

Sirael - Virtual Metabolic Machine

Tomas Koutny

Contributing authors: tomas@sirael.science;

Abstract

A metabolic simulator is essential to reduce adverse effects of clinical studies. Moreover, it reduces monetary costs when developing new drugs and control-algorithms for new devices. Diabetes is one of the diseases, where the metabolic simulation has established. Such a simulator mimics real metabolism to provide simulated responses to external events such as meal ingestion, physical activity, drugs and pump-delivered insulin. In this paper, we propose a novel approach, which addresses limits of the current metabolic simulators. First, a simulator is usually based on a set of differential equations, which are de-facto black-boxes for physicians. The proposed approach, Sirael, provides a metabolic language, which describes chemical reactions and bodily fluid equilibration in a manner lectured at medical faculties. Second, the current metabolic simulators are authority-accepted with a single day scenario and few meals only. The proposed framework targets multiple weeks. Third, Sirael can model other diseases, drugs and treatments, because its language is not bound to diabetes only. Fourth, Sirael required readily-available equipment only, when applied to Diabetes Type 1. We created 18 digital twins from real patients, with 37 identified initial states, average relative error 8.7% and standard deviation 8.0%. This provides free, open-source, and statistically significant base for conducting Diabetes Type 1 in-silico studies.

Keywords: diabetes, in-silico, metabolism, simulator, language, multi-meal, multi-sensor, digital-twin

1 Introduction

Diabetes mellitus is a heterogeneous group of diseases, which manifest with elevated blood glucose level [1]. Elevated glucose level continuously damages multiple organs, thus leading to both chronic and acute complications, while contributing to a development of additional diseases. Acute complications arise from glucose level being either too low or too elevated. Sorted by disease severity, such a condition can be treated with adjusted physical activity and dietary habits, drugs and eventually the insulin dosage.

Success of each treatment depends on its fitness to a specific patient and particular diabetes type. The major types are Type-1, Type-2, gestational and pre-diabetes condition. With Type-1, pancreas produces near-to-zero insulin quantity. Without insulin, cells cannot utilize glucose as energy source. With Type-2, pancreas produces insulin, but cells require a considerably increased quantity of it to utilize glucose. Such a condition is called insulin resistance. Gestational diabetes and pre-diabetes are associated with insulin resistance as well as the Type-2. Reasons for any diabetes condition is multi-factorial with contributions from both lifestyle and genetic makeup. For each patient, this creates a complex solution space of lifestyle change, drugs and insulin dosage. As a result, a metabolic simulator is an important step in conducting pre-clinical studies to reduce adverse effects of any proposed treatment, while reducing monetary costs. The financial burden [2, 3] does not result from conducting clinical studies only. Readmissions of individual patients to a hospital is yet another source of the financial burden, which may result from sub-optimal treatment. To underscore the need for a metabolic simulator, we have to mention patient's quality of life too[4]. A metabolic simulator could accelerate optimization of the personalized treatment plan, to a) improve glycemic control and b) increase patient's adherence to such a treatment by increasing his/her quality of life. To do so, we need to create a digital twin from the real patient with a high degree of realism.

A number of metabolic models were already proposed. As the current state of the art, let us identify three main directions with individual representatives.

- First, there is the Sorensen model designed in 1978. Over the time, it was revised and some errors were corrected [5]. It is a detailed model with well documented equations and parameter values. This model could be further improved to model other types of diabetes and other forms of glucose and insulin appearance. Nevertheless, this would require a modification of an already large set of differential equations.
- Second, there is the DMMS.R simulator [6], with roots in the UVA/Padova models. Recent version provides physical activity simulations, drug administration and multiple meals. DMMS.R is a source-code closed simulator with patient parameters being closed-source too. FDA accepted this simulator for performing in-silico studies.
- Third, a new approach emerged in the form of GCT modeling approach [7]. This approach is notable for
 - moving away from differential equations to integration to simplify the mathematical apparatus,
 - and promoting the best practices of software engineering at the level of multi-compartment decomposition of the model.

With all these models, we can identify several issues:

1. A model is a large set of equations, mostly differential ones. Although it is a valid approach from the engineering point of view, it is not an approach being lectured at medical faculties.
2. A meal does not have carbohydrate content only. There are other constituents with different glycemic and insulin index, which affect blood glucose level. Adding these constituents would be a substantial amount of work.

3. It would take an Artificial Intelligence (AI) to possibly recombine the compartments and substances automatically to incorporate associated diseases, drugs and realistic meal composition. For example, reference [8] demonstrates considerable amount of work needed to simulate insulin distribution in plasma with an expert-designed compartment decomposition.
4. Reference [9] presents the advancement from a single-meal scenario to a single-day scenario. Multiple days, or yet better multiple-sensors scenarios would provide a significantly increased degree of realism.
5. Model identifiability is a problem. To illustrate it, study [10] generated one hundred patients from a joint distribution of estimated model parameters. It would be considerably better to identify one hundred real patients.

In this review, we focused on the distinctive features to identify the main modeling approaches. A review of open-source projects for diabetes modeling was given, e.g.; at ATTD 2024[11].

In this paper, we propose the Sirael virtual language & metabolic machine, which addresses described issues of the current state of the art. To compare Sirael to the discussed state of the art, there are three distinct approaches. The Sorensen, DMMS.R and UVA/Padova models are based on differential equations. GCT is based on integration. Sirael is based on the metabolic language.

2 Sirael

2.1 Definitions

- *Sirael* is a general name for a metabolic simulation, which has its own metabolic language to describe metabolic processes.
- *Sirael Metabolic Language* aka Sirael Language is the language, which describes metabolic processes in a human readable form. Purpose of this language is to be friendly to both engineers and physicians.
- *Sirael IR* aka Siraela Intermediate Language is a compiled product of a source code written in the Sirael Metabolic Language.
- *SC* is the Sirael compiler of the Sirael Metabolic Language to Sirael IR and vice versa.
- *Sirael VM* is the virtual metabolic machine (VM), which executes Sirael IR.
- *Siraela* is a reference implementation of Sirael VM and other Sirael-related tools.
- Different vendors may produce their own VM and tools.
- *Simulated Subject* aka digital twin is the Siraela Metabolic Language or Sirael IR encoded subject with at least one initial Sirael state and all information needed to execute the simulation.
- *Sirael State* represents a simulated-subject state at a particular time. It is represented with quantities of various substances across multiple compartments. In a computer-science analogy, the most close comes the thread context represented with CPU registers. Sirael Metabolic Language and Sirael IR can contain multiple states, for a single subject. Before executing the simulation, it is necessary to choose the initial state.

Sirael stands for Simulated Reality as the simulated subject can be created from real subject. To make the name sound nice, it was twisted to a popular medieval-like times word Sirael, sometimes written as Zirael.

2.2 Workflow

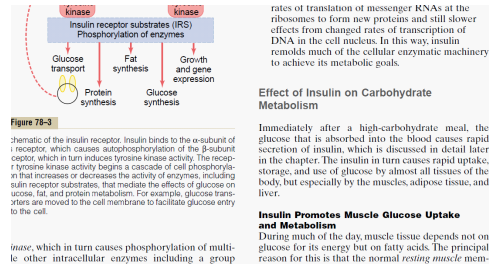
As depicted in Fig. 1 there is the following workflow to utilize Sirael:

1. Using the current state of the art of internal medicine, and other relevant scientific fields, an expert defines compartments, substances (including acting substances of drugs) and their interactions using the Sirael Metabolic Language. For example, a physician can translate a physiology textbook to the Sirael Metabolic Language as it is a physician-friendly language. Alternatively, AI expert can configure e.g.; Grammatical Evolution to produce the code.
2. As a human will likely produce a source code in the Sirael Metabolic Language, it has to be compiled using the Sirael Compiler to the Sirael IR code. AI is likely to produce the IR code directly.
3. Using an external solver, a computer/data scientist has to determine initial quantities of the substances and parameters of their interactions based on the input data.
4. Using Siraela Terminal, dynamic library or any other tool, Siraela VM is created and it loads the identified metabolism, encoded in the IR code. Sirael IR avoids the costs of parsing and validating textual metabolism representation.
5. The created VM is initialized with simulated subject's state and the VM is commanded to execute the simulation, while observing specified substances as their levels change during the simulation.

Sirael Terminal is a command-line utility, which accepts commands as given below. It creates Siraela VM, to subsequently modify its state and observe levels of designated substances in particular compartments. For example, glucose in blood and tissue represents arterial blood-glucose level and simulated CGM sensor respectively. Note that metabolism description and its parameters need to be coupled with subject's initial state – particular levels of substances in respective compartments. Therefore, VM must load both the metabolism and the state.

- **reset state_index Sirael_IR_file_path** – resets the VM to a particular state (identified by its ordinal number) to be simulated with a particular, IR-encoded metabolism
- **advance seconds** – advances the simulation by a specified number of seconds
- **emit** – emits the sample state as a collection of signals specified in the Sirael IR file
- **cho grams** – makes the subject to ingest specified number of carbohydrate grams
- **bolus units** – delivers the given amount of insulin units
- **heartbeat bpm** – sets current heartbeats per minute
- **consume signal_id quantity** – makes VM to consume a quantity of GUID-identified signal, which designates particular substance in a particular compartment

Guyton & Hall: Textbook of Medical Physiology



Physician, engineer, expert, AI (GE, GP) creates the metabolic program

.metabolism begin

/*passive*/ transport: glucose in tissue -> lymphatic_system
 active transport: insulin in insulin_depot -> tissue
 equalize: glucose in blood <-> muscle
 equalize: insulin in blood <-> muscle
 react: glucose & insulin -> pyruvate in muscle
 etc.

Sirael Compiler produces IR code

Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII
00000000:	53	69	72	61	65	6C	20	49	52	20	76	31	2E	30	00	0C	Sirael IR v1.0
00000010:	00	00	00	0A	00	00	00	80	96	98	00	00	00	00	00	47G
00000020:	16	00	00	00	00	00	00	4F	16	00	00	00	00	00	00	020
00000030:	17	00	00	00	00	00	00	67	00	00	00	00	00	00	00	04g
00000040:	00	00	00	00	00	00	00	E7	00	00	00	00	00	00	00	02J
00000050:	00	00	00	00	00	00	00	C7	0E	00	00	00	00	00	00	00
00000060:	00	00	00	00	00	00	00	7F	86	83	1A	09	2F	B6	4E	84N
00000070:	99	AC	1C	21	82	5A	2C	84	00	00	00	1E	00	00	00	00! Z,
00000080:	00	00	00	00	00	00	00	F0	22	4F	B4	92	E5	1E	49	BEm 7x0
00000090:	20	6D	F6	99	9A	C2	D8	84	00	00	00	28	00	00	00	00m (
000000A0:	00	00	00	00	00	00	00	F0	6A	AA	37	84	69	06	4A	92? j 7 i j
000000B0:	CC	A6	60	11	0D	0D	C7	02	00	00	00	00	00	00	00	5EA
000000C0:	BC	07	C4	AC	02	22	40	02	FD	FC	6D	8C	C4	E0	4C	B0@ m L

Sirael virtual metabolic machine loads the identified metabolism & state (described with the IR code) and executes the simulation (using terminal's advance, cho, emit, etc. or API)

Type help for a list of available commands.
 reset 0 sirael_ir.metabolism.t1d.sir

Success: Sirael VM is initialized and ready to use.
 advance 100
 Success: advance: 100; WallClock: 100; clock advanced
 cho 200
 Success: cho: 200; WallClock: 100; CHO delivered
 advance 300
 Success: advance: 300; WallClock: 400; clock advanced
 emit
 Emitting state (wall clock in seconds; signal id; level; signal human name
 Emit: 400; {B4412278-F592-491E-BE20-6DF6999AC2D8}; 4.13600969690685; IG (interstitial-fluid glucose level aka CGM sensor)
 Emit: 400; {1A03867F-2F09-4EB6-8499-AC1C21025A2C}; 4.203760019826461; BG (blood glucose level)
 Success: State emitted

Simulated metabolism can be visualized e.g.; with the Sirael API bindings to be used by different languages (C++, Python, JS...)

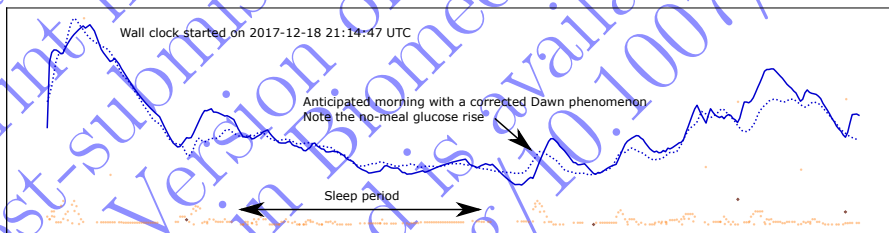


Fig. 1 Sirael Workflow

3 Metabolic Instructions

Any in-silico metabolic model, i.e., a large set of equations, is eventually compiled to machine instructions, dominantly the instruction sets of x86-64 and ARM computer processors. This effectively removes most physicians from contributing to the model as they usually have no expertise in advanced mathematical modeling nor software engineering. Although they could be users, they are unlikely contributors. Sirael addresses this issue by proposing a metabolic instruction set – a language, composed of physiological actions well-known to the physicians. Following instructions describe these actions.

1. Passive Transport – one way/facilitated diffusion of a single substance from one compartment to directly connected other compartment. It does not work against concentration gradient and it has a maximum transfer rate.
2. Equalization – according to the concentration gradient, a single substance equilibrates between two, directly-connected compartments at a given speed and maximum transfer rate.
3. Active Transport – it is a transport capable of working against the concentration gradient.
4. Reaction – it takes two substances to produce another one. For example, glucose and insulin produces adenosine triphosphate and pyruvate. Multiple reaction instructions describe production of each component, including intermediate products.
5. Inhibition – this is a reaction, whose reaction rate is suppressed by presence of a particular substance.
6. Set-Point Reaction – some reactions do not occur, if a particular substance is below or above a certain threshold.
7. No-Operation instruction – empty place as a helper for AI, when generating a metabolic code
8. End of Program – this instruction enables dynamic program length, primarily designed as a helper for AI

A physician could use these instructions to compose a metabolic program as illustrated with the following example. /*...*/ symbols denote a comment.

```
/*passive*/ transport: glucose in tissue -> lymphatic_system
active transport: insulin in insulin_depot -> tissue
equalize: glucose in blood <-> muscle
equalize: insulin in blood <-> muscle
react: glucose & insulin -> pyruvate in muscle
etc.
```

The acting quantity of each substance is determined with a Michaelis-Menten-like kinetics. The following equation describes it. Mass is the total quantity of a given substance being present in a given compartment. Factor is a parameter of a particular reaction.

$$acting\ mass = \begin{cases} \frac{mass}{mass+factor} & : mass > 0 \\ 0 & : mass = 0 \\ error & : mass < 0 \end{cases}$$

When compared to the Michaelis-Menten kinetics, the maximum rate is fused with parameters of the invoked instructions (thus reducing the number of parameters to identify). Then, mass is the concentration of substrate and factor is the constant. Once the program is established, an external solver such as Meta-Differential Evolution, Particle Swarm Optimization and others can be used to identify parameters of each instruction and initial quantity of each substance in each compartment.

To enable AI to adapt the program to a specific glucose time series, meal & drugs journal of a specific patient, Sirael uses very long instruction word architecture [12]. With this architecture, methods such as Grammatical Evolution (GE) and Genetic Programming (GP), can easily alter opcodes of instructions, compartments and substances, while optimizing their parameters.

Sirael VM loads a metabolic program, which declare modeled compartments and substances. Therefore, it is possible to add new compartments and new substances as needed to the model either manually or automatically, to specify any meal composition, drug or associated disease per patient. As a result, Sirael is not bounded to the Diabetes Mellitus only.

3.1 Computational Complexity

Current state-of-the-art models are well-defined with an exactly limited set of compartments, substances and their meanings. As result, their solution search-space is smaller and well defined. In the pursue of universality, Sirael sets no limits, which makes it hard to identify the parameters. It has a generic, large search-space.

Although it is possible to extend Sirael VM with new instructions, we decided not to do so. If we would do so, we would increase the Sirael search-space further. This is the reason, for e.g., having the reacting instruction to produce a single substance only, instead of two substances. To produce two substances, two reacting instructions must be chained as a universal solution. Adding new metabolic instructions could be done in the future, once it is rewarded by a respective progress in methods needed to the identify the parameters.

4 Experimental Setup

JAEB Center for Health Research has released a dataset of "An Observational Study of Individuals with Type 1 Diabetes Using the Loop System for Automated Insulin Deliver" [13]. From this dataset, we extracted the following time series:

- Measured blood glucose level
- Subcutaneous glucose level measured with a sensor
- Carbohydrate intake
- Calories burned during exercise and exercise time
- Insulin delivery – we transformed insulin basal rate to a sequence of micro-boluses to simulate a real insulin pump

Then, we extracted additional patient information, from which we estimated the basal metabolism need of the body. We considered exercise as an additional metabolic need. Eventually, we inputted insulin delivery, carbohydrate intake and the resulting metabolic need to the Sirael Virtual Metabolic Machine.

A Sirael metabolic program contains:

- Compartment and substance definitions
- Instructions describing the metabolism along their parameters, which personalize the program to a particular, real patient
- Initial quantities of the substances in the defined compartments

To prove that Sirael is capable of multiple-sensor scenario, we optimized a metabolic program over the entire range of measurements, which spanned across several months. For each patient, we partitioned the measured time series to multiple segments of continuous glucose-sensor measurements. For each segment, we identified the initial substance quantities, while requiring that the metabolic-program parameters stay the same for all the segments. This is important because:

- We over-determined Sirael parameters, thus achieving a stable solution.
- It proves that Sirael works by an explainable design, not by a coincidence.
- It demonstrates that a real patient can be identified with a readily available equipment. The identification can be done without a positron emission tomography to monitor glucose-like tracers, or to perform any other time and money consuming procedures to obtain reference quantities, which result to relatively small, but expensive datasets from which artificial, digital twins are generated then with statistical methods.

To create the digital twin, we identify instruction parameters and the initial quantities per each segment. For this purpose, we created a training set by reducing the original time series by 30%. Then, we used a random-seeded Differential Evolution to estimate the quantities and parameters. To validate this approach, we evaluated the simulation with the original series, which served as the testing set. Thus, we observed how well the simulation performed under circumstances, which were not observed during the training phase. We required that each segment lasted at least 24 hours to guarantee over-determination of Sirael metabolic-program parameters.

5 Results and Discussion

The publicly available dataset was a limiting factor. For example, a heartbeat sensor might have provided additional time series to estimate the physical activity more precisely than the exercise-calories estimate only. Blood glucose levels are mostly missing, making it difficult (not impossible) to capture the blood-interstitium variability [14]. Along with the precision of measuring devices, this contributed to the resulting error. Nevertheless, the resulting error is still acceptable and we consider this dataset as an important step in the diabetes research.

The initial quantities and the current metabolic-need together determine future course of glucose level. By initializing the Sirael VM with an initial state, it computes

the glucose level based on the user input – insulin dosing, food intake and physical activity. As each in-silico patient was identified using multiple segments, the patient has multiple initial states (one state per segment).

When identifying the parameters, we used exactly the same metabolic program for all the patients. This actually introduces an error, because some patients may have had additional diseases - either short-term or chronic ones. To avoid complexity needed to model of unknown, additional diseases, we considered Diabetes Type 1 as the only disease.

We allowed optimization of metabolic-program parameters only within a single subject. Within a single subject, all states are evolved using the same metabolic program with the same parameters. We did so to guarantee the over-determination as the patients provided measurements, which spanned across several months. Due to such a long time span, it is logical to assume some illness, which occurred but was not reported in the dataset. As such an illness might have had altered metabolic needs and activated different metabolic pathways e.g. due to additional drugs, the assumed metabolic program had to exhibit excessive errors under such circumstances. As we considered Diabetes Type-1 as the only disease, we excluded those segments, whose average relative error exceeded 10%. In a principle, we cannot get too close the sensor measurement error, because the sensor itself exhibits a measuring error. In addition, there was a practical reason for doing so – to limit the electrical power needed to identify larger set of states with desired accuracy. Therefore, it is likely that enabling more computing power would reduce the fitting error further to identify more initial states with average relative error up to 10%.

Table 1 gives cumulative markers of the identified segments. Individual statistics and plots for each segment are available in the released dataset. Figures 2-3 illustrates identified segments. Particularly, they demonstrate that it is possible to fit multiple days with multiple meals with various phenomenon using the same metabolic program and readily available time series. Solid lines demonstrate simulated CGM signal, while dotted lines demonstrate measured CGM signal. Diamonds represent meal ingestion. Insulin boluses and micro-boluses (derived from the insulin basal rate) are represented with small, semi-transparent circles.

CGM sensor exhibits physiological and technological lags, which arise from the equilibration of bodily fluids, chemical processes occurring at the sensor needle, and from algorithms used to smooth the measured signal – the reported glucose level. Sirael VM outputs directly a simulated substance-level, thus applying no such algorithms. Therefore, it has to exhibit a difference from the sensor signal by a principle. The simulated signal should exhibit a greater rate of change due to the absence of signal smoothing/denoising filters.

If we would interpret insulin basal rate as a flat increase of insulin in the simulated subcutaneous insulin depot, we would obtain non-realistic behavior. No insulin pump runs its motor permanently. Rather, they run the motor multiple times with a specified running time and non-running period, to inject the specified amount of insulin during a designated period (1 hour typically). Therefore, we transformed the insulin basal rates to a set of micro-boluses to obtain the realistic behavior. Particular amounts can be inspected in the released datasets.

6 Conclusion and Future Work

Let us consider that we identify a precise digital twin using the Sirael VM and a readily available equipment. Then, we can create a personalized diabetes-treatment plan – optimal drug administration, dietary plan, exercise duration and intensity. It would be no longer so frustrating [15] for the patient to visit a physician, agree on some treatment rules, follow those rules for weeks and then make another visit to adjust these rules, and repeat. With Sirael, such a tedious and frequently demotivating burden could be lifted from the patient with the in-silico, metabolic simulation.

Not limited to such a use-case, Sirael could be useful to estimate drug kinetics, development of insulin-pump controller and for making education software, e.g.: games where it could provide the gaming physics to support kids and teenage patients.

To support the vision, we released 18 digital twins with initial 37 initial states, which we identified from real patients with average relative error 8.7% and standard deviation 8.0%. This provides free, open-source, and statistically significant base for conducting Diabetes Type 1 in-silico studies.

In the future, we will continue with releasing new identified subjects. For example, the data repository already host an attempt to identify pre-diabetes patients. From the Type-1 in-silico population, they just differ in the metabolic program but share the same parameter-identification procedure. This proves the practical aspect of the proposed Sirael metabolic language & VM concept.

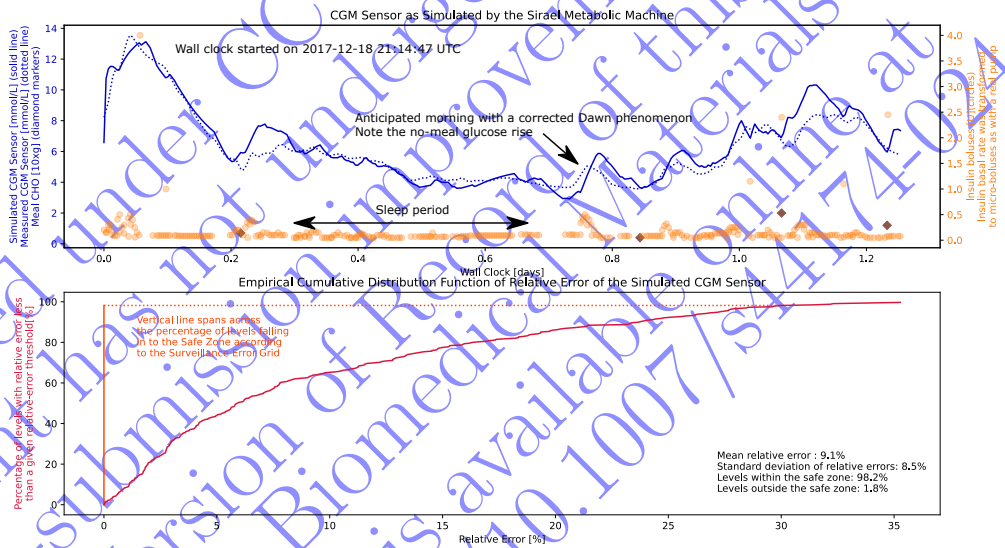


Fig. 2 Patient ID 821, Segment ID 60; evening - night - morning - afternoon

Acknowledgments. The source of the data is the Loop Study (sponsored by the Jaeb Center for Health Research and funded by the Helmsley Charitable Trust), but

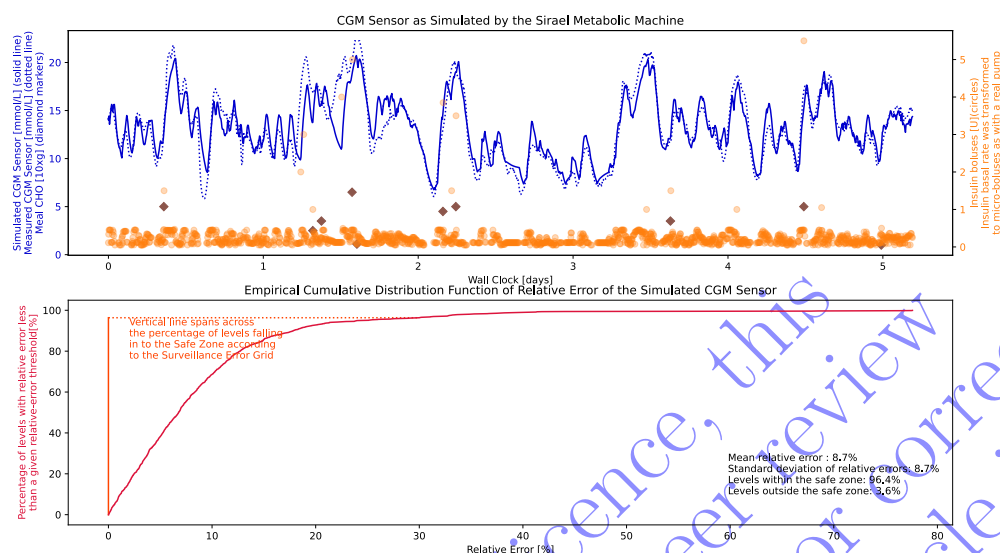


Fig. 3 Patient ID 763, Segment ID 85; 5 days

Table 1 Cumulative markers for all in-silico captured patient segments.

Marker	Absolute Error [mmol/L]	Relative Error [%]
Average	0.78	8.7
Standard Deviation	0.80	8.0
1st Quartile	0.24	2.9
Median	0.54	6.4
2nd Quartile	1.35	18.0
78th Percentile	1.03	11.7
90th Percentile	1.75	18.9
95th Percentile	2.09	22.8
Fitted Sensor Samples	15607 aka 54 days	
Identified In-Silico Segments	37	
Identified In-Silico Patients	18	

the analyses, content and conclusions presented herein are solely the responsibility of the authors and have not been reviewed or approved by the study sponsor.

Conflict of Interest

Tomas Koutny is a consultant to Abbott Laboratories, which own the DMMS.R metabolic simulator. GCT is a research project of Martin Ubl, who is a doctoral candidate supervised by Tomas Koutny.

Data and Code Availability

Siraela source code, selected in-silico transformed patients, the digital twins, and experimental setups are available for open-source download at the following repository:

https://gitlab.com/sirael_metabolic_simulator

References

- [1] Jameson, J.L., Fauci, A.S., Kasper, D.L., Hauser, S.L., Longo, D.L., Loscalzo, J.: Harrison's Principles of Internal Medicine, Twentieth Edition (Vol.1 & Vol.2). McGraw-Hill Education, New York (2018)
- [2] Herman, W.H.: The global burden of diabetes: an overview. Diabetes mellitus in developing countries and underserved communities, 1–5 (2017)
- [3] Robbins, T.D., Keung, S.L.C., Sankar, S., Randeva, H., Arvanitis, T.: Risk factors for readmission of inpatients with diabetes: a systematic review. Journal of Diabetes and its Complications **33**(5), 398–405 (2019)
- [4] Speight, J., Holmes-Truscott, E., Hendrieckx, C., Skovlund, S., Cooke, D.: Assessing the impact of diabetes on quality of life: what have the past 25 years taught us? Diabetic Medicine **37**(3), 483–492 (2020)
- [5] Panunzi, S., Pompa, M., Borri, A., Piemonte, V., De Gaetano, A.: A revised sorenson model: Simulating glycemic and insulinemic response to oral and intravenous glucose load. Plos one **15**(8), 0237215 (2020)
- [6] Ubl, M., Koutny, T., Della Cioppa, A., De Falco, L., Tarantino, E., Scafuri, U.: Distributed assessment of virtual insulin-pump settings using SmartCGMS and DMMS.R for diabetes treatment. Sensors **22**(23), 9445 (2022)
- [7] Ubl, M., Koutny, T.: A novel approach to multi-compartmental model implementation to achieve metabolic model identifiability on patient's cgm data. Procedia Computer Science **210**, 116–123 (2022)
- [8] Visentin, R., Schiavon, M., Giegerich, C., Klabunde, T., Man, C.D., Cobelli, C.: Incorporating long-acting insulin glargine into the uva/padova type 1 diabetes simulator for in silico testing of mdi therapies. IEEE Transactions on Biomedical Engineering **66**(10), 2889–2896 (2019) <https://doi.org/10.1109/TBME.2019.2897851>
- [9] Visentin, R., Campos-Náñez, E., Schiavon, M., Lv, D., Vettoretti, M., Breton, M., Kovatchev, B.P., Dalla Man, C., Cobelli, C.: The uva/padova type 1 diabetes simulator goes from single meal to single day. Journal of diabetes science and technology **12**(2), 273–281 (2018)
- [10] Visentin, R., Cobelli, C., Dalla Man, C.: The padova type 2 diabetes simulator

from triple-tracer single-meal studies: In silico trials also possible in rare but not-so-rare individuals. *Diabetes technology & therapeutics* **22**(12), 892–903 (2020)

- [11] Peuscher, H., *et al.*: A survey on existing open-source projects in diabetes simulation. In: *The 17th International Conference on Advanced Technologies and Treatments for Diabetes* (2024)
- [12] Fisher, J.A.: Very long instruction word architectures and the eli-512. In: *Proceedings of the 10th Annual International Symposium on Computer Architecture*, pp. 140–150 (1983)
- [13] Lum, J.W., Bailey, R.J., Barnes-Lomen, V., Naranjo, D., Hood, K.K., Lal, R.A., Arbiter, B., Brown, A.S., DeSalvo, D.J., Pettus, J., *et al.*: A real-world prospective study of the safety and effectiveness of the loop open source automated insulin delivery system. *Diabetes technology & therapeutics* **23**(5), 367–375 (2021)
- [14] Guyton, A.C., Hall, J.E.: *Textbook of Medical Physiology*, 11e (Guyton Physiology). Saunders, Philadelphia (2005)
- [15] Craven, M., Simons, Z., Groot, M.: Diabetes distress among healthcare providers: a qualitative study. *Diabetes Research and Clinical Practice* **150**, 211–218 (2019)