Using low-rank tensor formats to enable computations of cancer progression models in large state spaces

Simon Pfahler[1], Peter Georg[1], Y. Linda Hu[2], Stefan Vocht[2], Rudolf Schill[2,3], Andreas Lösch[2], Kevin Rupp[2,3], Stefan Hansch[2], Maren Klever[4], Lars Grasedyck[4], Rainer Spang[2], Tilo Wettig[1]

[1]Department of Physics, University of Regensburg
[2]Department of Informatics and Data Science, University of Regensburg
[3]Department of Biosystems Science and Engineering, ETH Zürich
[4]Institute for Geometry and Applied Mathematics, RWTH Aachen University

Scan for digital version

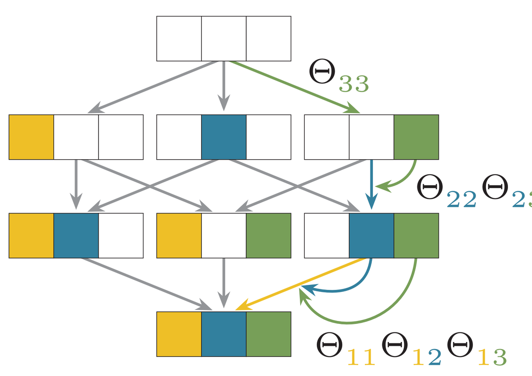
Summary

- Comprehensive cancer progression models should include a high number D of genomic events
- Mutual Hazard Networks model the progression process using only D^2 parameters [1]
  - Computational complexity of a straight-forward implementation still scales exponential in D
  - Calculations using $\gtrsim 25$ events are computationally infeasible [2]
- Tensor Trains allow for cost-efficient storage and calculations for high-dimensional tensors
  - This method reduces the computational complexity from exponential to polynomial in D

Mutual Hazard Network (MHN) model

- MHN models tumor progression as a continuous-time Markov chain on the 2^D-dimensional state space of possibly active events [1]
- Events can only occur one at a time
- Transition rates are given by

$$\mathbf{Q}_{\mathbf{x}\to\mathbf{x}_{+i}} = \underbrace{\Theta_{ii}}_{\text{base rate}} \prod_{x_j=1} \underbrace{\Theta_{ij}}_{\text{influences}}$$



- $\mathbf{Q}_\Theta \in \mathbb{R}^{2^D\times 2^D}$ is sparse, Θ-matrix describes the model in compact form using only D^2 parameters
- Time marginal probability distribution from Θ:

$$\mathbf{p}_\Theta = (\mathbb{1} - \mathbf{Q}_\Theta)^{-1}\,\mathbf{p}_\varnothing, \qquad \mathbf{p}_\varnothing = (100\%, 0\%, 0\%, \dots, 0\%)^T$$

- Optimal Θ matrices are found by optimizing the time marginalized Kullback-Leibler divergence from the given data distribution $\mathbf{p}_\mathcal{D}$:

$$S_{\mathrm{KL}}(\mathbf{p}_\Theta) = \sum_{\mathbf{x}} (\mathbf{p}_\mathcal{D})_\mathbf{x} \log\left((\mathbf{p}_\Theta)_\mathbf{x}\right)$$

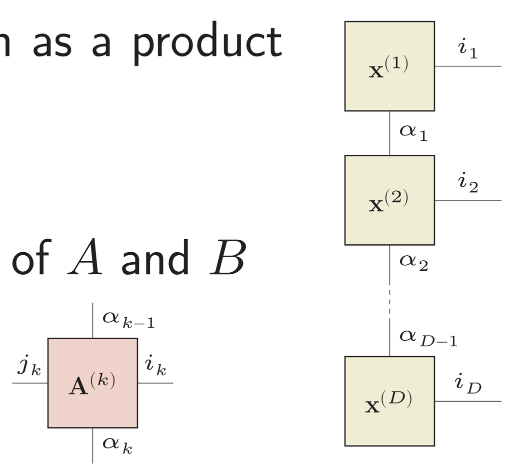
- Gradients can be calculated analytically:

$$\frac{\partial S_{\mathrm{KL}}}{\partial \Theta_{ij}} = \underbrace{\frac{\partial S_{\mathrm{KL}}}{\partial (\mathbf{p}_\Theta)_\mathbf{x}} (\mathbb{1}-\mathbf{Q}_\Theta)^{-1}_{\mathbf{xy}}}_{\mathbf{q}_\mathbf{y}} \left(\frac{\partial \mathbf{Q}_\Theta}{\partial \Theta_{ij}}\right)_{\mathbf{yz}} (\mathbf{p}_\Theta)_\mathbf{z}$$

- Score and gradient calculation time is dominated by solution time of two linear equations:

$$(\mathbb{1}-\mathbf{Q}_\Theta)\,\mathbf{p}_\Theta = \mathbf{p}_\varnothing \qquad (\mathbb{1}-\mathbf{Q}_\Theta)^T\,\mathbf{q} = \frac{\partial S_{\mathrm{KL}}}{\partial \mathbf{p}_\Theta}$$

Tensor Train (TT) representation



- D-dimensional tensors $\mathbf{x} \in \mathbb{C}^{n_1\times\dots\times n_D}$ can be written as a product of D Tensor Train cores $\mathbf{x}^{(k)} \in \mathbb{C}^{r_{k-1}\times n_k\times r_k}$:

$$\mathbf{x}(i_1,\dots,i_D) = \mathbf{x}^{(1)} \circ \dots \circ \mathbf{x}^{(D)}$$

  $A \circ B$ denotes contraction of the last and first indices of A and B
- Similar for operators $\mathbf{A} \in \mathbb{C}^{(m_1\times\dots\times m_D)\times(n_1\times\dots\times n_D)}$: TT cores $\mathbf{A}^{(k)} \in \mathbb{C}^{r_{k-1}\times m_k\times n_k\times r_k}$
- Storage cost goes from exponential to linear in D, but additional dependency on TT ranks is introduced
- Many arithmetic operations are performable directly in the TT format, reducing the computational complexity [3]:
  - Superposition $\lambda\mathbf{a}+\nu\mathbf{b}$: $\mathcal{O}\left(Dn(r_\mathbf{a}+r_\mathbf{b})^2\right)$
  - Inner product $\langle\mathbf{a},\mathbf{b}\rangle$: $\mathcal{O}\left(Dnr_\mathbf{a}r_\mathbf{b}(r_\mathbf{a}+r_\mathbf{b})\right)$
  - Operator-by-Tensor product $\mathbf{Ab}$: $\mathcal{O}\left(Dmn(r_\mathbf{A}r_\mathbf{b})^2\right)$

  $n := \max(n_k)$, analogously for $m, r_\mathbf{x}$
- Linear equations $\mathbf{Ax}=\mathbf{b}$ can also be solved efficiently directly in the format

References

[1] R. Schill, S. Solbrig, T. Wettig, and R. Spang, *Modelling cancer progression using mutual hazard networks*, *Bioinformatics* **36** (January, 2020) 241.

[2] P. Georg, L. Grasedyck, M. Klever, R. Schill, R. Spang, and T. Wettig, *Low-rank tensor methods for markov chains with applications to tumor progression models*, *Journal of Mathematical Biology* **86** (December, 2022).

[3] P. Georg, *Tensor train decomposition for solving high-dimensional mutual hazard networks*, PhD thesis, Universität Regensburg, October, 2022.

Tensor Trains for MHN

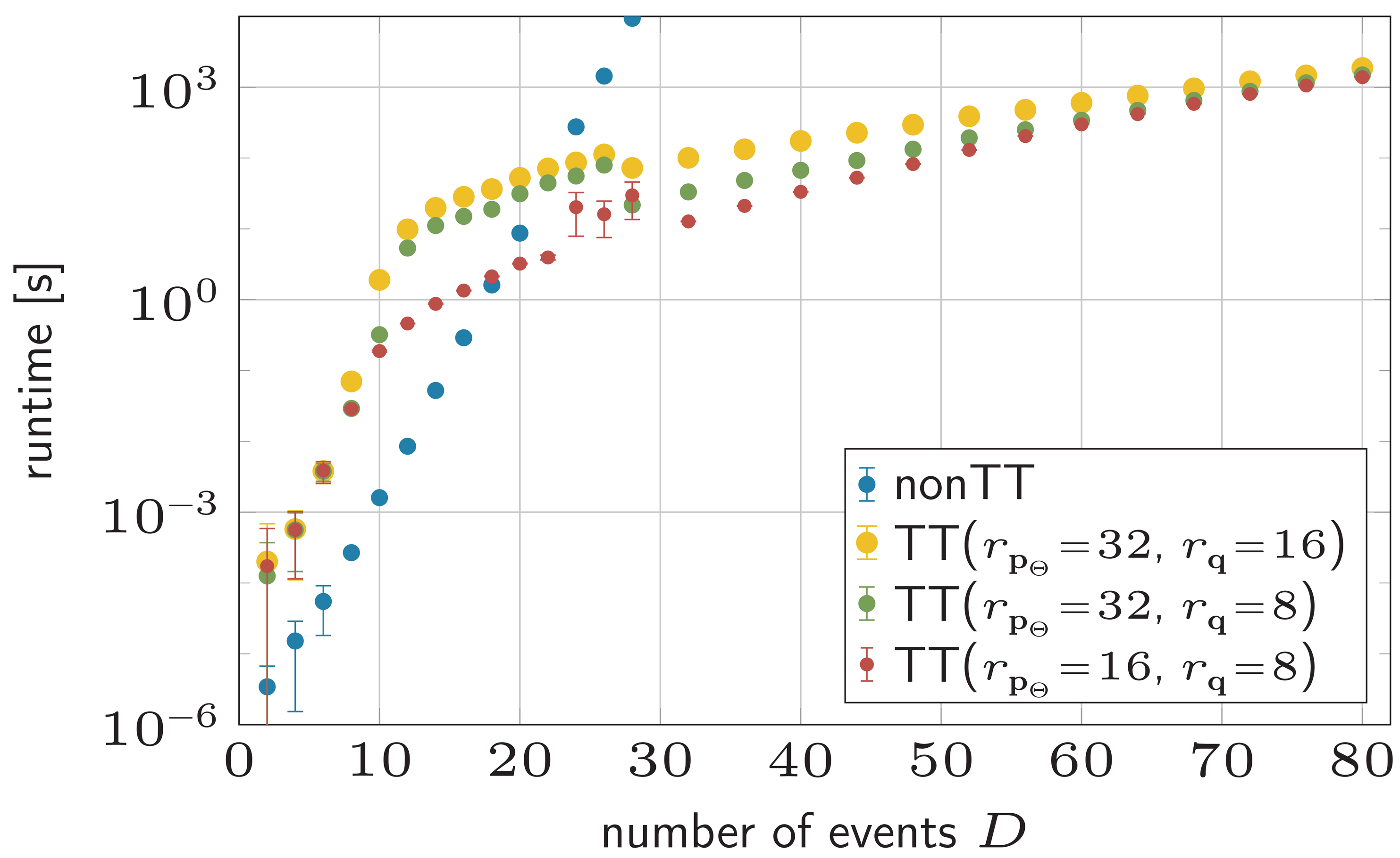
- Events are binary $\to n_k = 2$ for all mode sizes
- $\mathbf{Q}_\Theta$ can naturally be written as a Tensor Train [1]:

$$\mathbf{Q}_\Theta = \sum_{i=1}^{D}\left(\bigotimes_{j=1}^{i-1}\underbrace{\begin{pmatrix}1 & 0\\ 0 & \Theta_{ij}\end{pmatrix}}_{\in\mathbb{R}^{1\times2\times2\times1}} \otimes \underbrace{\begin{pmatrix}-\Theta_{ii} & 0\\ \Theta_{ii} & 0\end{pmatrix}}_{\in\mathbb{R}^{1\times2\times2\times1}} \otimes \bigotimes_{j=i+1}^{D}\underbrace{\begin{pmatrix}1 & 0\\ 0 & \Theta_{ij}\end{pmatrix}}_{\in\mathbb{R}^{1\times2\times2\times1}}\right)$$

  - $\mathbf{Q}_\Theta$ is a sum of D rank-1 Tensor Trains
  - $\mathbf{Q}_\Theta$ has TT-ranks D
- $\mathbf{p}_\varnothing$ is a canonical unit Tensor Train, also has TT-rank 1
- $\mathbf{p}_\Theta$ and $\mathbf{q}$ can be calculated in the TT format (max. TT ranks $r_{\mathbf{p}_\Theta}$ and $r_\mathbf{q}$)
- for gradients, each non-zero element in $\mathbf{p}_\mathcal{D}$ has to be treated individually
  - one linear equation has to be solved for each entry of $\mathbf{p}_\mathcal{D}$ (usually ~ 1000)
  - this can trivially be parallelized

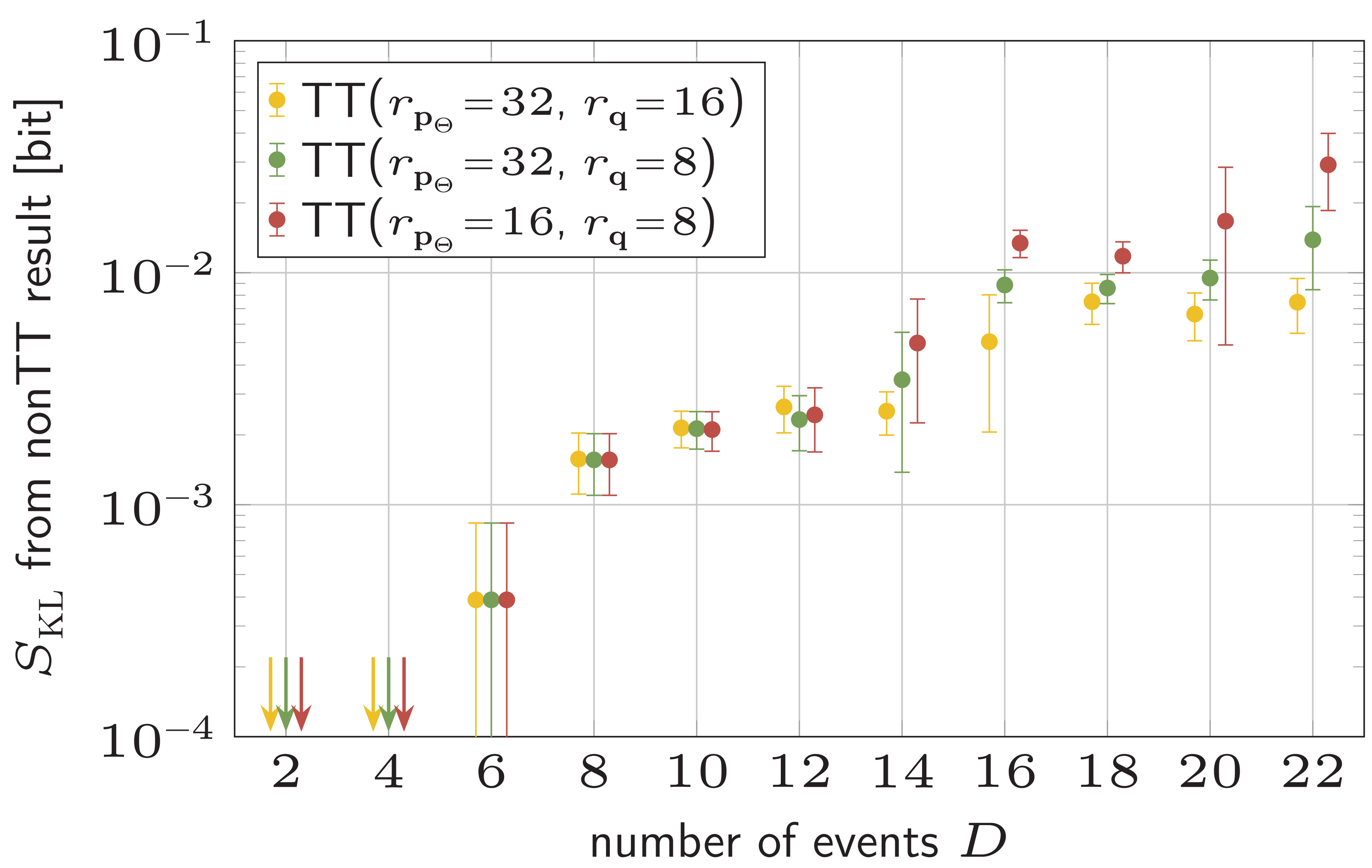
Results: Runtime improvement

- Runtime for one score and gradient calculation
- $\mathbf{p}_\mathcal{D}$ constructed from 1000 random samples, Θ at independence model
- Runtime grows with $\sim D^{5.4}$ for large $D \Rightarrow$ **polynomial growth!**



Results: Accuracy of the TT solution

- KL-divergence from nonTT result after full optimization of Θ



Code availablility

- C++ library for TT-calculations pRC: gitlab.com/pjgeorg/pRC
- application-specific C++ library cMHN that utilizes pRC for MHN-calculations: soon to be open-source

Future improvements

- Reduce runtime by accellerating solution of linear equations in the TT format
- Include development of metastases into the model

Supported by the German Research Foundation (DFG)

Correspondence: simon.pfahler@ur.de