The Algebra of Ciliates

Hendrik Jan Hoogeboom
Leiden

with

Robert Brijder
Hasselt
Computation in Living Cells
– Gene Assembly in Ciliates
A.Ehrenfeucht, T.Harju, I.Petre, D.M.Prescott, G.Rozenberg
Springer, 2004

the overview


Unlike most other eukaryotes, ciliates have two different sorts of nuclei: a small, diploid micronucleus (reproduction), and a large, polyploid macronucleus (general cell regulation). The latter is generated from the micronucleus by amplification of the genome and heavy editing.

http://en.wikipedia.org/wiki/Ciliate
from micro to macro

http://oxytricha.princeton.edu/cgi-bin/get_MDS_IES_Info.cgi?num=38

micronucleus

DNA: 2374 bp

macronucleus

DNA: 1604 bp

gene

pointers

pointers – overlapping segments (for glueing)

e.g., pointer 5 of actin gene: 13 bp
rc₄  recombination on pointer 4  ‘generic’

before

after  ‘math view’

after  ‘ciliate view’
1. Loop recombination

2. Hairpin recombination

3. Double-loop recombination

Pointers between two copies of 4
quest for the “right” model

- strings + graphs
- matrices
- set systems
abstraction: pointers

3 4
2
3 2
4

3 4 2 3 2 4

3423 24

‘legal’ string

realistic strings vs. generalizations

... 4774 ...
\[ rc_p( u_1 p p u_2 ) = u_1 p p u_2 \]

\[ rc_p( u_1 p u_2 \bar{p} u_3 ) = u_1 \bar{p} u_2 \bar{p} u_3 \]

\[ rc_{p,q}( u_1 p u_2 q u_3 p u_4 q u_5 ) = u_1 p u_4 q u_3 p u_2 q u_5 \]
circle & interval / overlap graph
(for signed graphs instead of looped graphs)
local complementation

\[ rC_p \]

N(p) \n N(q)

\[ N(p) \cap N(q) \]

N(p) \ \ N(q)

edge complementation

\[ rC_{p,q} \]

N(p) \cap N(q)

unlooped edge pq
$rc_{3,4}$ on edge 3,4
questions:

how do $rc_{p,q}$ and $rc_{p',q'}$, or $rc_{p',q'}$, interact?

is the result of reductions dependent on (order) operations chosen?
quest for the “right” model

- strings
- graphs ⇔ matrices
- set systems
graphs and matrices
reconsider local/edge complementation

\[
\begin{array}{cccccccc}
2 & 3 & 4 & 5 & 6 & 7 \\
2 & 1 & 0 & 1 & 1 & 0 & 1 \\
3 & 0 & 0 & 1 & 1 & 1 & 0 \\
4 & 1 & 1 & 0 & 1 & 1 & 0 \\
5 & 1 & 1 & 1 & 1 & 1 & 0 \\
6 & 0 & 1 & 1 & 1 & 0 & 1 \\
7 & 1 & 0 & 0 & 0 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
2 & 3 & 4 & 5 & 6 & 7 \\
2 & 1 & 0 & 1 & 1 & 0 & 1 \\
3 & 0 & 0 & 1 & 1 & 1 & 0 \\
4 & 1 & 1 & 1 & 0 & 1 & 1 \\
5 & 1 & 1 & 0 & 0 & 1 & 1 \\
6 & 0 & 1 & 1 & 1 & 0 & 1 \\
7 & 1 & 0 & 1 & 1 & 1 & 0 \\
\end{array}
\]
reconsider edge complementation
**principal pivot transform**

A * X is defined iff A[X] is invertible/nonsingular/det ≠ 0

\[
A = \begin{pmatrix} x & P & Q \\ P & R & S \end{pmatrix}
\]

\[
A * X = \begin{pmatrix} P^{-1} & -P^{-1} Q \\ R P^{-1} & S - R P^{-1} Q \end{pmatrix}
\]

**partial inverse**

\[
A \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \text{ iff } A^* X \begin{pmatrix} x_2 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_2 \end{pmatrix}
\]

X pointers

other

case: unlooped edge

principal pivot transform

A * X is defined iff A[X] is invertible/nonsingular/det ≠ 0

\[
A = X \begin{pmatrix} P & Q \\ R & S \end{pmatrix} \quad \text{and} \quad A * X = \begin{pmatrix} P^{-1} & -P^{-1} Q \\ R P^{-1} & S - R P^{-1} Q \end{pmatrix}
\]

principal pivot transform

\[
A \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \text{ iff } A^*X \begin{pmatrix} x_2 \\ y_1 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_2 \end{pmatrix}
\]

using partial inversion

\[
(A^* X) * Y = A^* (X \oplus Y)
\]

(xor)

when defined

\[
A * \{p_1, p_2\} \ldots * p_n = A^* V = A^{-1}
\]

(all pointers)

any sequence involving all pointers

this shows
- how the rc_p and rc_{p,q} interact
- result does not depend on order of operations
what is happening?

\[
\begin{align*}
3423 \underline{24} & \quad \quad 3 \underline{24}234 \\
\begin{array}{ccc}
2 & 3 & 4 \\
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 0
\end{array} & \quad \quad \\
\begin{array}{ccc}
2 & 3 & 4 \\
1 & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 0
\end{array}
\end{align*}
\]

multiply (over the binary numbers)

\[
\begin{align*}
\begin{array}{ccc}
1 & 1 & 0 \\
1 & 1 & 1 \\
0 & 1 & 0
\end{array} & \quad \quad \\
\begin{array}{ccc}
1 & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 0
\end{array} & \quad \quad = & \quad \quad \\
\begin{array}{ccc}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{array}
\end{align*}
\]

+ xor \( \oplus \) \( 1+1=0 \)

* and \( \land \)

sorting DNA = computing the inverse

ppt is (partial) inverse
A * X is defined iff A[X] is invertible

rc_2

rc_3,4

rc_4 undefined
by careful modeling we find that gene assembly is actually principal pivot transform (ppt)

we can use results about ppt to know more about gene assembly
  • independent order operations
  • interaction operations
quest for the “right” model

- strings
- graphs
- matrices
- set systems

the most elegant model was hidden
A * X is defined iff A[X] is invertible

A[\{3,4\}]

V = \{2,3,4\}
D = \{\emptyset, \{2\}, \{3\}, \{2,4\}, \{3,4\}, \{2,3,4\}\}

*\{3,4\} is not rc_{3,4}
how simple can it get ...

graphs $\subseteq$ set systems (strict)

342324
2 \[2, 3, 4\]
3 \[1, 1, 1\]
4 \[0, 1, 0\]

$V = \{2, 3, 4\}$
$D = \{ \emptyset, \{2\}, \{3\}, \{2, 4\}, \{3, 4\}, \{2, 3, 4\} \}$

rc3
⊕3 xor 3

324324
2 \[0, 1, 1\]
3 \[1, 1, 1\]
4 \[1, 1, 1\]

$V = \{2, 3, 4\}$
$D' = \{ \{3\}, \{2, 3\}, \emptyset, \{2, 3, 4\}, \{4\}, \{2, 4\} \}$

applicability (!) XOR \{4\} is defined, while rc\{4\} is not, nb. \{4\} not in D
algebra of set systems

{ \emptyset, \{q\}, \{p,q\}, \{p,r\}, \{p,q,r\} }
algebra of set systems

\{ \emptyset, \{q\}, \{p,q\}, \{p,r\}, \{p,q,r\} \}
**p and +p generate group $S_3$**
edge complement vs. local complement

ignoring loops
**edge complement vs. local complement**

\[+3 \ast 3 \ast 4 + 3 \ast 3 + 3 = \ast 3 \ast 3 + 3 \ast 4 = \ast 3 \ast 3 \ast 3 + 3 \ast 3 \ast 4 = +3 + 3 \ast 3 \ast 4 = \ast 3 \ast 4 = \ast \{3, 4\}\]

**basic algebra** \(S_3\)

\[\ast 3 \ast 4 = \ast 4 \ast 3\]

\[\ast 3 \ast 3 = \text{id} = +3 + 3\]

\[+3 \ast 3 + 3 = \ast 3 + 3 \ast 3\]
by careful modeling we find that gene assembly is \textit{actually} principal pivot transform (ppt) and XOR

we can use results about ppt (on matrices) and XOR (on set systems) to know more about gene assembly

but also inspiration the other way around ...

however ...  

- parallellism  
- ‘simple’ operations

\textit{thnx!}
recombination is ppt


*p +p algebra of operations

general
