Reproducibility study of the modular and reusable model of epithelial transport in the proximal convoluted tubule

Leyla Noroozbabaee[1*], Pablo J. Blanco[2], Soroush Safaei[1], and David P. Nickerson[1]

[1]Auckland Bioengineering Institute, University of Auckland, New Zealand
[2]National Laboratory for Scientific Computing, Petrópolis, Brazil

ORIGINAL

Abstract

We describe here our implementation of a renal epithelial model as published in Noroozbabaee et al. (2022). The flexible and modular model we presented in Noroozbabaee et al. (2022) can be adapted to specific configurations of epithelial transport. The model describes the cellular and sub-cellular mechanisms of the transporters, intracellular buffering, solute fluxes, and other processes. We provide free and open access to the Python implementation to ensure our multiscale proximal tubule model is accessible, enabling the reader to explore the model through configuring their own simulations, executing reproducibility tests and sensitivity analyses, and reusing the model in new work. Here we present the reproduction of a selection of results from Noroozbabaee et al. (2022), providing readers with brief instructions on using the Python implementation to produce these results from the primary article.

Keywords: Proximal Tubule, Epithelial transport, Computational Model, Epithelial cell model, Python

1 Introduction

In the primary article (Noroozbabaee et al., 2022), we introduced a consistent, reproducible, and reusable proximal convoluted tubule (PCT) epithelial transport model. The comprehensive PCT epithelial model consists of cellular and lateral intercellular compartments located between luminal and peritubular solutions. The model includes fifteen solutes in each compartment and two impermeant species. The current work involves the mathematical curation of the model implementation in Python. A persistent workspace for this work is available under an open-source and permissive license at https://github.com/iNephron/W-PCT-E/releases/tag/v1.0.2. Here we demonstrate a quick instruction to reproduce figures related to the structural analysis section.

2 Model description

The comprehensive W-PCT-E model consists of cellular and lateral intercellular compartments between luminal and peritubular solutions. Figure 1 shows a schematic representation of PCT
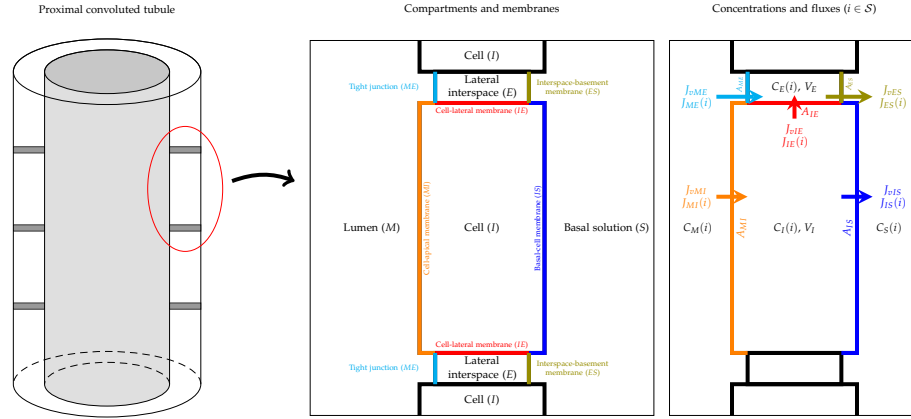
Figure 1. Schematic representation of the proximal convoluted tubule (PCT) epithelium, consisting of cell and lateral intercellular space, and a tubule model, in the way lumen is lined by epithelium (Noroozbabaee et al., 2022, Figure 2).

epithelium and features both configurations, in which cellular and lateral intercellular (LIS) compartments line the tubule lumen. Within each compartment, the concentration of species (i) is designated $C_\alpha(i)$, where α is lumen (M), lateral interspace (E), cell (I) or basal solution (S). Separating membranes are combinations of letters such as luminal cell membrane (lumen-cell membrane, MI), tight junction (ME), cell-lateral membrane (IE), interspace basement membrane (ES), or cell-basal membrane (IS). The order of the two letters indicates the positive direction of the mass flow. $J_{\alpha\beta}$ and $J_{v\alpha\beta}$ represent the solute flux and water flux, respectively, through the corresponding membrane; A is the related membrane surface area; V is the volume; E is the trans-epithelial potential difference. There are 14 transporters (symporters, antiporters, complex transporters, and ATPases) that produce electrochemical fluxes in the current model.

2.1 Modular Description

The current Python code divides the mathematical W-PCT-E model into distinct components, which can be reused, eliminated or added to the model according to the desired experiments. Here we focus on the solute transport aspects of the model as listed in Table 1. The reader is referred to the primary article for the complete description of all modules in the W-PCT-E model (Noroozbabaee et al., 2022).

Each of these modules in Table 1 is a Python model, enabling the researchers to reuse these components in their studies.

In this study, the designed experiments aim to investigate the model behaviour corresponding to either biological or, in some cases, extreme assumptions concerning the model configuration. Considering these assumptions, we can test the models' limitations, flexibility, or reliability.

3 Model status

The current W-PCT-E model implementation in Python was executed using Python 3.9.0 [MSC v.1927 64 bit (AMD64)] on win32. The W-PCT-E model has been validated through the following experiments:

- The W-PCT-E model reproducibility was tested by replicating some simulation experiments reported in Weinstein (1992) and Weinstein et al. (2007).

- The W-PCT-E model flexibility and reusability were tested by reproducing a simple model of Na+ transport in the mammalian urinary bladder to study the salt sensitivity, see Latta et al. (1984).

- Structural analysis of the W-PCT-E model validated the application of the model to the wide range of physiological perturbations.

Table 1. Python Transport Modules (W_PCT_E_Sim_Fig.py. Here, all the equation numbers refer to the equation numbers in the primary paper.)

| Compartment	Membrane	Stoichiometry	Equation	Python Module	On-Off Parameter
		Convective Transport			
Convective Flux	peritubular & Apical	—	Eq:(37)	def csf()	param_csf
		Simple Diffusion			
Passive Flux	peritubular & Apical	—	Eq(39-41)	def goldman()	param_goldman
		Simple Transport (Symporter)			
K^+-Cl^-	peritubular	[1:1]	Eq:(42)	def k-cl()	param_kcl
Na^+-glucose	apical	[1:1]	Eq:(43)	def sglt_mi()	param_sglt_mi
Na^+-$H_2PO_4^-$	apical	[1:1]	Eq:(44)	def nah2po4_mi()	param_nah2po4_mi
		Simple Transport (Antiporter)			
Na^+/H^+	apical	[1:-1]	Eq:(45)	def nah()	param_nah_mi
Na^+/NH_4^+	apical	[1:-1]	Eq:(46)	def nanh4()	param_nanh4_mi
Cl^-/HCO_2^-	apical	[1:-1]	Eq:(47)	def clhco2_mi()	param_clhco2_mi
Cl^-/HCO_3^-	apical	[1:-1]	Eq:(48)	def clhco3_mi()	param_clhco3_mi
NHE3	apical	Weinstein (1995)	Weinstein (1995)	def nhe3()	param_nhe3
		Complex Transport			
Na^+-3 HCO_3^-	peritubular	[1:3]	Eq:(49)	def na_hco3()	param_na_hco3
Na^+-2 HCO_3^-/Cl^-	peritubular	[1:-1:2]	Eq:(50)	def na_cl_hco3()	param_na_cl_hco3
		Active Transport			
H^+	apical	–	Eq:(53)	def h_atp_mi()	param_mi_h
Na^+/K^+-ATPS	peritubular	–	Eqs:(54-58)	def nak_atp()	param_nak_atp
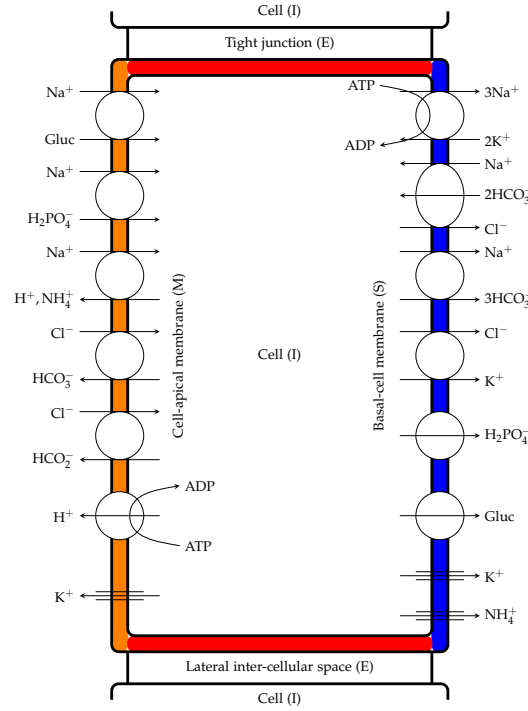
Figure 2. Proximal tubule cells showing coupled transport pathways and some ion channels within the luminal and peritubular cell membranes, (Noroozbabaee et al., 2022, Figure 3).

4 Model Computational Simulation

For the W-PCT-E simulations, the 35 nonlinear ordinary differential equations are solved using a finite difference numerical method for time discretisation along the Python solver "scipy. optimise.root". Evaluation of the model involves integrating the mass conservation equations from an initial time [t0 = 0 s] to a final time [tf = 2000 s] using small time increments [dt = 0.1 s]. The simulation time is chosen to ensure that a steady state is reached. To ensure your Python environment has the required Python modules used in these scripts, one needs to install all the Python requirements by performing "pip install -r requirements.txt" or as appropriate for your platform and Python environment.

5 Model results

Here, we performed structural analysis on the W-PCT-E model to demonstrate its flexibility and explore its application to a range of physiological perturbations. To investigate the effect of each transporter in the W-PCT-E model on the overall behaviour, we performed experiments in which we individually inhibited each of the transporters and compared the total epithelial fluxes.

Structural analyses were performed by inhibiting key transporters in different membranes, such as the Na+/K + −ATPase in the peritubular membrane or SGLT, NHE3, and Na + −H2PO4 transporters in the apical membrane. The reproduction of structural analyses results of Noroozbabaee et al. (2022) is given in Figures 3–4, with the following sections providing instruction on how to use the associated Python codes to obtain these results.

To limit our focus on the structural analysis results and investigate the effect of different transporters in the W-PCT-E model [W_PCT_E_Sim_Fig.py], we define the model parameter "Figure_6_7_8 = 1". It is important to mention that to stop the reproduction of the rest of the model simulation results; we need to make the following adjustment in the W-PCT-E model parameters: Figure_4a = Figure_4b = Figure_4c = Figure_5 = Figure_9_10 = 0. The version of the implementation associated with this manuscript has these settings, so no changes are needed. The following description defines the various parameters used throughout the [W_PCT_E_Sim_Fig.py] script to configure

and execute the simulations required to produce the data presented here. No changes are required in the implementation. The resultant figures can be produced using the [Figures_W_PCT_E.py] script by setting the desired figure parameter to 1, with the figure numbering in the script aligned with the primary article (i.e., figures 6A, 6B, 7, 8A, and 8B).

5.1 Inhibition of Peritubular (IS and IE) Transporters

We separately eliminated the Na^+/K^+-ATPase and two symporters (K^+-Cl^- and Na^+-HCO_3^-) on both the cell-basal and cell-lateral membranes and observed the resulting membrane fluxes and cellular concentrations. Inhibition of each transporter was accomplished by setting the coupling transport coefficient to zero. Figure 3 presents the inhibition of basal and cell-lateral transporters – this reproduces the results presented in Figure 6 of Noroozbabaee et al. (2022). On the first row in each panel, we present four sets of results for four different configurations that depict the total membrane fluxes for the following species: Na+, K+, Cl−, and Glucose. The total membrane fluxes include all the membrane activities from five membranes. Panel (a) represents the original full model (control configuration). Panel (b) illustrates the scenario due to the Na+-K+ pump elimination. Panel (c) corresponds to the K+- Cl − elimination scenario. Panel (d) is for the inhibition of Na+-HCO3 − transporters. In the second row, we illustrate the cellular concentrations corresponding to the related configuration for the same species: Na+, K+, Cl − , Glucose.

To obtain panel (a), we applied the default parameters [control configuration]. We ran the W-PCT-E model simulation [W_PCT_E_Sim_Fig.py] for the total time simulation of 2000 s and saved the related results in the python files to be used later by the Python file "Figures_W_PCT_E.py".

To obtain panel (b), we eliminate the Na+/K + −ATPase by defining the "param_sodium_pumps = 0", while the rest of the system's parameters are the same as the default values. To reproduce panel (c), we eliminate the symporter K^+-Cl^- by setting "param_kcl = 0"; while we reset "param_sodium_pumps = 1". Panel (d) illustrates the response of the W-PCT-E model in the case of elimination of Na^+-HCO_3^- transporter by setting "param_nahco3 =0"; while we reset "param_sodium_pumps = 1" and "param_kcl = 1".

When we change the model configuration, we must ensure that the system's initial conditions correspond to physiological values. For this reason, we ran the model simulation for the first 500 s based on the control configuration. Then we applied the new configuration and continued the model simulation. We saved the results related to the fluxes in the python file: "Data_Figure_6A.py" and those related to solute concentrations in "Data_Figure_6B.py".

5.2 Inhibition of Apical Membrane (MI) Transporters

We separately eliminate the NHE3 antiporter and apical symporters (SGLT and Na^+-$H_2PO_4^-$). Then we study the behaviour of the W-PCT-E model by analysing the results for membrane fluxes and cellular concentrations relative to each scenario. Inhibition of each transporter was accomplished by setting the coupling transport coefficient to zero. Figure 4 presents the inhibition of apical membrane transporters – this reproduces the results presented in Figure 8 of Noroozbabaee et al. (2022).

On the first row in each panel, we present four sets of results for four different configurations that depict the total membrane fluxes for the indicated species. Panel (a) represents the original full model (control configuration). Panel (b) illustrates the scenario due to the NHE3 elimination. Panel (c) corresponds to the SGLT elimination scenario. Panel (d) is for the inhibition of Na^+-$H_2PO_4^-$ – transporters. In the second row, we illustrate the cellular concentrations corresponding to the related configuration for the same species.

To obtain panel (a), we applied the default parameters [control configuration]. We ran the W-PCT-E model simulation [W_PCT_E_Sim_Fig.py] for the total time simulation of 2000 s.

To obtain panel (b), we eliminate the NHE3 by defining the "param_nhe3 = 0" while the rest of the system's parameters are the same as the default values. To reproduce panel (c), we eliminate the apical symporter SGLT by setting "param_sglt_mi = 0"; while we reset "param_nhe3 = 1". Panel

(d) illustrates the response of the W-PCT-E model in the case of elimination of Na^+-$H_2PO_4^-$ transporter by setting "param_nah2po4_mi = 0"; while we reset "param_nhe3 = 1" and "param_sglt = 1".

We ran the model simulation for the first 500 s based on the control configuration. Then we applied the new configuration and continued the model simulation. We saved the results related to all the new configurations for the different fluxes in the python file: "Data_Figure_8A.py" and related results for the solute concentrations are saved in "Data_Figure_8B.py" to be used later by the Python file "Figures_W_PCT_E.py".

5.3 Total Epithelial Sodium Fluxes

The first row in (Noroozbabaee et al., 2022, Figure 7) represents the sodium fluxes for the full W-PCT-E model, considered the control version. In the second row, we illustrate the sodium fluxes due to the elimination of the Na+/K + −ATPase. And the third row corresponds to the elimination of NHE3. Figure 5 reproduces the results presented in Figure 7 of Noroozbabaee et al. (2022).

Panel (a) illustrates the different membrane fluxes. Panel (b) presents the epithelial sodium fluxes classified into convective, passive and electrochemical types. Panel (c) details the electrodiffusive activities into their segments: NHE3, SGLT, and Na^+-HPO_4.

To reproduce Figure 5, we used the related results already produced by the previous simulations in subsection 5.1 and subsection 5.2. The results were collected in the python folder "Data_Figure_7.py".

6 Discussion

We have presented in Noroozbabaee et al. (2022) what we believe to be a comprehensive and FAIR epithelial model for the PCT of the renal nephron. Here, we aim to highlight the flexibility of the W-PCT-E model, as the user can have a better insight into the system behaviour due to the elimination or cooperation of transporters by simply turning them on or off.

Because of the flexibility of the W-PCT-E model, there is a future opportunity for similar analyses to describe the system behaviour due to the elimination of other transporters and their impact on the different solutes.

The simulation experiments performed here aim to investigate the model behaviour corresponding to either biological or, in some cases, extreme assumptions concerning the model configuration. By considering these assumptions, we can test the models' limitations, flexibility, or reliability. In some cases, this pushed the boundaries of the biological system into the non-biological. These significant changes in configuration could, for example, generate an unrealistic stimulus in the model, which results in the model producing negative concentrations. To avoid the numerical error due to Eq. 52 in our primary paper (concerning its logarithmic nature), we added the if-else statements in the model implementation in the W_PCT_E_Sim_Fig.py file as shown below:

```
def f_eps(c, z, v):
    if c > 0 and c != 0:
        return rte * np.log(c) + z * f * v * 1.e-6
    else:
        print('uy')
        return rte * np.log(abs(c)) + z * f * v * 1.e-6
```
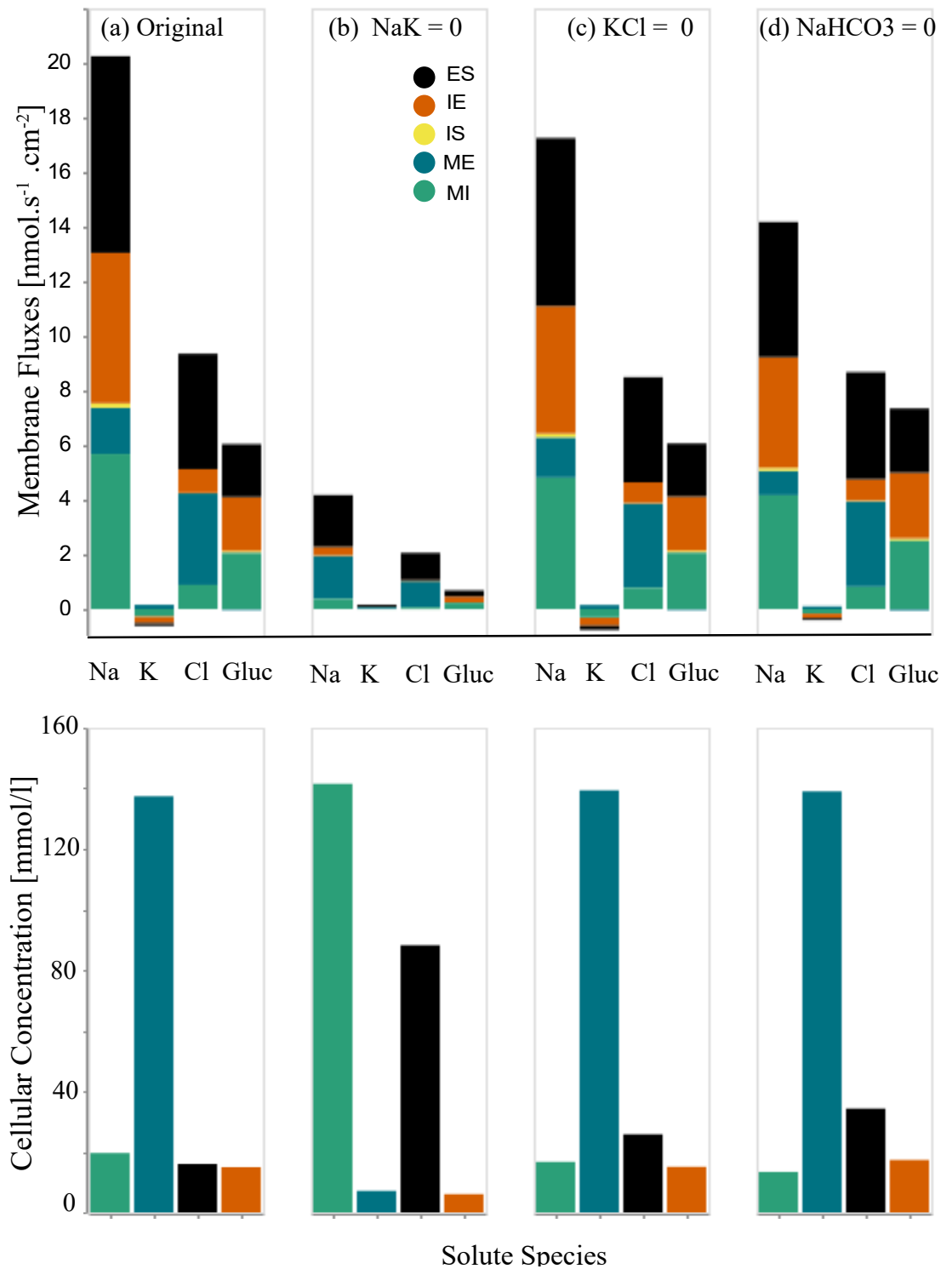
Figure 3. Changes in the membrane fluxes (first row) and cellular concentrations (second row) due to the inhibition of transporters on the cell-basal and cell-lateral membranes, (Noroozbabaee et al., 2022, Figure 6).
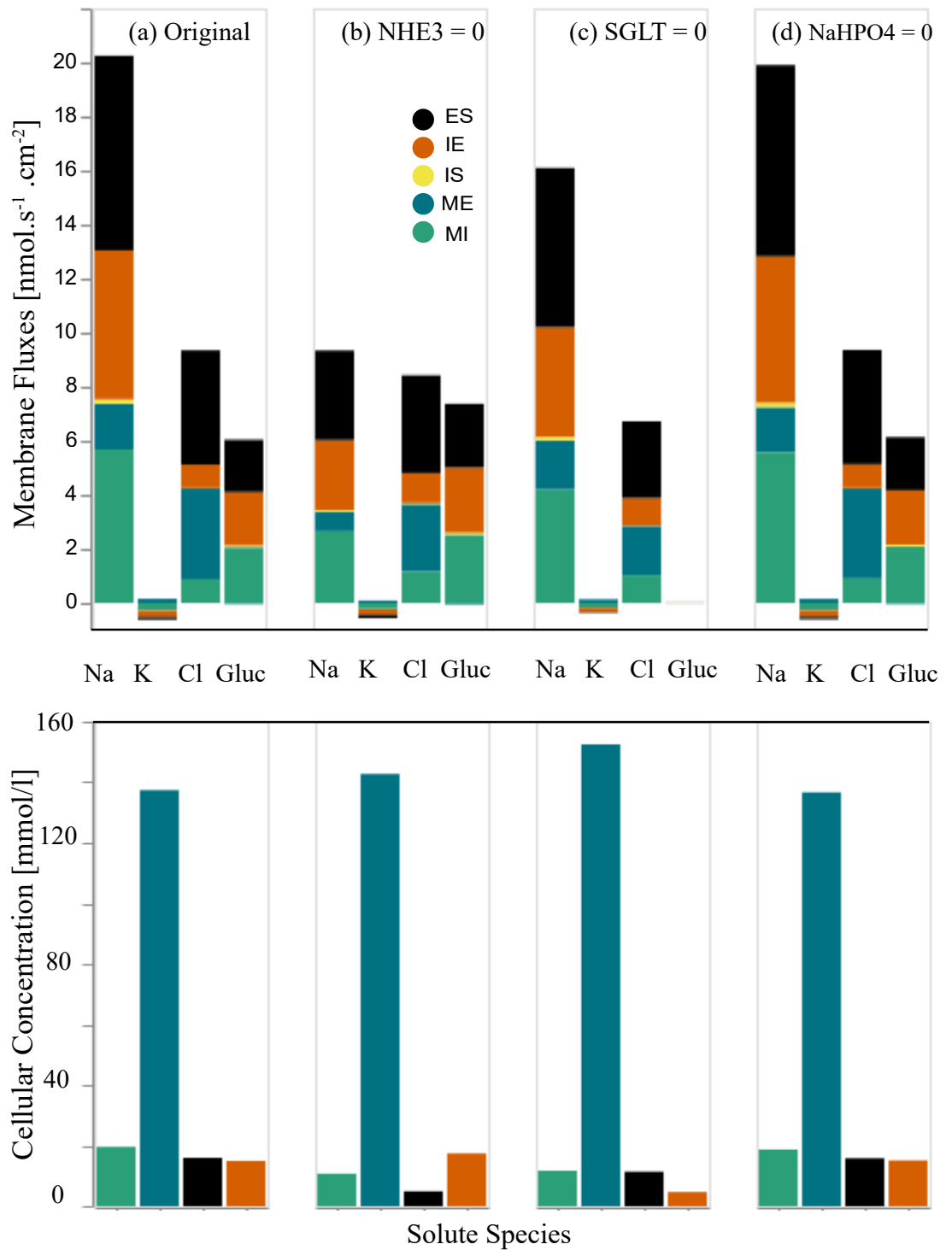
Figure 4. Changes in the membrane fluxes (first row) and cellular concentrations (second row) due to the inhibition of transporters on the apical cell membrane, (Noroozbabaee et al., 2022, Figure 8).
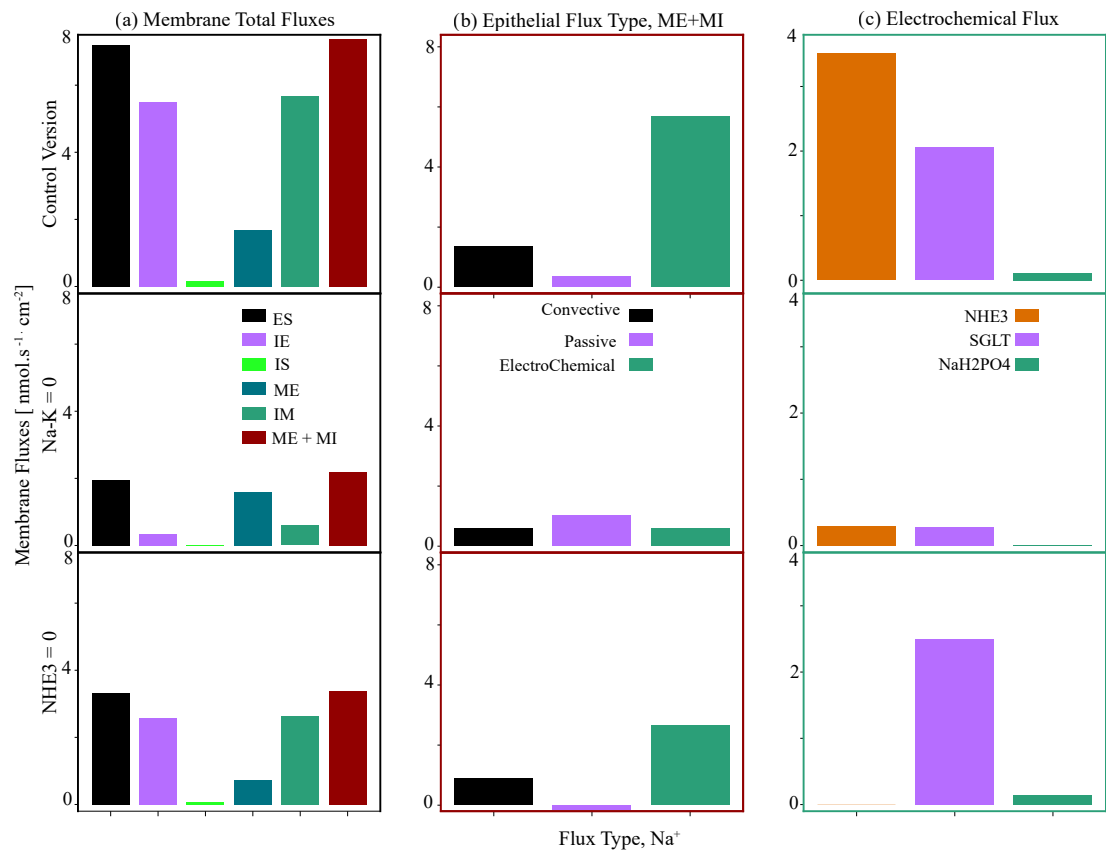
Figure 5. Total epithelial sodium fluxes and the contribution of various sodium flux types, (Noroozbabaee et al., 2022, Figure 7).

References

R. Latta, C. Clausen, and L. C. Moore. General method for the derivation and numerical solution of epithelial transport models. *J Membrane Biol*, 82(1):67–82, Feb. 1984. ISSN 0022-2631, 1432-1424. doi: 10.1007/BF01870733. URL http://link.springer.com/article/10.1007/BF01870733.

L. Noroozbabaee, P. J. Blanco, S. Safaei, and D. P. Nickerson. A modular and reusable model of epithelial transport in the proximal convoluted tubule. *PLoS ONE*, 17(11):e0275837, 2022. doi: 10.1371/journal.pone.0275837.

A. M. Weinstein. Chloride transport in a mathematical model of the rat proximal tubule. *American Journal of Physiology-Renal Physiology*, 263(5):F784–F798, 1992.

A. M. Weinstein. A kinetically defined na$^+$/h$^+$ antiporter within a mathematical model of the rat proximal tubule. *The Journal of General Physiology*, 105(5):617–641, May 1995. ISSN 0022-1295, 1540-7748. doi: 10.1085/jgp.105.5.617. URL https://rupress.org/jgp/article/105/5/617/26432/A-kinetically-defined-NaH-antiporter-within-a.

A. M. Weinstein, S. Weinbaum, Y. Duan, Z. Du, Q. Yan, and T. Wang. Flow-dependent transport in a mathematical model of rat proximal tubule. *American Journal of Physiology-Renal Physiology*, 292(4):F1164–F1181, 2007.

Reproducibility report for: Reproducibility study of the modular and reusable model of epithelial transport in the proximal convoluted tubule.
Submitted to: *Physiome*
Manuscript identifier: S000024

Curation outcome summary: We successfully reproduced Figure 3, Figure 4 and Figure 5 as shown in the manuscript.

Box 1: Criteria for repeatability and reproducibility

☑ **Model source code provided:**

    ☑ Source code: a standard procedural language is used (e.g. MATLAB, Python, C)

        ☐ There are details/documentation on how the source code was compiled
        ☑ There are details on how to run the code in the provided documentation
        ☑ The initial conditions are provided for each of the simulations
        ☑ Details for creating reported graphical results from the simulation results

    ☐ Source code: a declarative language is used (e.g. SBML, CellML, NeuroML)

        ☐ The algorithms used are defined or cited in previous articles
        ☐ The algorithm parameters are defined
        ☐ Post-processing of the results are described in sufficient detail

☐ **Executable model provided:**

    ☐ The model is executable without source (e.g. desktop application, compiled code, online service)

        ☐ There are sufficient details to repeat the required simulation experiments

☑ **The model is described mathematically in the article(s):**

    ☑ Equations representing the biological system

    ☑ There are tables or lists of parameter values

    ☑ There are tables or lists of initial conditions

    ☑ Machine-readable tables of parameter values

    ☑ Machine-readable tables of initial conditions

☑ **The simulation experiments using the model are described mathematically in the article:**

    ☑ Integration algorithms used are defined

    ☐ Stochastic algorithms used are defined

    ☐ Random number generator algorithms used are defined

    ☐ Parameter fitting algorithms are defined

    ☑ The paper indicates how the algorithms yield the desired output

Box 2: Criteria for accessibility

■ Model/source code is available at a public repository or researcher's web site

☐ Prohibitive license provided

☐ Permissive license provided

■ Open-source license provided

■ All initial conditions and parameters are provided

■ All simulation experiments are fully defined (events listed, collection times and measurements specified, algorithms provided, simulator specified, etc.)

Box 3: Rules for Credible practice of Modeling and Simulation[a]

[a]Model credibility is assessed using the Interagency Modeling and Ananlysis Group conformance rubric:
https://www.imagwiki.nibib.nih.gov/content/10-simple-rules-conformance-rubric

■ Define context clearly: Extensive

■ Use appropriate data: Extensive

■ Evaluate within context: Extensive

■ List limitations explicitly: Adequate

■ Use version control: Adequate

■ Document adequately: Adequate

■ Conform to standards: Extensive

Box 4: Evaluation

■ Model and its simulations could be repeated using provided declarative or procedural code

■ Model and its simulations could be reproduced

CENTER FOR REPRODUCIBLE
BIOMEDICAL MODELING

Director: Professor Herbert M. Sauro
University of Washington, Seattle, WA
https://reproduciblebiomodels.org

Summary comments: Model and source code are available in the associated OMEX archive. This was used in our attempt to reproduce the results presented in the paper. We successfully installed the required dependencies and ran the python files provided to reproduce Figure 3, Figure 4 and Figure 5 as shown in the manuscript.

Anand K. Rampadarath[1], PhD
Curator at Center for Reproducible Biomedical Modeling

---

[1]Contact: info@reproduciblebiomodels.org