

# Partitioning into Colorful Components by Minimum Edge Deletions

Sharon Bruckner<sup>1</sup>   Falk Hüffner<sup>2</sup>  
Christian Komusiewicz<sup>2</sup>   Rolf Niedermeier<sup>2</sup>   Sven Thiel<sup>3</sup>  
Johannes Uhlmann<sup>2</sup>

<sup>1</sup>Institut für Mathematik, Freie Universität Berlin

<sup>2</sup>Institut für Softwaretechnik und Theoretische Informatik, TU Berlin

<sup>3</sup>Institut für Informatik, Friedrich-Schiller-Universität Jena

3 July 2012

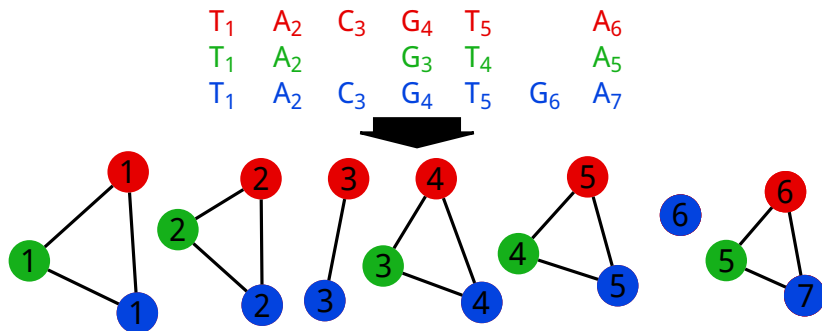
# Multiple Sequence Alignment

T <sub>1</sub>	A <sub>2</sub>	C <sub>3</sub>	G <sub>4</sub>	T <sub>5</sub>	A <sub>6</sub>	
T <sub>1</sub>	A <sub>2</sub>	G <sub>3</sub>	T <sub>4</sub>	A <sub>5</sub>		
T <sub>1</sub>	A <sub>2</sub>	C <sub>3</sub>	G <sub>4</sub>	T <sub>5</sub>	G <sub>6</sub>	A <sub>7</sub>

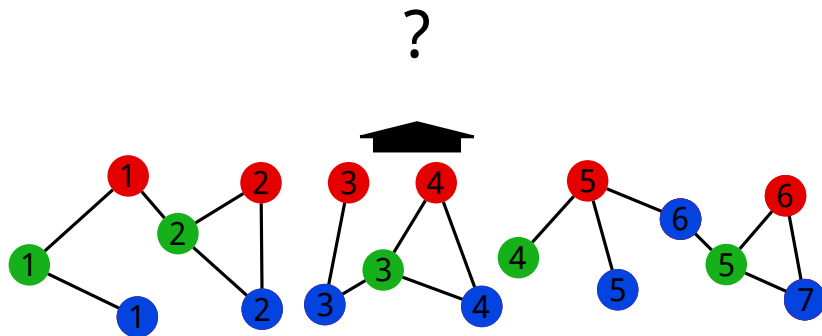
# Multiple Sequence Alignment

T <sub>1</sub>	A <sub>2</sub>	C <sub>3</sub>	G <sub>4</sub>	T <sub>5</sub>		A <sub>6</sub>
T <sub>1</sub>	A <sub>2</sub>		G <sub>3</sub>	T <sub>4</sub>		A <sub>5</sub>
T <sub>1</sub>	A <sub>2</sub>	C <sub>3</sub>	G <sub>4</sub>	T <sub>5</sub>	G <sub>6</sub>	A <sub>7</sub>

# Multiple Sequence Alignment



# Multiple Sequence Alignment

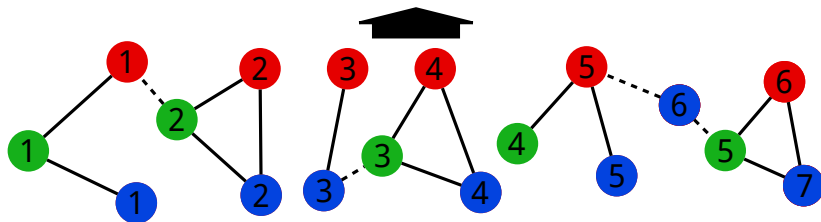


## Idea

Use alignment graph constructed by local alignment to reconstruct global alignment.

# Multiple Sequence Alignment

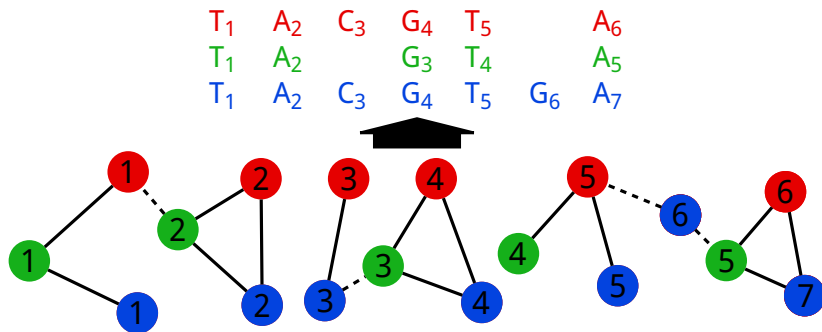
?



## Idea

Use alignment graph constructed by local alignment to reconstruct global alignment.

# Multiple Sequence Alignment



## Idea

Use alignment graph constructed by local alignment to reconstruct global alignment.

# Colorful Components

Part of a Multiple Sequence Alignment pipeline suggested by Corel, Pitschi & Morgenstern (Bioinformatics 2010).



# Colorful Components

Part of a Multiple Sequence Alignment pipeline suggested by Corel, Pitschi & Morgenstern (Bioinformatics 2010).

## COLORFUL COMPONENTS

**Instance:** An undirected graph  $G = (V, E)$  and a coloring of the vertices  $\chi : V \rightarrow \{1, \dots, c\}$ .

**Task:** Delete a minimum number of edges such that all connected components are *colorful*, that is, they do not contain two vertices of the same color.

# Other application: Wikipedia interlanguage links

Labyrinthulomycetes - Wikipedia, the free encyclopedia - Iceweasel

File Edit View History Bookmarks Tools Help

W Labyrinthulomycetes - Wikipedi... +

W en.wikipedia.org/wiki/Labyrinthulomycetes ☆ ↻ Google

Log in / create account

Article Talk

Read Edit View history Search

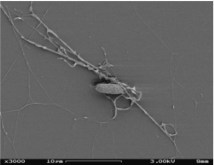
**Labyrinthulomycetes**

From Wikipedia, the free encyclopedia

The **Labyrinthulomycetes** (**ICBN**) or **Labyrinthulea**<sup>[1]</sup> (**ICZN**), or **Slime nets** are a **class** of **protists** that produce a network of **filaments** or tubes,<sup>[2]</sup> which serve as tracks for the cells to glide along and absorb **nutrients** for them. There are two main groups, the **labyrinthulids** and **thraustochytrids**. They are mostly **marine**, commonly found as **parasites** on **alga** and **seagrass** or as decomposers on dead plant material. They also include some parasites of marine invertebrates.

Although they are outside the cells, the filaments are surrounded by a **membrane**. They are formed and connected with the cytoplasm by a unique organelle called a **sagenogen** or **bothrosome**. The cells are uninucleate and typically ovoid, and move back and forth along the **amorphous** network at speeds varying from 5-150 µm per minute. Among the labyrinthulids the cells are enclosed within the tubes, and among the thraustochytrids they are attached to their sides.

**Slime nets**



The cell with the network of filaments *Aplanochytrium* sp.

**Scientific classification**

Domain: **Eukaryota**

Kingdom: **Chromalveolata**

Phylum: **Heterokontophyta**

Class: **Labyrinthulomycetes** DICK, 2001 or

WIKIPEDIA  
The Free Encyclopedia

Main page  
Contents  
Featured content  
Current events  
Random article  
Donate to Wikipedia

► Interaction

► Toolbox

► Print/export

▼ Languages

Česky  
Deutsch  
Español  
日本語  
Македонски  
Norsk (bokmål)

# Other application: Wikipedia interlanguage links

Labyrinthulomycetes - Wikipedia, the free encyclopedia - Iceweasel

File Edit View History Bookmarks Tools Help

W Labyrinthulomycetes - Wikipedi... +

en.wikipedia.org/wiki/Labyrinthulomycetes

Log in / create account

Article Talk

Read Edit View history Search

## Labyrinthulomycetes

From Wikipedia, the free encyclopedia

The **Labyrinthulomycetes** (**ICBN**) or **Labyrinthulea**<sup>[1]</sup> (**ICZN**), or **Slime nets** are a **class** of **protists** that produce a network of **filaments** or tubes,<sup>[2]</sup> which serve as tracks for the cells to glide along and absorb **nutrients** for them. There are two main groups, the **labyrinthulids** and **thraustochytrids**. They are mostly **marine**, commonly found as **parasites** on **alga** and **seagrass** or as decomposers on dead plant material. They also include some parasites of marine invertebrates. Although they are outside the cells, the filaments are

**Slime nets**



The cell with the network of filaments *Aplanochytrium* sp.

**Scientific classification**

Domain: **Eukaryota**  
 Kingdom: **Chromalveolata**  
 Phylum: **Heterokontophyta**  
 Class: **Labyrinthulomycetes** DICK, 2001 or

**Netzschleimpilze**

Die **Netzschleimpilze** oder **Schleimnetze** (Labyrinthulomycetes) bilden ein **Taxon** innerhalb der **Stramenopilen** und sind somit näher mit **Braunalgen**, **Goldalgen**

attached to their sides.

Deutsch

Espanol

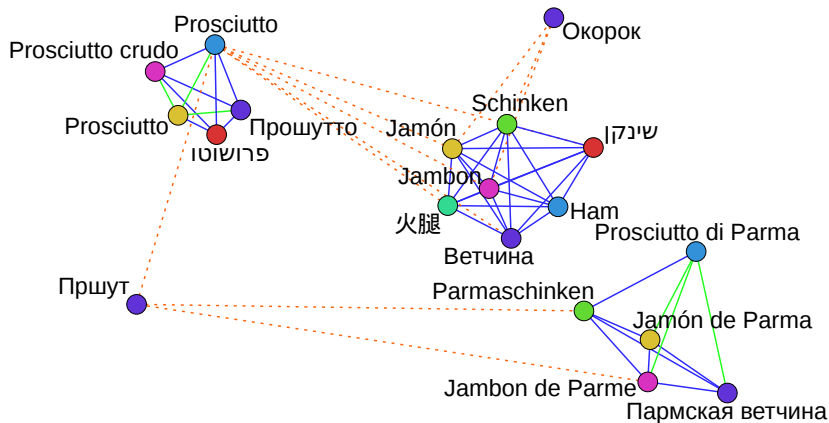
日本語

Македонски

Norsk (bokmål)

Русский

# Wikipedia interlanguage link graph example



# Complexity of Colorful Components

- COLORFUL COMPONENTS with two colors can be solved in  $O(\sqrt{nm})$  time by matching techniques.

# Complexity of Colorful Components

- COLORFUL COMPONENTS with two colors can be solved in  $O(\sqrt{nm})$  time by matching techniques.
- COLORFUL COMPONENTS is NP-hard already with three colors.

# Complexity of Colorful Components

- COLORFUL COMPONENTS with two colors can be solved in  $O(\sqrt{nm})$  time by matching techniques.
- COLORFUL COMPONENTS is NP-hard already with three colors.
- COLORFUL COMPONENTS is NP-hard on trees.

# Complexity of Colorful Components

- COLORFUL COMPONENTS with two colors can be solved in  $O(\sqrt{nm})$  time by matching techniques.
- COLORFUL COMPONENTS is NP-hard already with three colors.
- COLORFUL COMPONENTS is NP-hard on trees.
- COLORFUL COMPONENTS on trees with  $c$  colors can be solved in  $2^c \cdot n^{O(1)}$  time.



# Fixed-parameter algorithm

## Observation

COLORFUL COMPONENTS can be seen as the problem of destroying by edge deletions all **bad paths**, that is, simple paths between equally colored vertices.

# Fixed-parameter algorithm

## Observation

COLORFUL COMPONENTS can be seen as the problem of destroying by edge deletions all **bad paths**, that is, simple paths between equally colored vertices.

## Observation

Unless the graph is already colorful, we can always find a bad path with at most  $c$  edges, where  $c$  is the number of colors.

# Fixed-parameter algorithm

## Observation

COLORFUL COMPONENTS can be seen as the problem of destroying by edge deletions all **bad paths**, that is, simple paths between equally colored vertices.

## Observation

Unless the graph is already colorful, we can always find a bad path with at most  $c$  edges, where  $c$  is the number of colors.

## Theorem

*COLORFUL COMPONENTS can be solved in  $O(c^k \cdot m)$  time, where  $k$  is the number of edge deletions.*

# Improved fixed-parameter algorithm

## Theorem

*COLORFUL COMPONENTS can be solved in  $O((c - 1)^k \cdot m)$  time, where  $k$  is the number of edge deletions.*

# Improved fixed-parameter algorithm

## Theorem

*COLORFUL COMPONENTS can be solved in  $O((c - 1)^k \cdot m)$  time, where  $k$  is the number of edge deletions.*

## Proof.

If there is a degree-3 or higher vertex  $v$ , find a bad path with at most  $(c - 1)$  edges by BFS from  $v$ . Otherwise, the instance is easy. □

# Limits of fixed-parameter algorithms

## Question

How much further can we improve this algorithm?

# Limits of fixed-parameter algorithms

## Question

How much further can we improve this algorithm?

## Exponential Time Hypothesis (ETH)

For all  $x \geq 3$ ,  $x$ -SAT, which asks whether a Boolean input formula in conjunctive normal form with  $n$  variables and  $m$  clauses and at most  $x$  variables per clause is satisfiable, cannot be solved within a running time of  $2^{o(n)}$  or  $2^{o(m)}$ .

# Limits of fixed-parameter algorithms

## Question

How much further can we improve this algorithm?

## Exponential Time Hypothesis (ETH)

For all  $x \geq 3$ ,  $x$ -SAT, which asks whether a Boolean input formula in conjunctive normal form with  $n$  variables and  $m$  clauses and at most  $x$  variables per clause is satisfiable, cannot be solved within a running time of  $2^{o(n)}$  or  $2^{o(m)}$ .

## Theorem

*COLORFUL COMPONENTS with three colors cannot be solved in  $2^{o(k)} \cdot n^{O(1)}$  unless the ETH is false.*



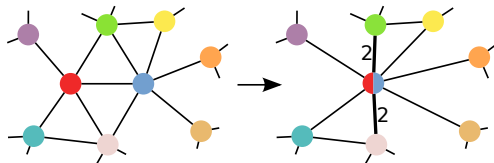
# Weighted version

## Problem

If we know that two vertices must belong to the same colorful component, we want to be able to simplify the instance by merging them.

## Idea

Introduce color *sets* per vertex and edge weights.



# Uses of the merge operation

## Edge branching

Can branch into two cases: delete an edge, or merge its endpoints.

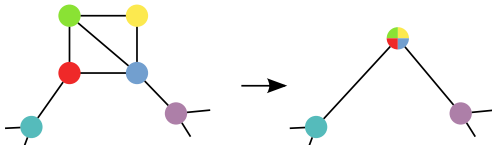
# Uses of the merge operation

## Edge branching

Can branch into two cases: delete an edge, or merge its endpoints.

## Data reduction

Let  $V' \subseteq V$  be a colorful subgraph. If the cut between  $V'$  and  $V \setminus V'$  is at least as large as the connectivity of  $V'$ , then merge  $V'$  into a single vertex.



# Merge-based heuristic

## Idea

Repeatedly merge the two vertices “most likely” to be in the same component, while immediately deleting edges connecting vertices with intersecting color sets.

# Merge-based heuristic

## Idea

Repeatedly merge the two vertices “most likely” to be in the same component, while immediately deleting edges connecting vertices with intersecting color sets.

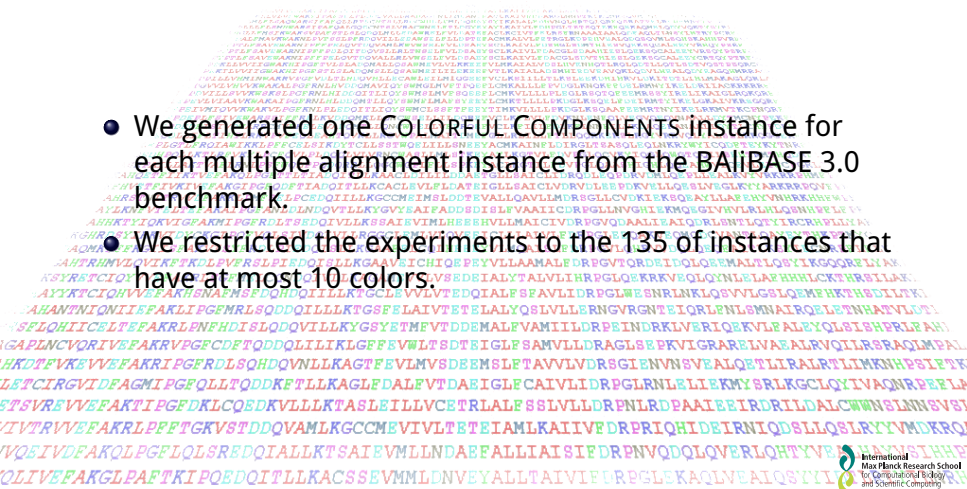
We always merge the endpoints of the edge that maximizes cut cost minus merge cost.

- *merge cost*: weight of the edges that would need to be deleted when merging
- *cut cost*:

$$3w(\{u, v\}) + \sum_{w \in V | \{\{u, w\}, \{v, w\}\} \subseteq E} \min\{w(\{u, w\}), w(\{v, w\})\}$$

## Data

- We generated one COLOREFU COMPONENTS instance for each multiple alignment instance from the BALiBASE 3.0 benchmark
- We restricted the experiments to the 135 of instances that have at most 10 colors



# Data reduction: Largest connected component

- (1) originally
- (2) after data reduction in the COLORFUL COMPONENTS formulation
- (3) after data reduction in the weighted formulation

	(1)			(2)			(3)		
	<i>n</i>	<i>m</i>	<i>c</i>	<i>n</i>	<i>m</i>	<i>c</i>	<i>n</i>	<i>m</i>	<i>c</i>
average	504	921	6.2	407	697	4.7	354	607	5.3
median	149	232	6	46	90	5	42	58	5

# Branching algorithms: running time

	< 1 s	1 s to 10 min	> 10 min
bad-path branching	61	6	68
merging branching	70	9	56



# Branching algorithms: running time

	< 1 s	1 s to 10 min	> 10 min
bad-path branching	61	6	68
merging branching	70	9	56

## Note

In ongoing research, we are able to solve several more instances to optimality with integer linear programming (ILP) based approaches.

# Heuristics: relative error

	min.	max.	avg.	med.
min-cut heuristic [1]	0 % (1)	70.0 %	29.2 %	27.8 %
merging heuristic	0 % (76)	12.7 %	0.6 %	0 %

[1] Corel, Pitschi & Morgenstern (Bioinformatics 2010)

# Sequence alignment quality

DIALIGN with several methods for solving the COLORFUL COMPONENTS subproblem:

	TC score
min-cut heuristic	53.6 %
merge heuristic	55.1 %
exact algorithm	56.6 %

# Sequence alignment quality

DIALIGN with several methods for solving the COLORFUL COMPONENTS subproblem:

	TC score
min-cut heuristic	53.6 %
merge heuristic	55.1 %
exact algorithm	56.6 %

DIALIGN with the min-cut heuristic is about 10 percentage points worse than current state-of-the-art multiple alignment methods. Hence, an improvement of 3 percentage points is a sizable step towards closing the gap between DIALIGN and these methods.

# Outlook

- ILP-based solutions
- Application to network alignment
- Relaxation of the colorfulness constraint

# Acknowledgments

Sharon Bruckner

Freie Universität



Berlin



International  
Max Planck Research School  
for Computational Biology  
and Scientific Computing



Falk Hüffner, Christian Komusiewicz, Rolf Niedermeier,  
Johannes Uhlmann



Sven Thiel



seit 1558

