Computing with Molecules at Dresden University of Technology

Thomas Hinze

Dresden University of Technology
Computer Science Department
Theoretical Computer Science

email: hinze@tcs.inf.tu-dresden.de
www.molecular-computing.de

Outline

Who we are

- part of **Biosaxony Network** in Dresden
- research group of 10 persons
- group leader: Dr. Monika Sturm

What we do

→ **aims and visions**

→ **projects and results**

→ **scientific collaborations**

→ **teaching activities**

approx. 80 companies
foundations
external partners

Max Planck Institute of Molecular Cell Biology and Genetics

biosaxony
www.biosaxony.com

Biotec
BioInnovation Centre

TECHNISCHE UNIVERSITÄT DRESDEN

Genetics
Nanotechnology
Bioinformatics
Artificial Intelligence
Theoretical Computer Science

Research Group Computing with Molecules

Aims and Visions

- modelling and simulation of molecular biological processes
- investigation, description, and optimization of models for computation
- biomolecule based algorithmic design

$\longrightarrow$ bridging gap between **theoretical models** and **lab implementations**

Projects and Results

- wetware solution of the knapsack problem
- simulation system for phenomena undergoing side effects (Sisyphus)
- simple artificial chemistry experimental system (Saces)
- draft of universal programmable DNA based computer (TT6)
- library of data parallel algorithms based on Chomsky grammars
- genetic computing via microbial circuits *in vivo*

Knapsack Problem

NP-complete, exponential need of resources, combinatorics

Problem Definition

There are n natural numbers $a_1, \ldots, a_n$ and reference number $b \in \mathbb{N}$

Is there a subset $I \subseteq \{1, \ldots, n\}$ with $\sum_{i \in I} a_i = b$?

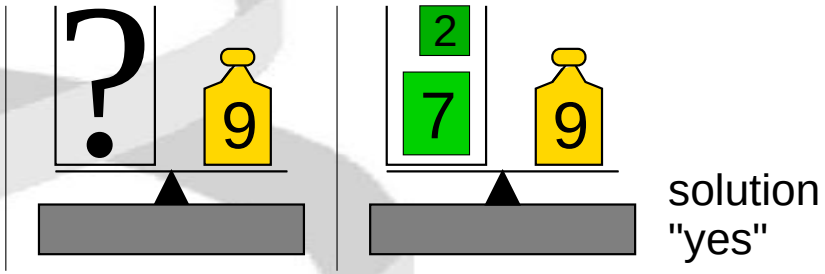
Explanation

$a_1, \ldots, a_n$: **weights** of **objects** $1, \ldots, n$.

Is there a possibility to pack a selection of these objects into the knapsack and to meet the overall weight b exactly?

Example

$a_1 = 5$ 5 object 1
$a_2 = 7$ 7 object 2
$a_3 = 2$ 2 object 3
$b = 9$



solution "yes"

Wetware Solution of the Knapsack Problem

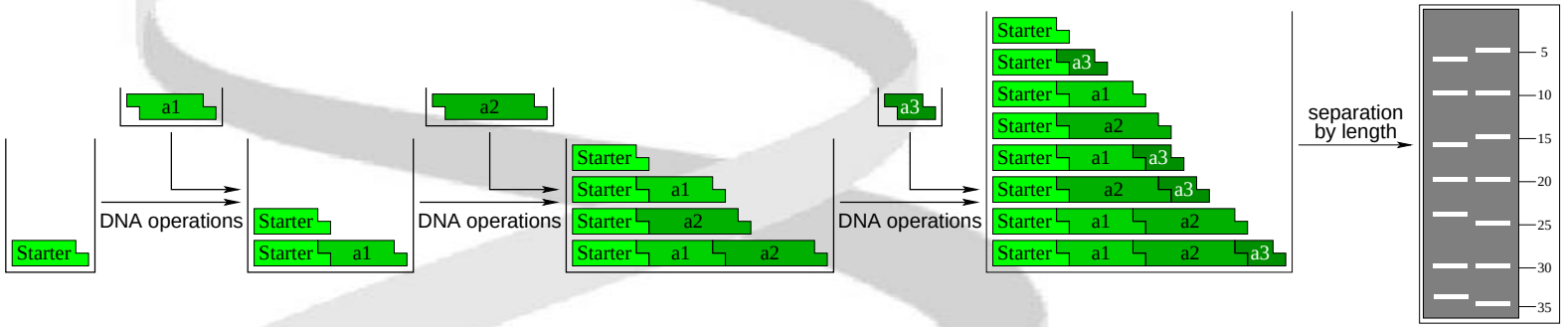
Brute Force Approach

- encode $a_1, \ldots, a_n$ into DNA double strands by length $(c \cdot a_i)$
- generate all solution candidates by a controlled **split-and-combine strategy**
- separate the final DNA pool by length
- detect DNA at *Starter length* $+ c \cdot b$ and answer yes

$\longrightarrow$ problem instance of size $n = 3$ implemented *in vitro*

$\longrightarrow$ exponential need of resources moves from time to space

$\longrightarrow$ limited scalability of the algorithm because of **side effects** and amount of DNA



Modelling DNA Molecules

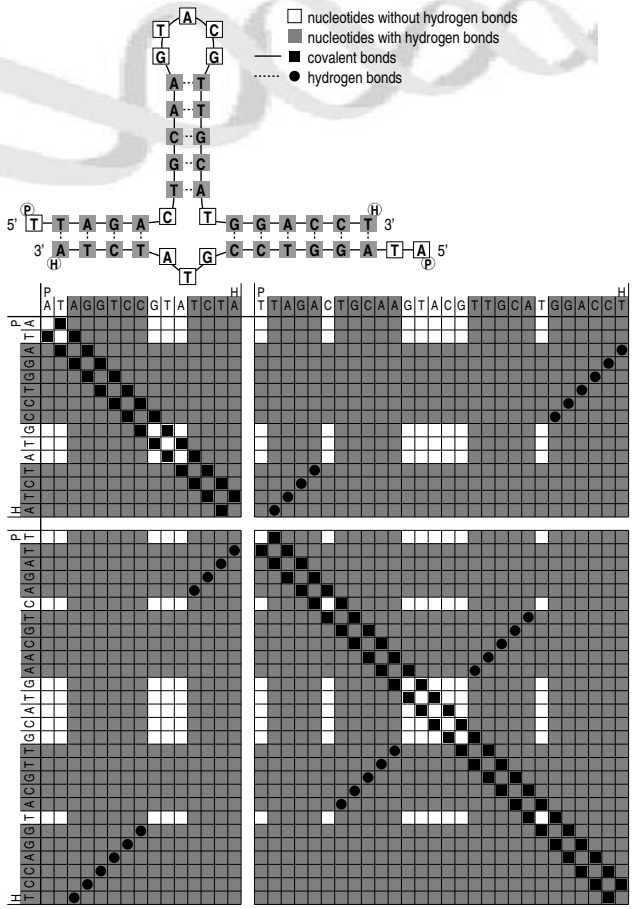
Primary Structure

- word over $\{A, C, G, T\}$

Secondary Structure

- set of nucleotides $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$

  $\sigma_i \in \{A, C, G, T\}$
- relation of covalent bonds $C \subseteq \Sigma \times \Sigma$
- relation of hydrogen bonds $H \subseteq \Sigma \times \Sigma$
- inductive definition of C and H



$\longrightarrow$ mathematical model to describe term (graph) **rewriting rules**

$\longrightarrow$ **self assembly** towards linear and nonlinear polymers

$\longrightarrow$ allows formalization of DNA operations on a moderate abstraction level

DNA Operations (I)

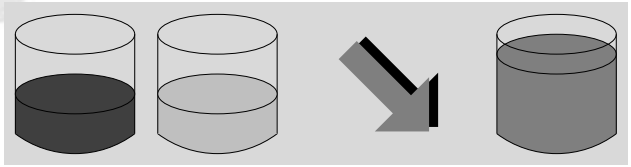
Generating DNA Strands

- **synthesis** (oligonucleotides)

- **isolation** . . . . . from organisms

Merging and Aliquoting

- **union/split**



Modifying Hydrogen Bonds

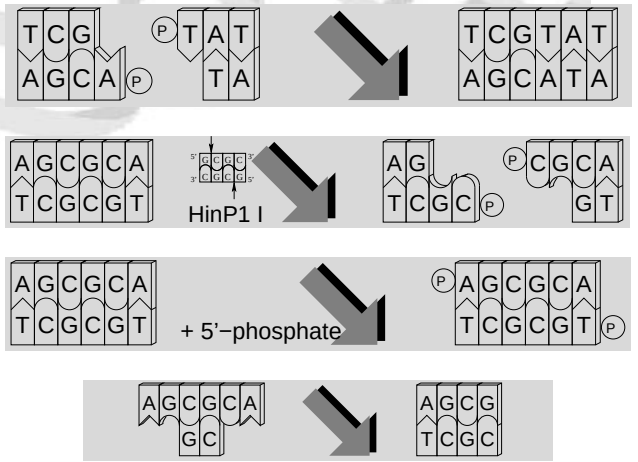
- **annealing** (hybridization)

- **melting** (denaturation)

DNA Operations (II)

Enzymatic Reactions

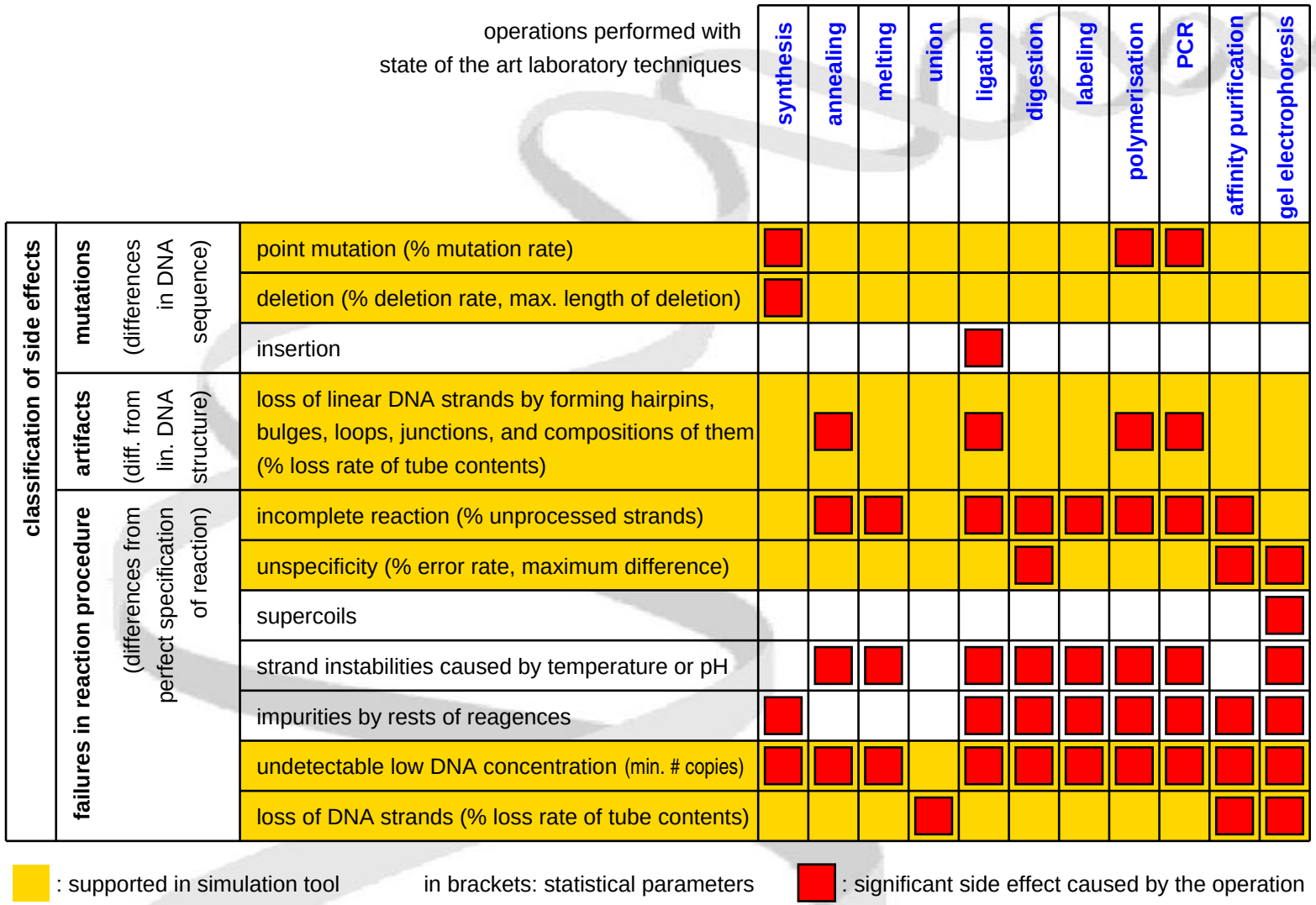
- **ligation** (concatenation) . . . . . .
- **digestion** (cleavage) . . . . . .
- **labeling** (strand end modification)
- **polymerisation** (blunting) . . . . . .
- **PCR** (polymerase chain reaction) . . . . . duplicate strands



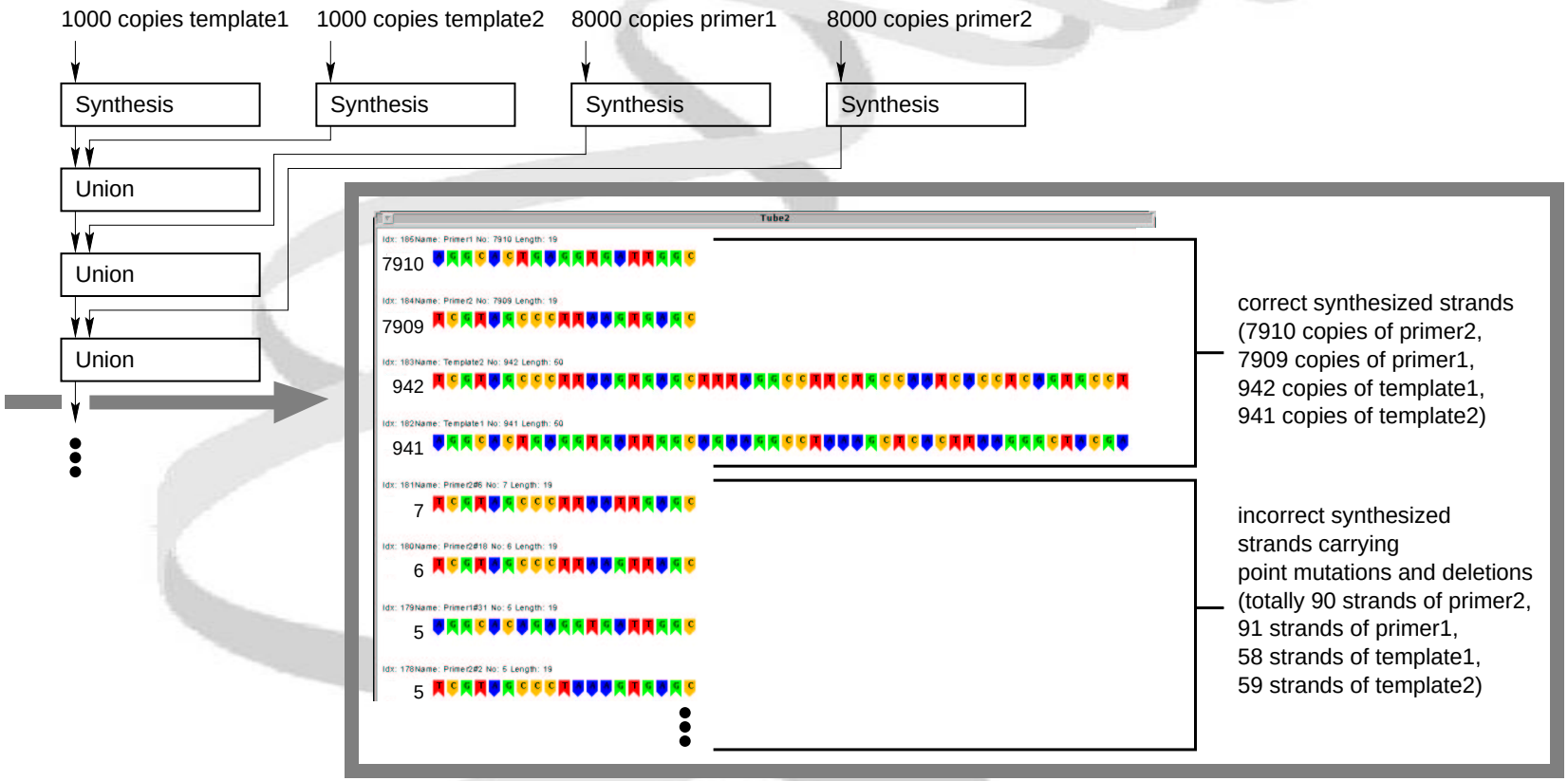
Separating and Analyzing of DNA Pools

- **affinity purification** (sep. by biotin)
- **gel electrophoresis** (sep. by length) . . . sort and detect strands
- **sequencing** (read out strand) . . . . .

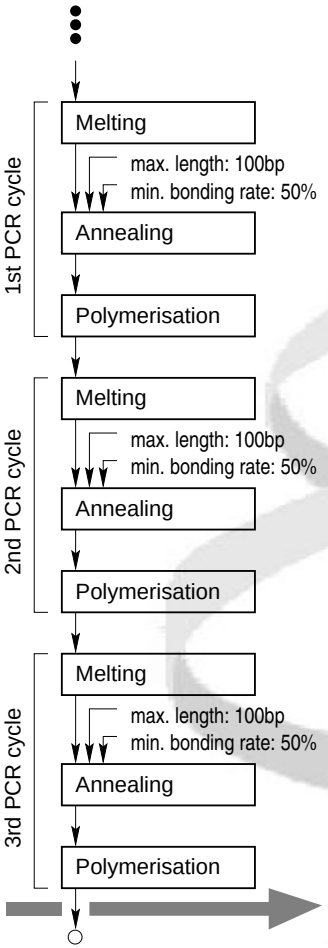
Side Effects of DNA Operations

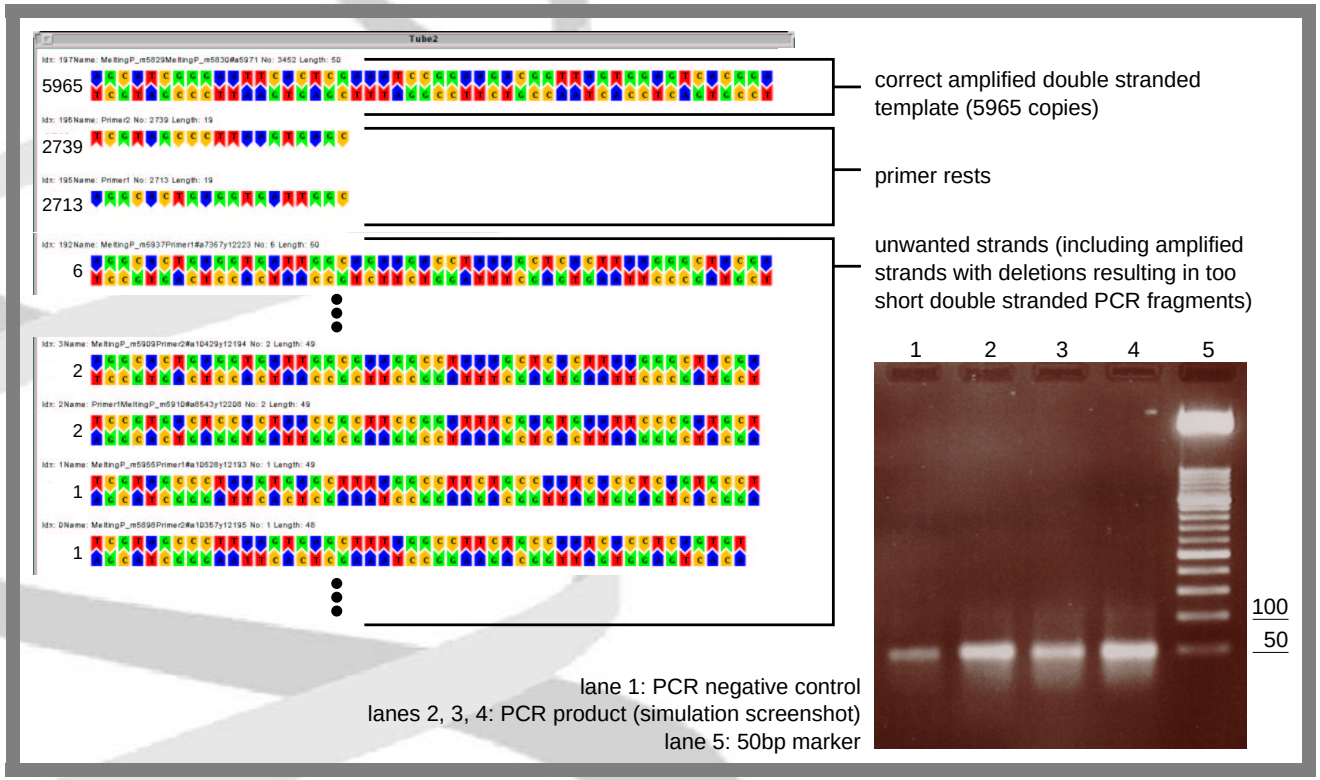


A PCR Example (I)

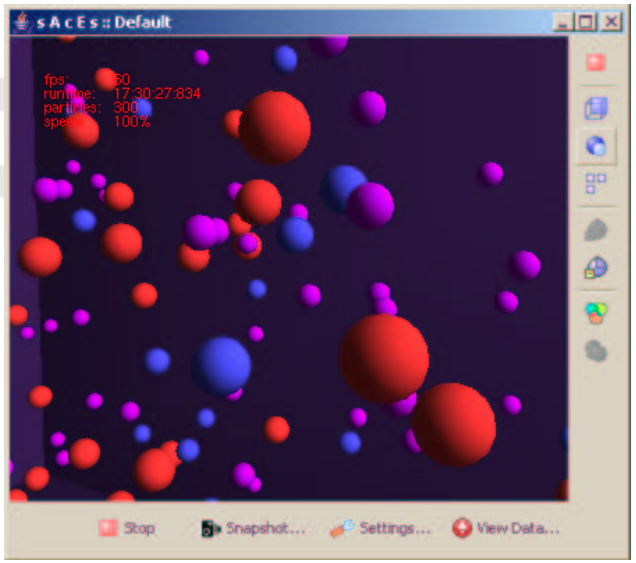


A PCR Example (II)





Simple Artificial Chemistry Experimental System



Idea

- behaviour of ideal gases can compute

Principle

- **set of particles**

  molecules with parameters $(m, E_\mathrm{b}, \#, \ldots)$
- randomly placed/speeded due to

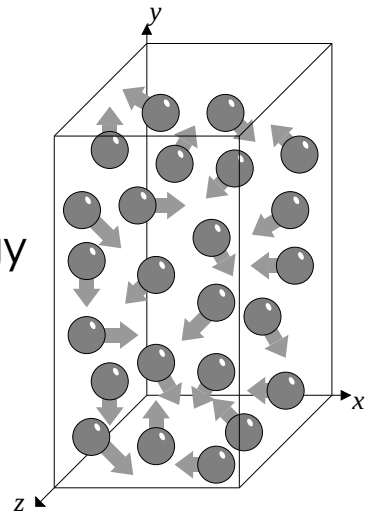
  Maxwell-Boltzmann distribution
- **set of reactions and global parameters**

  $A + B \longrightarrow C + D;\ E_\mathrm{activation}$ (B or D can be empty)
- **algorithm:** Brownian movement (random walk) with

  elastic/inelastic collisions, retains momentum and energy
- **analysis:** animation, log, report, histogram

$\longrightarrow$ suitable for solution of NP-complete problems

$\longrightarrow$ info and software download: `http://saces.yce.ch`



Splicing Operation

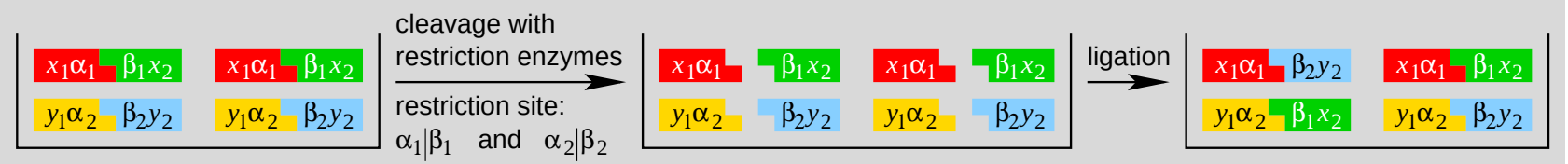
- DNA recombination by cleavage and ligation can compute
- enables nondeterministic computation based on term rewriting
- main operation of programmable splicing systems (EH systems)

Definition (T. Head, 1987)

Let V an alphabet and $\$, \# \notin V$. A **splicing rule** over V is a word $r = \alpha_1 \# \beta_1 \$ \alpha_2 \# \beta_2$ with $\alpha_i, \beta_i \in V^*, i \in \{1, 2\}$.

We define for each $r \in R$ and for words $x, y, w, z \in V^*$:

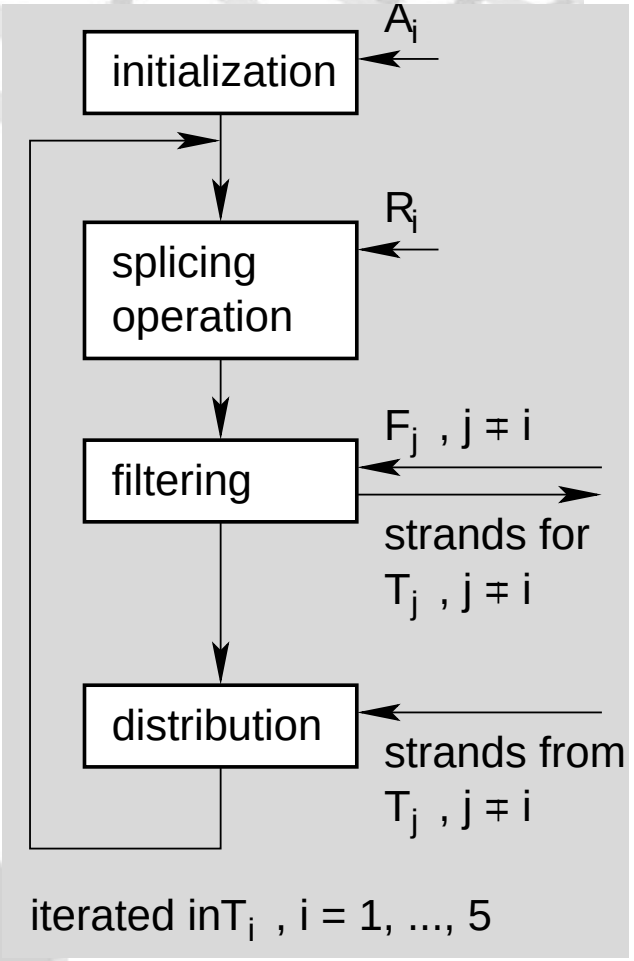
$$(x, y) \rightarrow_r (z, w) \quad \text{iff} \quad x = x_1\alpha_1\beta_1x_2, \quad y = y_1\alpha_2\beta_2y_2,$$
$$z = x_1\alpha_1\beta_2y_2, \quad w = y_1\alpha_2\beta_1x_2.$$



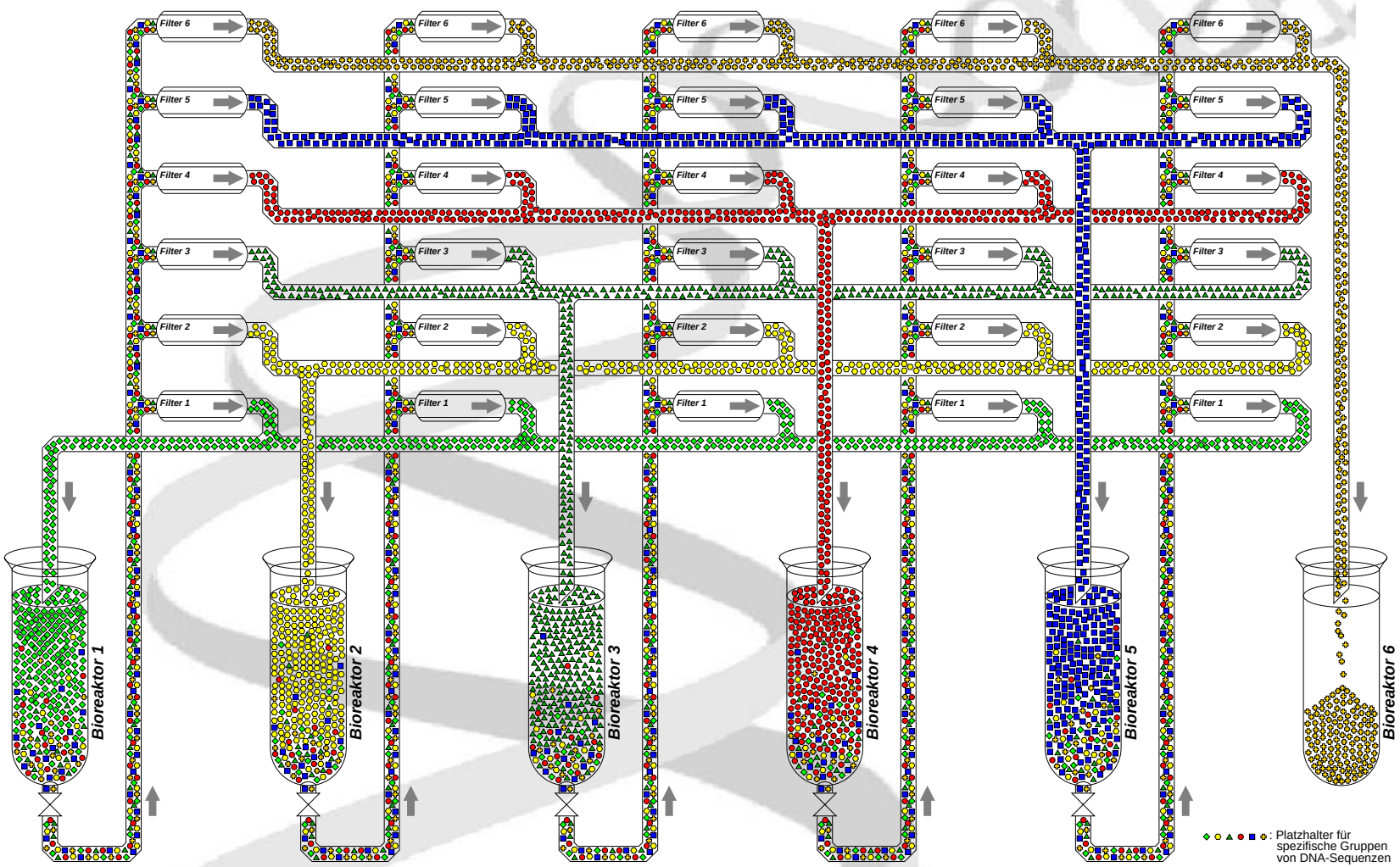
Universal Distributed Splicing System (TT6)

Properties

- computational complete
- finite system components
- static structure
- programmable by Chomsky grammars
- massive data parallel processing
- use of linear DNA strands
- computational results provided in separate test tube T_6
- system can be described using well-known DNA operations
- minimization of distributed strands
- PCR based filtering method



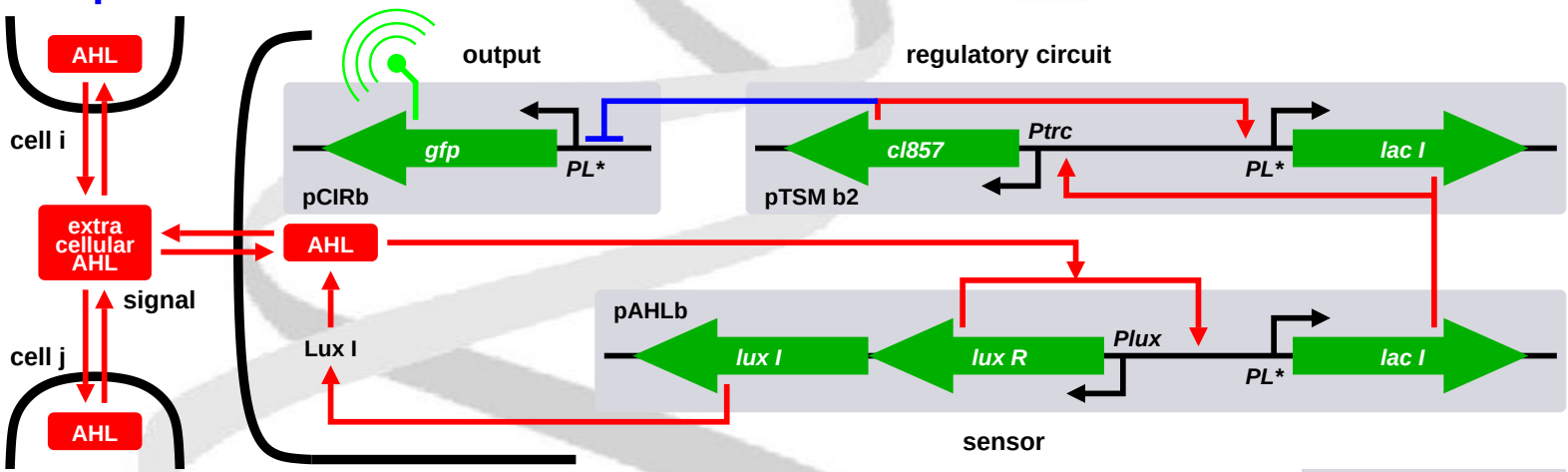
Principle of Univers. Distributed Splicing System TT6



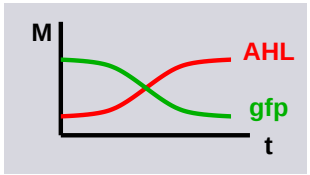
Genetic Computing via Microbial Circuits *in vivo*

- construct connection free logic gates (NOT, NAND, FF, . . .) using **cell signalling pathways** and **controlled gene expression**

Example NOT Gate



- organism *vibrio fischeri*, signalling network of **promoter (|)** and **repressor (⊢)** proteins
- **input:** signalling molecules AHL (N-acryl homoserine lactones)
- **output:** gfp (green fluorescence protein)



AHL	gfp
0	1
1	0

Commitment outside Biosaxony

Scientific Collaborations

- European Molecular Computing Consortium (EMCC)
- Leiden Institute of Advanced Computer Science (LIACS)
- Vienna University of Technology, Theory and Logic Group
- Berne University of Applied Sciences, School of Engineering
- Fraunhofer Institute for Integrated Circuits (IIS)
- Philipps University Marburg, Dpt. Clinical Cytobiology and Cytopathology

Teaching Activities

Th. Hinze, M. Sturm.

Rechnen mit DNA – Eine Einführung in Theorie und Praxis.

Oldenbourg Wissenschaftsverlag München, 2004

The German-speaking book provides a comprehensive and systematic introduction into the interdisciplinary field of DNA computing including its mathematical and molecular biological background. The transfer of basic knowledge about DNA computing is completed by the detailled introduction of models, methods, and techniques that prepare implementations in vitro. Particularly process simulations on a submolecular level are considered.