Tensor Computations: Efficiency Or Productivity?

Paolo Bientinesi – Umeå Universitet & RWTH Aachen University
Rasmus Bro – University of Copenhagen
Edoardo Di Napoli – Jülich Supercomputing Centre
Lars Karlsson – Umeå Universitet

March 1, 2019
SIAM Conference on Computational Science and Engineering
Spokane, Washington, USA

HPAC High Performance and Automatic Computing

Part I

The HPC perspective

Disclaimer: Talk about awareness, not results

Typical HPC “workflow” – not limited to tensor computations

- **1. Target problem**
  - Fixed problem size?
  - Target architecture? (parallel paradigm)

Typical HPC "workflow" – not limited to tensor computations

- **1. Target problem**
  - Fixed problem size?
  - Target architecture? (parallel paradigm)
- **2.** Identify building blocks, bottlenecks, hotspots, critical paths, . . .

Typical HPC "workflow" – not limited to tensor computations

- **1. Target problem**
  - Fixed problem size?
  - Target architecture? (parallel paradigm)
- **2.** Identify building blocks, bottlenecks, hotspots, critical paths, ...
- **3.** Algorithmic & code optimizations
  Objectives: speedups, scalability, portability, ...

Typical HPC "workflow" – not limited to tensor computations

- **1. Target problem**
  - Fixed problem size?
  - Target architecture? (parallel paradigm)

- **2.** Identify building blocks, bottlenecks, hotspots, critical paths, ...

- **3.** Algorithmic & code optimizations
  Objectives: speedups, scalability, portability, ...

- : High exploitation of platform's potential, time savings, green computing
  (publications, "more science per hour", ...)

Typical HPC “workflow” – not limited to tensor computations

- **1. Target problem**
  - Fixed problem size?
  - Target architecture? (parallel paradigm)
- **2.** Identify building blocks, bottlenecks, hotspots, critical paths, . . .
- **3.** Algorithmic & code optimizations
  Objectives: speedups, scalability, portability, . . .
- ☺: High exploitation of platform’s potential, time savings, green computing
  (publications, “more science per hour”, . . .)
- ☹: However, ... often limited integration into actual scientific codes
  (little funding, not researchy/publishable effort, no credits, . . .)

Typical HPC "workflow" – not limited to tensor computations

- **1. Target problem**
  - Fixed problem size?
  - Target architecture? (parallel paradigm)
- **2.** Identify building blocks, bottlenecks, hotspots, critical paths, . . .
- **3.** Algorithmic & code optimizations
  Objectives: speedups, scalability, portability, . . .
- ☺: High exploitation of platform's potential, time savings, green computing
  (publications, "more science per hour", . . .)
- ☹: However, ... often limited integration into actual scientific codes
  (little funding, not researchy/publishable effort, no credits, . . .)

⇒ **Caveat**: Gains in the building blocks... often lost at the higher levels

Examples (1/2) Genome-Wide Association Studies (GWAS)

- What is it? Data correlation analysis. 2D grid of generalized least squares problems (GLS)

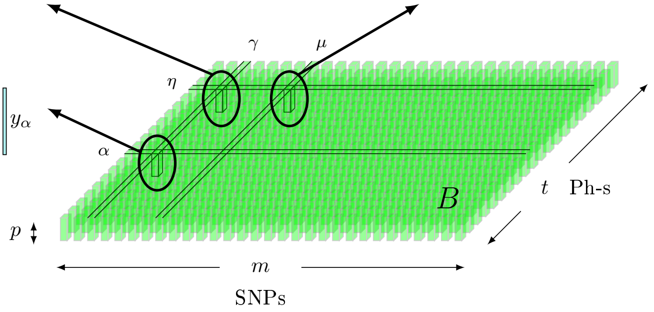
Examples (1/2) Genome-Wide Association Studies (GWAS)

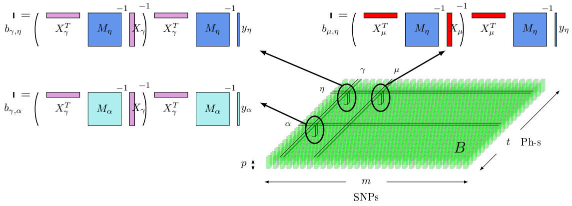
- How is it related to tensors? 1D×1D cartesian product of GLSs, 2D output = 4D data

$$b_{\gamma,\eta} = \left(X_\gamma^T M_\eta^{-1} X_\gamma \right)^{-1} X_\gamma^T M_\eta^{-1} y_\eta$$

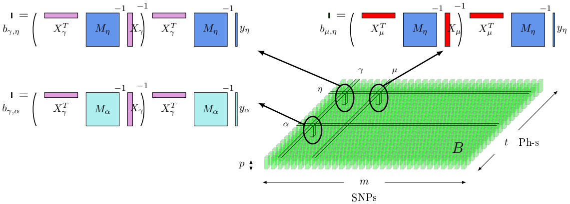
$$b_{\mu,\eta} = \left(X_\mu^T M_\eta^{-1} X_\mu \right)^{-1} X_\mu^T M_\eta^{-1} y_\eta$$

$$b_{\gamma,\alpha} = \left(X_\gamma^T M_\alpha^{-1} X_\gamma \right)^{-1} X_\gamma^T M_\alpha^{-1} y_\alpha$$





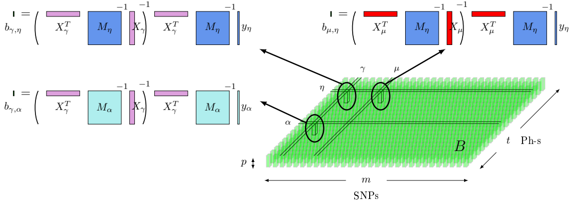
- Algorithmic improvement: lower computational complexity
- HPC optimizations: asynch I/O, overlap, BLAS-3, parallelism, ...



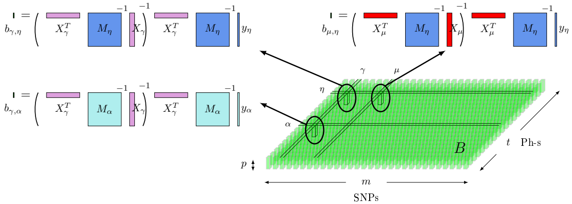
Computing Petaflops over Terabytes of Data: The Case of Genome-Wide Association Studies.
ACM TOMS, 2014

- Algorithmic improvement: lower computational complexity
- HPC optimizations: asynch I/O, overlap, BLAS-3, parallelism, ...
- 100x – 1000x speedups! Library available: `OmicABEL`

Computing Petaflops over Terabytes of Data: The Case of Genome-Wide Association Studies.
ACM TOMS, 2014



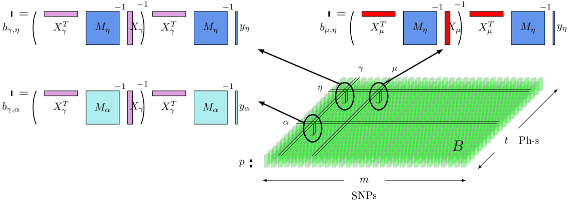
- Algorithmic improvement: lower computational complexity
- HPC optimizations: asynch I/O, overlap, BLAS-3, parallelism, ...
- 100x – 1000x speedups! Library available: `OmicABEL`
- But...



Computing Petaflops over Terabytes of Data:
The Case of Genome-Wide Association Studies.
ACM TOMS, 2014

- Algorithmic improvement: lower computational complexity
- HPC optimizations: asynch I/O, overlap, BLAS-3, parallelism, . . .
- 100x – 1000x speedups! Library available: `OmicABEL`
- But...
  - Interface: C vs. R
  - Data management: data formats, overwriting, multiple files
  - Data manipulation: imputation, filtering, selection
  - Workflow not as fixed as first understood: M vs no-M, . . .
  - *“The pre-processing is slower than the analysis”*

Computing Petaflops over Terabytes of Data:
The Case of Genome-Wide Association Studies.
ACM TOMS, 2014



- Algorithmic improvement: lower computational complexity
- HPC optimizations: asynch I/O, overlap, BLAS-3, parallelism, ...
- 100x – 1000x speedups! Library available: `OmicABEL`
- But...
  - Interface: C vs. R
  - Data management: data formats, overwriting, multiple files
  - Data manipulation: imputation, filtering, selection
  - Workflow not as fixed as first understood: M vs no-M, ...
  - *"The pre-processing is slower than the analysis"*

⇒ Performance is important, but not as much as we like to think

Examples (2/2)

- **Tensor Transpositions**

$$\mathcal{B}_{i_1 i_2 \ldots i_N} \leftarrow \alpha \cdot \mathcal{A}_{\pi(i_1 i_2 \ldots i_N)} + \beta \cdot \mathcal{B}_{i_1 i_2 \ldots i_N}$$

- **Summations** — linear summation over tensor transpositions

$$\mathcal{B}_{i_0 i_1 i_2} \leftarrow 2\mathcal{A}_{i_0 i_1 i_2} - \mathcal{A}_{i_2 i_1 i_0} - \mathcal{A}_{i_0 i_2 i_1}$$

$$\mathcal{B}_{i_0 i_1 i_2} \leftarrow 4\mathcal{A}_{i_0 i_1 i_2} - 2\mathcal{A}_{i_1 i_0 i_2} - 2\mathcal{A}_{i_2 i_1 i_0} + \mathcal{A}_{i_1 i_2 i_0} - 2\mathcal{A}_{i_0 i_2 i_1} + \mathcal{A}_{i_2 i_0 i_1}$$

$$\mathcal{B}_{i_0 i_1 i_2 i_3} \leftarrow 2\mathcal{A}_{i_0 i_1 i_2 i_3} - \mathcal{A}_{i_2 i_1 i_0 i_3} - \mathcal{A}_{i_0 i_2 i_1 i_3} - \mathcal{A}_{i_0 i_1 i_3 i_2}$$

- **Tensor Contractions**

$$\mathcal{C}_{\pi_{\mathcal{C}}(I_m \cup I_n)} \leftarrow \alpha \cdot \mathcal{A}_{\pi_{\mathcal{A}}(I_m \cup I_k)} \times \mathcal{B}_{\pi_{\mathcal{B}}(I_n \cup I_k)} + \beta \cdot \mathcal{C}_{\pi_{\mathcal{C}}(I_m \cup I_n)}$$

- **Tensor Transpositions**

  TTC: A high-performance Compiler for Tensor Transpositions. ACM TOMS, 2018

  Compiler: `https://github.com/HPAC/TTC` **Library**: `https://github.com/HPAC/hptt`

- **Summations** — linear summation over tensor transpositions

  Spin Summations: A High-Performance Perspective. ACM TOMS, 2019

  Generator: `https://github.com/springer13/spin-summations`

- **Tensor Contractions**

  Design of a high-performance GEMM-like Tensor-Tensor Multiplication. ACM TOMS, 2018

  Compiler: `https://github.com/HPAC/tccg` **Library**: `https://github.com/springer13/tcl`

But...

But...

- “Wrong” level of abstraction for domain scientists

But...

- “Wrong” level of abstraction for domain scientists
- Kernels: good for developers – too low level for most end users

But...

- “Wrong” level of abstraction for domain scientists
- Kernels: good for developers – too low level for most end users
- Mismatch $\rightarrow$ mapping problem

2D case: "Right" level of abstraction

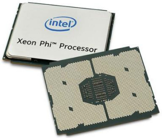
Generalized Least Squares	$b := (X^T M^{-1} X)^{-1} X^T M^{-1} y$	$n > m$; $M \in \mathbb{R}^{n \times n}$, SPD; $X \in \mathbb{R}^{n \times m}$; $y \in \mathbb{R}^{n \times 1}$
Signal Processing	$x := \left(A^{-T} B^T B A^{-1} + R^T L R\right)^{-1} A^{-T} B^T B A^{-1} y$	
Kalman Filter	$K_k := P_k^b H^T (H P_k^b H^T + R)^{-1}$; $x_k^a := x_k^b + K_k (z_k - H x_k^b)$; $P_k^a := (I - K_K H) P_k^b$	
Ensemble Kalman Filter	$X^a := X^b + \left(B^{-1} + H^T R^{-1} H\right)^{-1} \left(Y - H X^b\right)$	
Image Restoration	$x_k := (H^T H + \lambda \sigma^2 I_n)^{-1} (H^T y + \lambda \sigma^2 (v_{k-1} - u_{k-1}))$	
Rand. Matrix Inversion	$X_{k+1} := S(S^T A S)^{-1} S^T + (I_n - S(S^T A S)^{-1} S^T A) X_k (I_n - A S (S^T A S)^{-1} S^T)$	
Stochastic Newton	$B_k := \frac{k}{k-1} B_{k-1} (I_n - A^T W_k ((k-1) I_l + W_k^T A B_{k-1} A^T W_k)^{-1} W_k^T A B_{k-1})$	
Optimization	$x_f := W A^T (A W A^T)^{-1} (b - Ax)$; $x_o := W (A^T (A W A^T)^{-1} A x - c)$	
Tikhonov Regularization	$x := (A^T A + \Gamma^T \Gamma)^{-1} A^T b$	$A \in \mathbb{R}^{n \times m}$; $\Gamma \in \mathbb{R}^{m \times m}$; $b \in \mathbb{R}^{n \times 1}$
Gen. Tikhonov Reg.	$x := (A^T P A + Q)^{-1} (A^T P b + Q x_0)$	$P \in \mathbb{R}^{n \times n}$, SSPD; $Q \in \mathbb{R}^{m \times m}$, SSPD; $x_0 \in \mathbb{R}^{m \times 1}$
LMMSE estimator	$K_{t+1} := C_t A^T (A C_t A^T + C_z)^{-1}$; $x_{t+1} := x_t + K_{t+1} (y - A x_t)$; $C_{t+1} := (I - K_{t+1} A) C_t$	

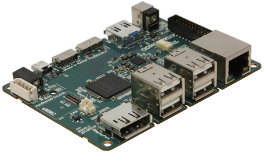
$$x := A(B^TB + A^TR^T\Lambda RA)^{-1}B^TBA^{-1}y$$

$$\begin{cases} C_\dagger := PCP^T + Q \\ K := C_\dagger H^T(HC_\dagger H^T)^{-1} \end{cases}$$

$$E := Q^{-1}U(I + U^TQ^{-1}U)^{-1}U^T \quad \ldots$$

?





MUL | ADD | MOV

MOVAPD

VFMADDPD ...

$$x := A(B^TB + A^TR^T\Lambda RA)^{-1}B^TBA^{-1}y$$

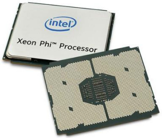
$$\begin{cases} C_\dagger := PCP^T + Q \\ K := C_\dagger H^T(HC_\dagger H^T)^{-1} \end{cases}$$

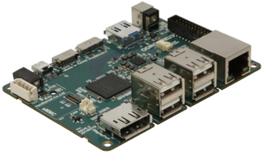
$$E := Q^{-1}U(I + U^TQ^{-1}U)^{-1}U^T \qquad \ldots$$

$\boxed{y := \alpha x + y}$ $\boxed{LU = A}$ $\cdots$ $\boxed{C := \alpha AB + \beta C}$

$\boxed{X := A^{-1}B}$ $\boxed{C := AB^T + BA^T + C}$ $\boxed{X := L^{-1}ML^{-T}}$ $\boxed{QR = A}$

BLAS ... **LAPACK** ...





MUL | ADD | MOV

MOVAPD

VFMADDPD ...

$x := A(B^TB + A^TR^T\Lambda RA)^{-1}B^TBA^{-1}y$

$$\begin{cases} C_\dagger := PCP^T + Q \\ K := C_\dagger H^T(HC_\dagger H^T)^{-1} \end{cases}$$

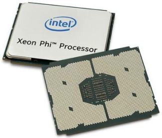
$E := Q^{-1}U(I + U^TQ^{-1}U)^{-1}U^T$ . . .

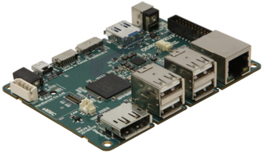
?

$\boxed{y := \alpha x + y}$ $\boxed{LU = A}$ $\cdots$ $\boxed{C := \alpha AB + \beta C}$

$\boxed{X := A^{-1}B}$ $\boxed{C := AB^T + BA^T + C}$ $\boxed{X := L^{-1}ML^{-T}}$ $\boxed{QR = A}$

BLAS . . . **LAPACK** . . .





$\boxed{\text{MUL}}$ $\boxed{\text{ADD}}$ $\boxed{\text{MOV}}$

$\boxed{\text{MOVAPD}}$

$\boxed{\text{VFMADDPD}}$ . . .

$$x := A(B^TB + A^TR^T\Lambda RA)^{-1}B^TBA^{-1}y$$

$$\begin{cases} C_\dagger := PCP^T + Q \\ K := C_\dagger H^T(HC_\dagger H^T)^{-1} \end{cases}$$

$$E := Q^{-1}U(I + U^TQ^{-1}U)^{-1}U^T$$

...

LINEAR ALGEBRA MAPPING PROBLEM (LAMP)

$y := \alpha x + y$

$:= \alpha AB + \beta C$

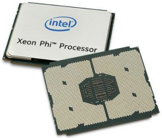
$X := A^{-1}B$

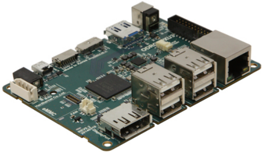
$C := A$

$^{-1}ML^{-T}$

$QR = A$

BLAS ... **LAPACK** ...





MUL ADD MOV

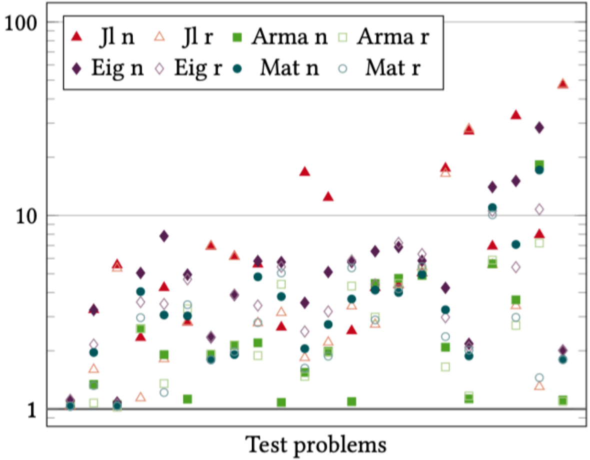
MOVAPD

VFMADDPD ...

LAMP: A problem often ignored

Linnea: A compiler for linear algebra – Henrik Barthels, Christos Psarras

Linnea's speedups



Jl: Julia, **Arma**: Armadillo, **Eig**: Eigen, **Mat**: Matlab. **n**/**r**: naive/recommended implementation

Tensor App #1 | Tensor App #2 | . . . | Tensor App #N

?

MUL | ADD | MOV

MOVAPD

VFMADDPD . . .

Tensor App #1 Tensor App #2 ... Tensor App #N

?

Transposition Contraction · · · Alternating LS

Khatri-Rao SpMTTKRP · · · TTV, TTM

HPTT **TCL** **...** **BLAS**

MUL ADD MOV

MOVAPD

VFMADDPD ...

nD case: Exemplary applications

Coupled-Cluster methods

$$
\begin{aligned}
\tau_{ij}^{ab} &= t_{ij}^{ab} + \frac{1}{2} P_b^a P_j^i t_i^a t_j^b, \\
\tilde{F}_e^m &= f_e^m + \sum_{fn} v_{ef}^{mn} t_n^f, \\
\tilde{F}_e^a &= (1-\delta_{ae}) f_e^a - \sum_m \tilde{F}_e^m t_m^a - \frac{1}{2} \sum_{mnf} v_{ef}^{mn} t_{mn}^{af} + \sum_{fn} v_{ef}^{an} t_n^f, \\
\tilde{F}_i^m &= (1-\delta_{mi}) f_i^m + \sum_e \tilde{F}_e^m t_i^e + \frac{1}{2} \sum_{nef} v_{ef}^{mn} t_{in}^{ef} + \sum_{fn} v_{if}^{mn} t_n^f,
\end{aligned}
$$

$$
\begin{aligned}
\tilde{W}_{ei}^{mn} &= v_{ei}^{mn} + \sum_f v_{ef}^{mn} t_i^f, \\
\tilde{W}_{ij}^{mn} &= v_{ij}^{mn} + P_j^i \sum_e v_{ie}^{mn} t_j^e + \frac{1}{2} \sum_{ef} v_{ef}^{mn} \tau_{ij}^{ef}, \\
\tilde{W}_{ie}^{am} &= v_{ie}^{am} - \sum_n \tilde{W}_{ei}^{mn} t_n^a + \sum_f v_{ef}^{ma} t_i^f + \frac{1}{2} \sum_{nf} v_{ef}^{mn} t_{in}^{af}, \\
\tilde{W}_{ij}^{am} &= v_{ij}^{am} + P_j^i \sum_e v_{ie}^{am} t_j^e + \frac{1}{2} \sum_{ef} v_{ef}^{am} \tau_{ij}^{ef}, \\
z_i^a &= f_i^a - \sum_m \tilde{F}_i^m t_m^a + \sum_e f_e^a t_i^e + \sum_{em} v_{ei}^{ma} t_m^e + \sum_{em} v_{im}^{ae} \tilde{F}_e^m + \frac{1}{2} \sum_{efm} \\
z_{ij}^{ab} &= v_{ij}^{ab} + P_j^i \sum_e v_{ie}^{ab} t_j^e + P_b^a P_j^i \sum_{me} \tilde{W}_{ie}^{am} t_{mj}^{eb} - P_b^a \sum_m \tilde{W}_{ij}^{am} t_m^b + P
\end{aligned}
$$

credits to D. Matthews, E. Solomonik, J. Stanton, and J. Gauss

nD case: Exemplary applications

Coupled-Cluster methods

$$
\begin{aligned}
\tau_{ij}^{ab} &= t_{ij}^{ab} + \frac{1}{2} P_b^a P_j^i t_i^a t_j^b, \\
\tilde{F}_e^m &= f_e^m + \sum_{fn} v_{ef}^{mn} t_n^f, \\
\tilde{F}_e^a &= (1-\delta_{ae}) f_e^a - \sum_m \tilde{F}_e^m t_m^a - \frac{1}{2} \sum_{mnf} v_{ef}^{mn} t_{mn}^{af} + \sum_{fn} v_{ef}^{an} t_n^f, \\
\tilde{F}_i^m &= (1-\delta_{mi}) f_i^m + \sum_e \tilde{F}_e^m t_i^e + \frac{1}{2} \sum_{nef} v_{ef}^{mn} t_{in}^{ef} + \sum_{fn} v_{if}^{mn} t_n^f, \\
\tilde{W}_{ei}^{mn} &= v_{ei}^{mn} + \sum_f v_{ef}^{mn} t_i^f, \\
\tilde{W}_{ij}^{mn} &= v_{ij}^{mn} + P_j^i \sum_e v_{ie}^{mn} t_j^e + \frac{1}{2} \sum_{ef} v_{ef}^{mn} \tau_{ij}^{ef}, \\
\tilde{W}_{ie}^{am} &= v_{ie}^{am} - \sum_n \tilde{W}_{ei}^{mn} t_n^a + \sum_f v_{ef}^{ma} t_i^f + \frac{1}{2} \sum_{nf} v_{ef}^{mn} t_{in}^{af}, \\
\tilde{W}_{ij}^{am} &= v_{ij}^{am} + P_j^i \sum_e v_{ie}^{am} t_j^e + \frac{1}{2} \sum_{ef} v_{ef}^{am} \tau_{ij}^{ef}, \\
z_i^a &= f_i^a - \sum_m \tilde{F}_i^m t_m^a + \sum_e f_e^a t_i^e + \sum_{em} v_{ei}^{ma} t_m^e + \sum_{em} v_{im}^{ae} \tilde{F}_e^m + \frac{1}{2} \sum_{efm} \\
z_{ij}^{ab} &= v_{ij}^{ab} + P_j^i \sum_e v_{ie}^{ab} t_j^e + P_b^a P_j^i \sum_{me} \tilde{W}_{ie}^{am} t_{mj}^{eb} - P_b^a \sum_m \tilde{W}_{ij}^{am} t_m^b + P
\end{aligned}
$$

credits to D. Matthews, E. Solomonik, J. Stanton, and J. Gauss

Finite Element 3D diffusion operator

```
TE.BeginMultiKernelLaunch();
TE("T2_e_i1_i2_k3 = B_k3_i3 X_e_i1_i2_i3", T2, B, X);
TE("T1_e_i1_k2_k3 = B_k2_i2 T2_e_i1_i2_k3", T1, B, T2);
TE("U1_e_k1_k2_k3 = G_k1_i1 T1_e_i1_k2_k3", U1, G, T1);
TE("T1_e_i1_k2_k3 = G_k2_i2 T2_e_i1_i2_k3", T1, G, T2);
TE("U2_e_k1_k2_k3 = B_k1_i1 T1_e_i1_k2_k3", U2, B, T1);
TE("T2_e_i1_i2_k3 = G_k3_i3 X_e_i1_i2_i3", T2, G, X);
TE("T1_e_i1_k2_k3 = B_k2_i2 T2_e_i1_i2_k3", T1, B, T2);
TE("U3_e_k1_k2_k3 = B_k1_i1 T1_e_i1_k2_k3", U3, B, T1);
TE("Z_m_e_k1_k2_k3 = U_n_e_k1_k2_k3 D_e_m_n_k1_k2_k3", Z, U,
TE("T1_e_i3_k1_k2 = B_k3_i3 Z1_e_k1_k2_k3", T1, B, Z1);
TE("T2_e_i2_i3_k1 = B_k2_i2 T1_e_i3_k1_k2", T2, B, T1);
TE("Y_e_i1_i2_i3 = G_k1_i1 T2_e_i2_i3_k1", Y, G, T2);
TE("T1_e_i3_k1_k2 = B_k3_i3 Z2_e_k1_k2_k3", T1, B, Z2);
TE("T2_e_i2_i3_k1 = G_k2_i2 T1_e_i3_k1_k2", T2, G, T1);
TE("Y_e_i1_i2_i3 += B_k1_i1 T2_e_i2_i3_k1", Y, B, T2);
TE("T1_e_i3_k1_k2 = G_k3_i3 Z3_e_k1_k2_k3", T1, G, Z3);
TE("T2_e_i2_i3_k1 = B_k2_i2 T1_e_i3_k1_k2", T2, B, T1);
TE("Y_e_i1_i2_i3 += B_k1_i1 T2_e_i2_i3_k1", Y, B, T2);
TE.EndMultiKernelLaunch();
```

credits to A. Fisher – https://github.com/LLNL/acrotensor

Awareness

- **Performance**:
  Speedups in building blocks do not always translate to speedups in applications

Awareness

- **Performance**:
  Speedups in building blocks do not always translate to speedups in applications

- **Research Problem**: Mapping onto building blocks
  - Solved by hand $\rightarrow$ loss of productivity (possibly loss of efficiency too)
  - Solved automatically $\rightarrow$ loss of efficiency

  Beware: It's challenging even for "simple" matrix computations!

Part II

The computational scientists' perspective

"The fastest FLOPS are those that are not executed."
– Lars

Chromatography-MS

PARAFAC | Tucker | PARAFAC2

Transposition | Contraction | · · · | Alternating LS

Khatri-Rao | SpMTTKRP | · · · | TTV, TTM

HPTT | **TCL** | **. . .** | **BLAS**

MUL | ADD | MOV

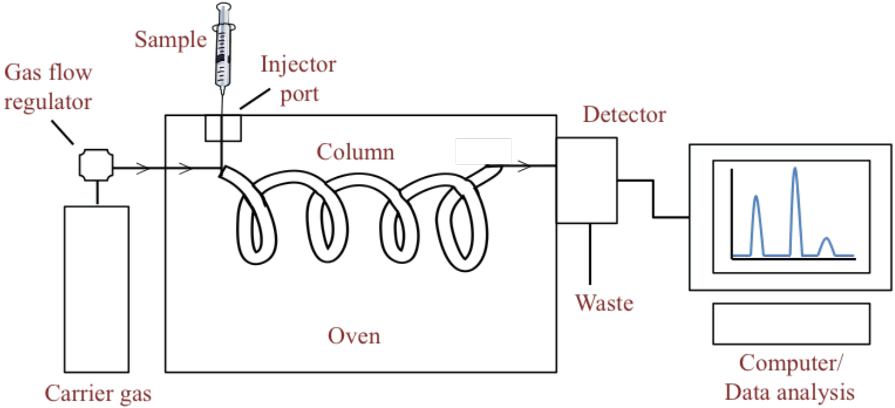
MOVAPD

VFMADDPD . . .

Example application: Untargeted chemical profiling

Chromatography with mass spectrometry detection

- Problem: Identify components in a sample

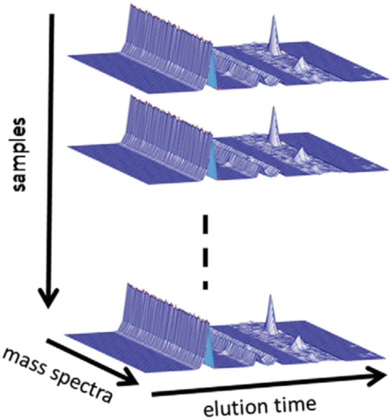


Jessica Torres – Bitesize Bio

Example application: Untargeted chemical profiling

Chromatography with mass spectrometry detection

- 3-way data: Mass-spectrum $\times$ elution time $\times$ sample

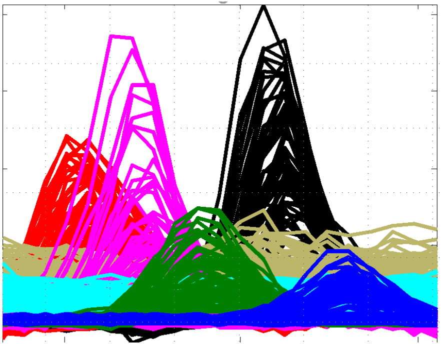


Rasmus Bro

Example application: Untargeted chemical profiling

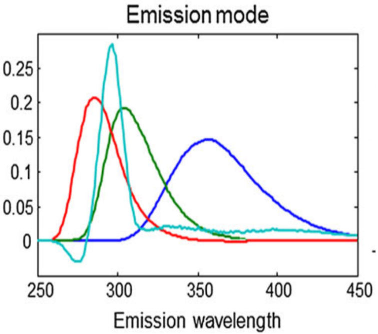
Chromatography with mass spectrometry detection

- 3-way tensor $\rightarrow$ Individual components



Rasmus Bro

$\rightarrow$



Workflow

- **1.** (Pre-processing: alignment)
- **2.** Choose time intervals across samples
- **3.** (Data preparation: remove background)

Workflow

- **1.** (Pre-processing: alignment)
- **2.** Choose time intervals across samples
- **3.** (Data preparation: remove background)
- **4. Fit model**: 1–15 components; if needed: non-negativity constraints

  PARAFAC — PARAFAC2 — TUCKER

Workflow

- **1.** (Pre-processing: alignment)
- **2.** Choose time intervals across samples
- **3.** (Data preparation: remove background)
- **4.** **Fit model**: 1–15 components; if needed: non-negativity constraints
  PARAFAC — PARAFAC2 — TUCKER
- **5.** Determine whether or not one of the models is “right”

Workflow

- **1.** (Pre-processing: alignment)
- **2.** Choose time intervals across samples
- **3.** (Data preparation: remove background)
- **4. Fit model**: 1–15 components;   if needed: non-negativity constraints
  PARAFAC — PARAFAC2 — TUCKER
- **5.** Determine whether or not one of the models is "right"
  - : Determine which of the components represent chemical information
  - : Start over;   add/change constraints, change model

Awareness & Conclusions

- **Performance**:
  Speedups in building blocks do not always translate to speedups in applications

- **Research Problem**: Mapping onto building blocks
  - Solved by hand $\rightarrow$ loss of productivity (possibly loss of efficiency too)
  - Solved automatically $\rightarrow$ loss of efficiency

Beware: It's challenging even for "simple" matrix computations!

Awareness & Conclusions

- **Performance**:
  Speedups in building blocks do not always translate to speedups in applications

- **Research Problem**: Mapping onto building blocks
  - Solved by hand $\rightarrow$ loss of productivity (possibly loss of efficiency too)
  - Solved automatically $\rightarrow$ loss of efficiency

  Beware: It's challenging even for "simple" matrix computations!

- **Application as a whole**
  - Massive redundancy! Careful with black-box libraries
  - Man-in-the-middle workflow. Manual "check & decide"

Awareness & Conclusions

- **Performance**:
  Speedups in building blocks do not always translate to speedups in applications

- **Research Problem**: Mapping onto building blocks
  - Solved by hand $\rightarrow$ loss of productivity (possibly loss of efficiency too)
  - Solved automatically $\rightarrow$ loss of efficiency

  Beware: It's challenging even for "simple" matrix computations!

- **Application as a whole**
  - Massive redundancy!   Careful with black-box libraries
  - Man-in-the-middle workflow. Manual "check & decide"

- **Gap**: Domain scientists' needs $\leftrightarrow$ Computer scientists' artifacts

Awareness & Conclusions

- **Performance**:
  Speedups in building blocks do not always translate to speedups in applications

- **Research Problem**: Mapping onto building blocks
  - Solved by hand $\rightarrow$ loss of productivity (possibly loss of efficiency too)
  - Solved automatically $\rightarrow$ loss of efficiency

  Beware: It's challenging even for "simple" matrix computations!

- **Application as a whole**
  - Massive redundancy! Careful with black-box libraries
  - Man-in-the-middle workflow. Manual "check & decide"

- **Gap**: Domain scientists' needs $\leftrightarrow$ Computer scientists' artifacts

- **Problem**: How to maximize scientific output?
  - Speedups in algorithms and building blocks
  - Efficient mapping
  - Reduction in computation in the application

Awareness & Conclusions

- **Performance**:
  Speedups in building blocks do not always translate to speedups in applications
- **Research Problem**: Mapping onto building blocks
  - Solved by hand $\rightarrow$ loss of productivity (possibly loss of efficiency too)
  - Solved automatically $\rightarrow$ loss of efficiency

  Beware: It's challenging even for "simple" matrix computations!
- **Application as a whole**
  - Massive redundancy! Careful with black-box libraries
  - Man-in-the-middle workflow. Manual "check & decide"
- **Gap**: Domain scientists' needs $\leftrightarrow$ Computer scientists' artifacts
- **Problem**: How to maximize scientific output?
  - Speedups in algorithms and building blocks
  - Efficient mapping
  - Reduction in computation in the application

Thank you

Questions?