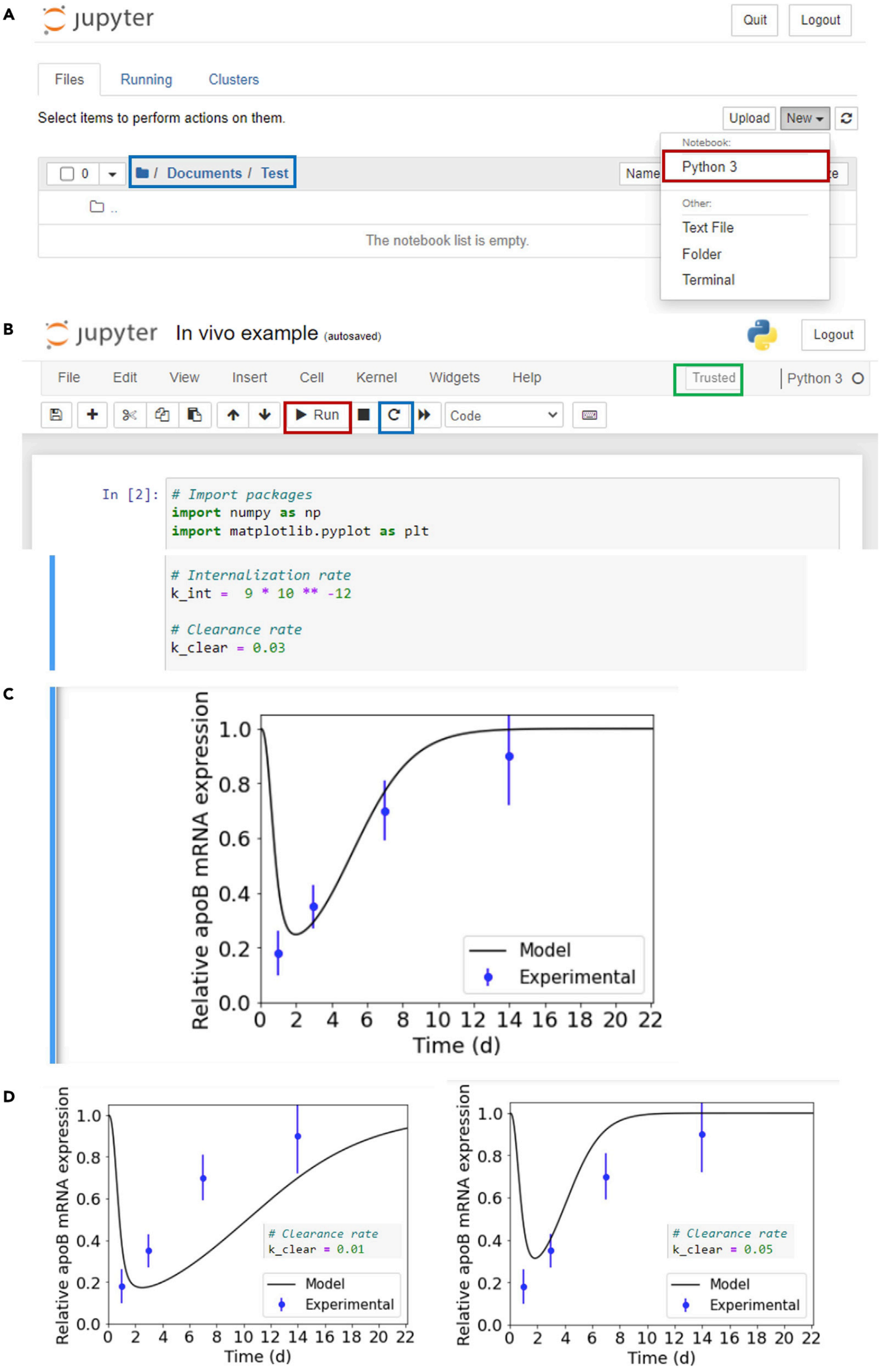
(A) As mentioned in previous section, after opening Jupyter Notebook, choose the correct directory in which to save your notebook (blue box). Click on the 'New' box and create a new Python 3 notebook (red box).

(B) Copy and paste all code provided in the 'modeling siRNA-induced changes in gene expression levels in vitro' section. Make sure that the internalization rate is the same as the optimized rate obtained in the 'fitting for internalization rate' section. If there is an error and users must re-run the code for any reason, make sure to reset the kernel (blue box) before re-running. Make sure that the kernel is trusted (green box).

(C) Once the code is finished, a plot will be generated with the details and data provided in the '# Plot figure' section of the code.

15. Copy and paste the following code, and define the final time point of interest as the variable "tf_transfection".

```
# Time and steps

t0 = 0

tf_transfection = 530.0

dt = 0.01

t = np.linspace(t0,tf_transfection,int(tf_transfection/dt + dt))

n = len(t)
```



```
In [2]: # Import packages
        import numpy as np
        import matplotlib.pyplot as plt
```

```
# Internalization rate
k_int =  9 * 10 ** -12

# Clearance rate
k_clear = 0.03
```

```
# Clearance rate
k_clear = 0.01
```

```
# Clearance rate
k_clear = 0.05
```

Figure 4. Modeling changes in gene expression levels *in vivo* using the fitted internalization rate

(A) As mentioned in previous section, after opening Jupyter Notebook, choose the correct directory in which to save your notebook (blue box). Click on the 'New' box and create a new Python 3 notebook (red box).

(B) Copy and paste all code provided in the 'modeling siRNA-induced changes in gene expression levels in vivo' section. Make sure that the internalization rate is the same as the optimized rate obtained in the 'fitting for internalization rate' section. If there is an error and users must re-run the code for any reason, make sure to reset the kernel (blue box) before re-running. Make sure that the kernel is trusted (green box). Input an initial guess for the clearance rate.

(C) Once the code is finished, a plot will be generated with the details and data provided in the '# Plot figure' section of the code.

(D) Vary the clearance rate until model results and experimental data are in good agreement.

16. Copy and paste the following code, and input the internalization rate fitted using the previous procedure in the 'fitting for internalization rate' section (Figure 4B).

Note: Make sure that the internalization rate is the same as the optimized rate obtained in the 'fitting for internalization rate' section. The example code given here includes a different internalization rate value from the previous section because the *in vivo* example is modeling a different experimental scenario.

```
# Internalization rate

k_int = 9 * 10 ** -12 # (h^-1)
```

17. Copy and paste the following code, and vary the clearance rate (Figure 4B).

Note: The clearance rate is used to correctly depict the amount of external siRNA over time.

⚠ CRITICAL: Exact clearance rates for different tissues are not as widely characterized experimentally, and users may have to vary this rate manually until a good alignment is achieved between experimental and modeling data.

```
# Clearance rate

k_clear = 0.03 # (h^-1)
```

18. Copy and paste the following code, run code, and obtain changes in mRNA and protein concentration levels.

```
# Parameters for volume scaling

V_in = 2000 * 10 ** -15 # Intracellular volume (L)

# Unit conversions

mol_M = 6.022 * 10 ** 23 # (number/mole)

rtot = 1.9 * 10 ** 15 # (number/L)

r_tot = rtot / mol_M * (10 ** 9) # (nM)

# Initial concentrations

z0 = [siRNA0*pen,0,0,100,100] # External siRNA, intracellular siRNA, RISC, mRNA, prot

DNA = 100

# Rates

k_sdeg = 0.69315 / siRNA_half_life # (h^-1)
```

```
k_sdil = 0.69315 / cell_doubling # (h^-1)
k_mRNA = 0.69315 / mRNA_half_life # (h^-1)
k_mdeg = 0.69315 / mRNA_half_life # (h^-1)
k_risc_orig = 2 * 10 ** -19 # (L/h/number)
k_risc_M = k_risc_orig * mol_M # (M^-1h^-1)
k_risc = k_risc_M * 10 ** -9 # (nM^-1h^-1)
k_rdeg = 0.077 # (h^-1)
k_mRNA = 0.69315 / mRNA_half_life # (h^-1)
k_mdeg = 0.69315 / mRNA_half_life # (h^-1)
k_cleav = 5 * 10 ** 2 # (nM^-1h^-1)
k_prot = 0.69315 / prot_half_life # (h^-1)
k_pdeg = 0.69315 / prot_half_life # (h^-1)
# Unpopulated lists
siRNA_ex = np.ones(n) * z0[0]
siRNA = np.ones(n) * z0[1]
RISC = np.ones(n) * z0[2]
mRNA = np.ones(n) * z0[3]
prot = np.ones(n) * z0[4]
# ODEs
for i in range(1,n):
  siRNA_ex[i] = (-k_int * siRNA_ex[i-1] - k_clear * siRNA_ex[i-1]) * dt + siRNA_ex[i-1]
  if i > (transfection_time / dt):
    siRNA_ex[i] = 0
  siRNA[i] = (k_int * (siRNA_ex[i-1] * (V_ex/V_in)) - k_sdeg * siRNA[i-1] - k_sdil * siRNA[i-1]) * dt + siRNA[i-1]
  RISC[i] = (k_risc * siRNA[i-1] * (r_tot - RISC[i-1]) - k_rdeg * RISC[i-1])* dt + RISC[i-1]
  if RISC[i] > r_tot:
    RISC[i] = r_tot
  mRNA[i] = (k_mRNA * DNA - k_mdeg * mRNA[i-1] - k_cleav * RISC[i-1] * mRNA[i-1]) * dt + mRNA[i-1]
  if mRNA[i] < 0:
    mRNA[i] = 0
  prot[i] = (k_prot * mRNA[i-1] - k_pdeg * prot[i-1]) * dt + prot[i-1]
  if prot[i] < 0:
    prot[i] = 0
# Normalized values
siRNA_norm = siRNA / z0[0]
mRNA_norm = mRNA / z0[3]
prot_norm = prot / z0[4]
```

19. Copy and paste the following code and plot desired data accordingly (Figure 4C).

    Note: In this example code, experimental data regarding apoB expression levels over time are incorporated within the code. Users can change data within the x-, y-, and err1-arrays for time points, gene expression levels, and associated error, respectively. The ‘# Experimental data’ section of the code can be omitted if users do not wish to plot experimental and modeling data together. Also, plot details including plot type, x-axis and y-axis ticks, and x-axis and y-axis limits can be customized to meet the users’ needs. Detailed explanations and examples can be found at matplotlib.org (Hunter, 2007).

```
# Experimental data

x = [24,72,168,336]

y = [0.18,0.35,0.7,0.9]

err1 = [0.08,0.08,0.11,0.18]

# Plot figure

plt.figure(1,figsize=(12,4))

fig,ax = plt.subplots(figsize=(7,5.5))

plt.errorbar(x,y,yerr=err1,fmt=’bo’,label=’Experimental’)

plt.plot(t,mRNA_norm,’k’,label=’Model’)

plt.ylim(0,1.05)

plt.yticks(np.arange(0,1.1,0.2),fontsize=18)

plt.xlim(0,530)

plt.xticks(np.arange(0,530,48),fontsize=18)

ax.set_xticklabels([’0’,’2’,’4’,’6’,’8’,’10’,’12’,’14’,’16’,’18’,’20’,’22’])

plt.xlabel(’Time (d)’,fontsize=18)

plt.ylabel(’Relative apoB mRNA expression’,fontsize=18)

plt.legend(loc=4,fontsize=15)

plt.savefig(’Strapps’,dpi=240)

plt.show()
```

20. Vary the clearance rate in step 17, if needed, until model results and experimental data are in good agreement (Figure 4D).

EXPECTED OUTCOMES

Determining a reasonable internalization rate

By running the code provided in ‘fitting for internalization rate,’ users will obtain an internalization rate that results in the lowest RMSE when using the experimental data point(s) defined in step 4. In an earlier publication (Roh et al., 2021) and as shown in Figure 5, we have validated that an internalization rate fitted from an *in vitro* experimental measurement of gene expression at a single time point produced predictions for gene silencing efficiencies that were in remarkable agreement with the predictions using an internalization rate fitted from experimental measurements of gene expression at multiple time points.

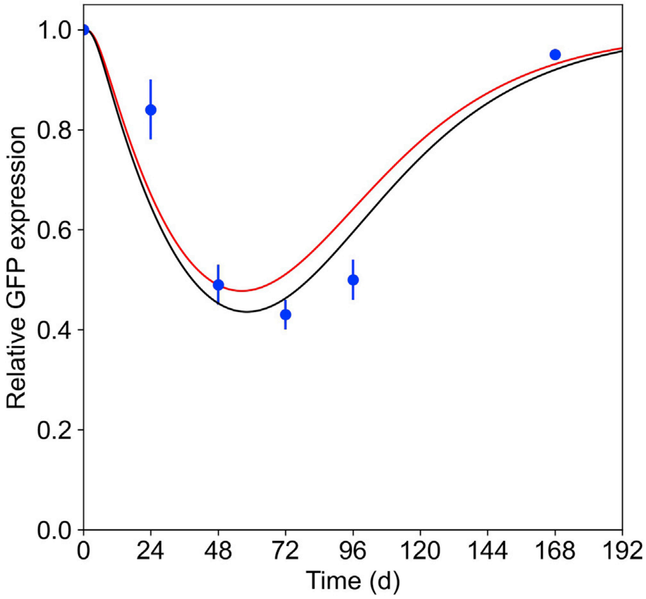


Figure 5. Comparison of model predictions using a single-time-point fit vs. multiple-time-point fit

GFP expression kinetics modulated by naked siRNA *via* electrotransfection in B16F10 cells predicted by the kinetic model using either a single-time-point fit at 48 h (red line) or a multiple-time-point fit (black line) plotted in conjunction with experimental data (blue dots). All data points and error bars for standard deviations were generated using data reported by the original authors (Paganin-Gioanni et al., 2011; Roh et al., 2021).

For this procedure, we recommend using a time point between 48 and 72 h, if possible, as mentioned in the troubleshooting and limitations sections. This time point is appropriate because (1) it is not short enough to run into the limitations associated with our definition of internalization rate as an aggregated parameter of multiple steps; (2) it is long enough to reflect changes in protein concentrations; and (3) it is not too long so that protein expressions start recovering back to normal expression levels. An example of model predictions obtained from a single-time-point fit of time points ranging from 24 to 168 h is shown in Figure 6.

Generating model predictions for siRNA formulation efficacy *in vitro* and *in vivo*

If a reasonable internalization rate was obtained using the above procedure, users will be able to get good agreement between model predictions and experimental data by running the code provided in 'modeling siRNA-induced changes in gene expression levels in vitro.' An example result is shown in Figure 7.

For the *in vivo* modeling, users will be required to vary tissue clearance rates manually because these rates are not widely available in literature. Example results of how model predictions change with different clearance rates are shown in Figure 8.

LIMITATIONS

Due to the assumptions made throughout the model, there is some limitation to the single-time-point fit approach. Specifically, we suspect that there are limitations to obtaining accurate fits for the single unknown parameter, the internalization rate, using earlier time points because we are aggregating multiple steps that are hard to characterize for the internalization process. For reasonable model predictions, users should select a time point that is consistent with the expected kinetics of the comprehensive internalization process. A suggested time point would be 48 h post-siRNA exposure because this time point is longer than most protein half-lives (i.e., a time point that would reflect changes in protein concentrations after siRNA internalization).

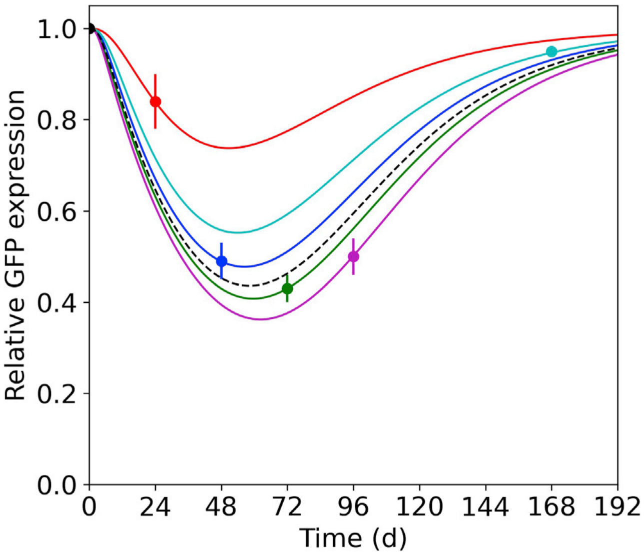


Figure 6. Comparison of model predictions using a single-time-point fit of different time points

GFP expression kinetics modulated by naked siRNA *via* electrotransfection in B16F10 cells predicted by the kinetic model using a single-time-point fit from experimental data collected at 24 h (red), 48 h (blue), 72 h (green), 96 h (magenta), and 168 h (cyan) compared to results using all five experimental data points to obtain an internalization rate (dotted black line). All data points and error bars for standard deviations were generated using data reported by the original authors (Paganin-Gioanni et al., 2011; Roh et al., 2021).

TROUBLESHOOTING

Problem 1

Optimization takes a very long time to complete (during internalization rate fitting, step 6).

Potential solution

A possible reason for this problem is the initial guess was far from the optimal value or an excessive number of experimental time points was specified. Update the internalization rate value in step 5 (recommended starting value is 10^{-10} h^{-1}) or reduce the number of experimental measurement inputs.

Problem 2

Optimization takes a very long time to complete (during internalization rate fitting, step 6) even after implementing the potential solution for problem 1.

Potential solution

This problem may occur when the updated internalization rate is on a significantly different order of magnitude. Change the steps taken during the internalization rate so that the order of magnitude is on the same scale as the updated internalization rate. For example, if the updated internalization rate in step 5 is 10^{-9}, change the steps in the linked section of the code to $0.1*10^{-9}$.

Problem 3

Optimization takes a very long time to complete (during internalization rate fitting, step 6) even after implementing the potential solutions for problems 1 and 2.

Potential solution

This problem may persist when there is a large number of time steps in step 3. The accuracy of model predictions is enhanced as more time steps are taken, but this increase results in an escalation in time needed to compute the ordinary differential equations. Adjust the value of the final time point of interest, tf_transfection, to be similar to the experimental time point that is used for optimizing the internalization

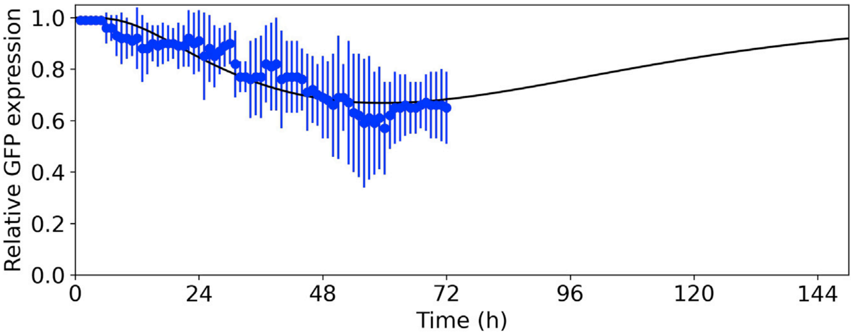


Figure 7. Comparison of model predictions and experimental results of *in vitro* GFP silencing
Model prediction (black) compared to experimental values (blue) for changes in *in vitro* GFP expression levels within HeLa cells using calcium phosphate nanoparticles. All data points and error bars for standard deviations were generated using data reported by the original authors (Chernousova and Epple, 2017; Roh et al., 2021).

rate. For example, if the experimental time point that is used for optimization is 48 h, a value of 50 is appropriate for tf_transfection. The timestep value, dt, can also be reduced, but a minimum value of 0.01 is recommended based on characteristic time calculations of the fastest rate (i.e., k_{cleav}).

Problem 4

The code reports an optimized internalization rate of 0 after step 6.

Potential solution

This problem occurs when the initial guess for the internalization rate specified in step 5 is too large. Reduce the k_int_init value in step 5 by an order of magnitude. Repeat this process until the code no longer reports an optimized internalization rate of 0.

Problem 5

Model results do not align with experimental data using optimized internalization rate based on fitting data from a single time point (during *in vitro* modeling, steps 7–11).

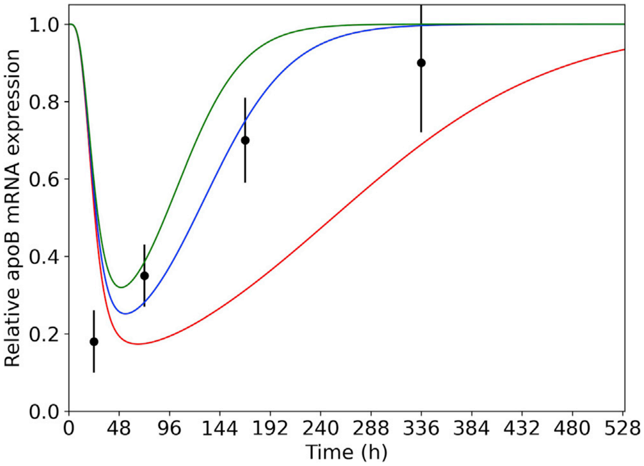


Figure 8. Comparison of model predictions using different clearance rates and experimental results of *in vivo* apoB silencing
In vivo model predictions (green line, clearance rate = 0.05 h^{-1}; blue line, clearance rate = 0.03 h^{-1}; red line, clearance rate = 0.01 h^{-1}) compared to *in vivo* experimental values (black) of mRNA expression levels in response to a single siRNA dose over time when delivered in the form of a siRNA-polymer conjugate. All data points and error bars for standard deviations were generated using data reported by the original authors (Strapps et al., 2010; Roh et al., 2021).

Potential solution

As discussed in the 'limitations' section, using experimental data taken at too early of a time point can result in inaccurate fits for the internalization rate. Choose a time point between 48 and 72 h.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Thomas H. Epps, III (thepps@udel.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The published article includes all datasets/code generated or analyzed during this study. Full code is also available in ipynb files that are included in the electronic information associated with this publication.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.xpro.2022.101723.

ACKNOWLEDGMENTS

This work was funded in part by the National Institutes of Health (T32 GM133395 and R21 EB028108-02) and in part by the National Science Foundation (NSF BMAT 1507540). The contents of the manuscript do not necessarily reflect the views of the funding agencies.

AUTHOR CONTRIBUTIONS

Conceptualization, E.H.R., M.O.S., and T.H.E.; Methodology, E.H.R., M.O.S., and T.H.E.; Investigation, E.H.R.; Resources, M.O.S. and T.H.E.; Writing—Original Draft, E.H.R., M.O.S., and T.H.E.; Writing—Review & Editing, E.H.R., M.O.S., and T.H.E.; Visualization, E.H.R.; Supervision, M.O.S. and T.H.E.; Funding Acquisition, M.O.S. and T.H.E.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Alexis, F., Pridgen, E., Molnar, L.K., and Farokhzad, O.C. (2008). Factors affecting the clearance and biodistribution of polymeric nanoparticles. Mol. Pharm. *5*, 505–515. https://doi.org/10.1021/mp800051m.

Bartlett, D.W., and Davis, M.E. (2006). Insights into the kinetics of siRNA-mediated gene silencing from live-cell and live-animal bioluminescent imaging. Nucleic Acids Res. *34*, 322–333. https://doi.org/10.1093/nar/gkj439.

Blanco, E., Shen, H., and Ferrari, M. (2015). Principles of nanoparticle design for overcoming biological barriers to drug delivery. Nat. Biotechnol. *33*, 941–951. https://doi.org/10.1038/nbt.3330.

Chernousova, S., and Epple, M. (2017). Live-cell imaging to compare the transfection and gene silencing efficiency of calcium phosphate nanoparticles and a liposomal transfection agent. Gene Ther. *24*, 282–289. https://doi.org/10.1038/gt.2017.13.

Cuccato, G., Polynikis, A., Siciliano, V., Graziano, M., Di Bernardo, M., and Di Bernardo, D. (2011). Modeling RNA interference in mammalian cells. BMC Syst. Biol. *5*, 19. https://doi.org/10.1186/1752-0509-5-19.

Haley, B., and Zamore, P.D. (2004). Kinetic analysis of the RNAi enzyme complex. Nat. Struct. Mol. Biol. *11*, 599–606. https://doi.org/10.1038/nsmb780.

Harris, C.R., Millman, K.J., Van Der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., et al. (2020). Array programming with NumPy. Nature *585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2.

He, C., Hu, Y., Yin, L., Tang, C., and Yin, C. (2010). Effects of particle size and surface charge on cellular uptake and biodistribution of polymeric nanoparticles. Biomaterials *31*, 3657–3666. https://doi.org/10.1016/j.biomaterials.2010.01.065.

Hoshyar, N., Gray, S., Han, H., and Bao, G. (2016). The effect of nanoparticle size on *in vivo* pharmacokinetics and cellular interaction. Nanomedicine *11*, 673–692. https://doi.org/10.2217/nnm.16.5.

Hunter, J.D. (2007). Matplotlib: a 2D graphics environment. Comput. Sci. Eng. *9*, 90–95. https://doi.org/10.1109/MCSE.2007.55.

Jasinski, D.L., Li, H., and Guo, P. (2018). The effect of size and shape of RNA nanoparticles on biodistribution. Mol. Ther. *26*, 784–792. https://doi.org/10.1016/j.ymthe.2017.12.018.

Oraiopoulou, M.E., Tzamali, E., Tzedakis, G., Vakis, A., Papamatheakis, J., and Sakkalis, V. (2017). *In vitro/in silico* study on the role of doubling time heterogeneity among primary glioblastoma cell lines. BioMed Res. Int. *2017*, 8569328. https://doi.org/10.1155/2017/8569328.

Paganin-Gioanni, A., Bellard, E., Escoffre, J.M., Rols, M.P., Teissié, J., and Golzio, M. (2011). Direct visualization at the single-cell level of siRNA electrotransfer into cancer cells. Proc. Natl. Acad. Sci. USA *108*, 10443–10447. https://doi.org/10.1073/pnas.1103519108.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: machine learning in Python. J. Mach. Learn. Res. *12*, 2825–2830.

Pérez-Campaña, C., Gómez-Vallejo, V., Puigivila, M., Martín, A., Calvo-Fernández, T., Moya, S.E., Ziolo, R.F., Reese, T., and Llop, J. (2013). Biodistribution of different sized nanoparticles assessed by positron emission tomography: a general strategy for direct activation of metal oxide particles. ACS Nano *7*, 3498–3505. https://doi.org/10.1021/nn400450p.

Pratt, A.J., and MacRae, I.J. (2009). The RNA-induced silencing complex: a versatile gene-silencing machine. J. Biol. Chem. *284*, 17897–17901. https://doi.org/10.1074/jbc.R900012200.

Roh, E.H., Epps, T.H., III, and Sullivan, M.O. (2021). Kinetic modeling to accelerate the development of nucleic acid formulations. ACS Nano *15*, 16055–16066. https://doi.org/10.1021/acsnano.1c04555.

Sakurai, K., Chomchan, P., and Rossi, J.J. (2010). Silencing of gene expression in cultured cells using small interfering RNAs. Curr. Protoc. Cell Biol. *Chapter 27*. Unit 27.1.1-28. https://doi.org/10.1002/0471143030.cb2701s47.

Strapps, W.R., Pickering, V., Muiru, G.T., Rice, J., Orsborn, S., Polisky, B.A., Sachs, A., and Bartz, S.R. (2010). The siRNA sequence and guide strand overhangs are determinants of in vivo duration of silencing. Nucleic Acids Res. *38*, 4788–4797. https://doi.org/10.1093/nar/gkq206.

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nat. Methods *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2.

Wittrup, A., Ai, A., Liu, X., Hamar, P., Trifonova, R., Charisse, K., Manoharan, M., Kirchhausen, T., and Lieberman, J. (2015). Visualizing lipid-formulated siRNA release from endosomes and target gene knockdown. Nat. Biotechnol. *33*, 870–876. https://doi.org/10.1038/nbt.3298.