On the Complexity of the Quartet Challenge

Simone Linz

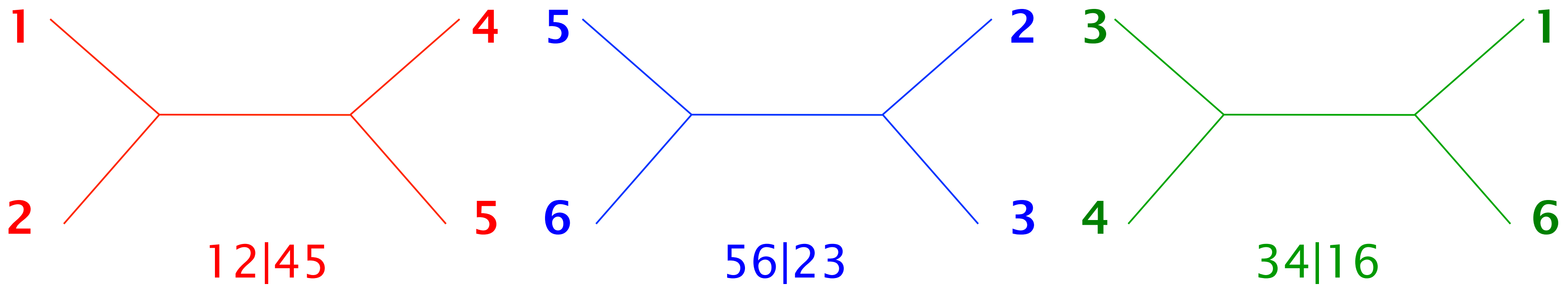
Center for Bioinformatics
University of Tübingen, Germany

Joint work with Maria Luisa Bonet and Katherine St. John

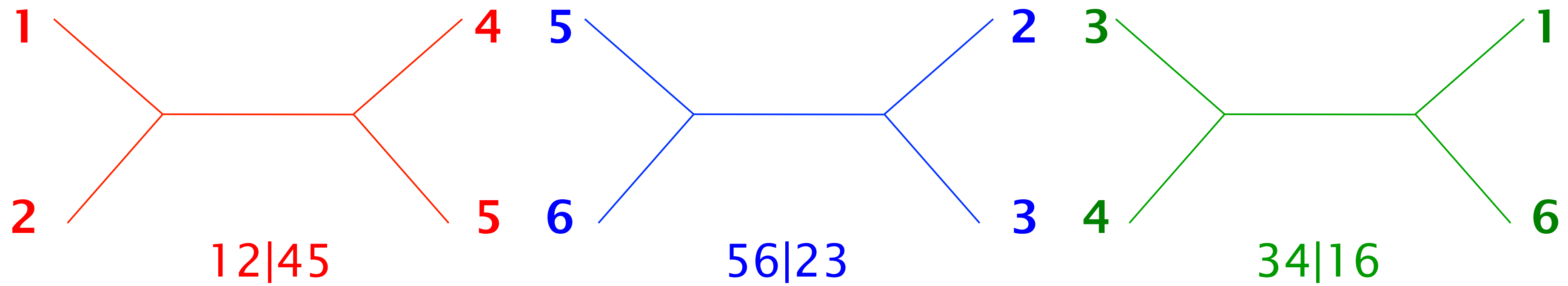
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

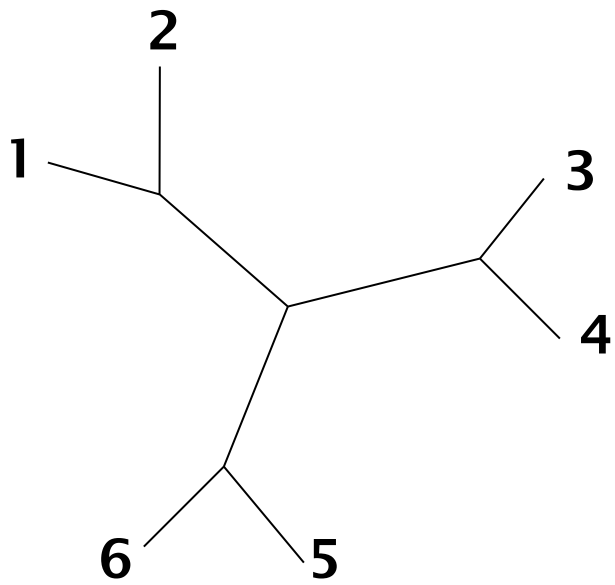
Isaac Newton Institute for
Mathematical Sciences,
Cambridge, June 2011.

Compatibility

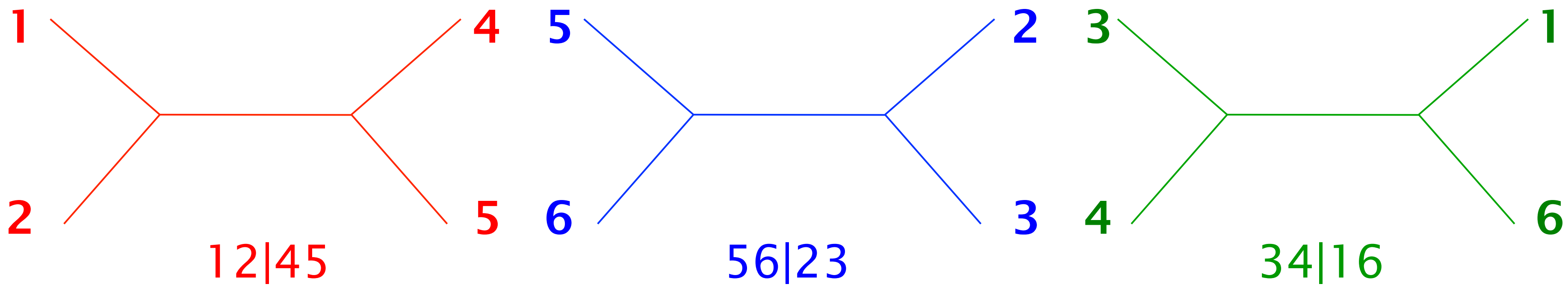


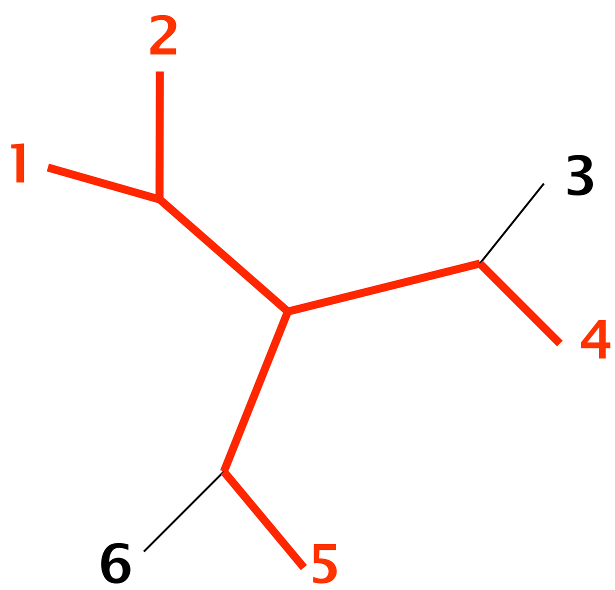
Compatibility



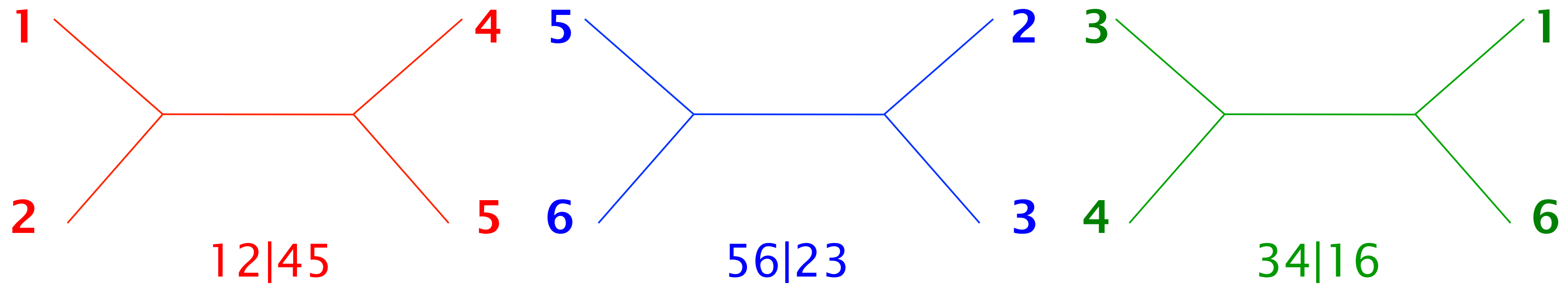


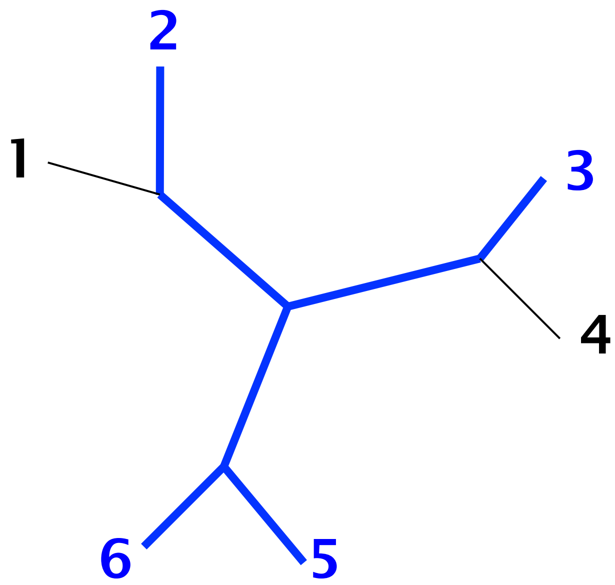
Compatibility



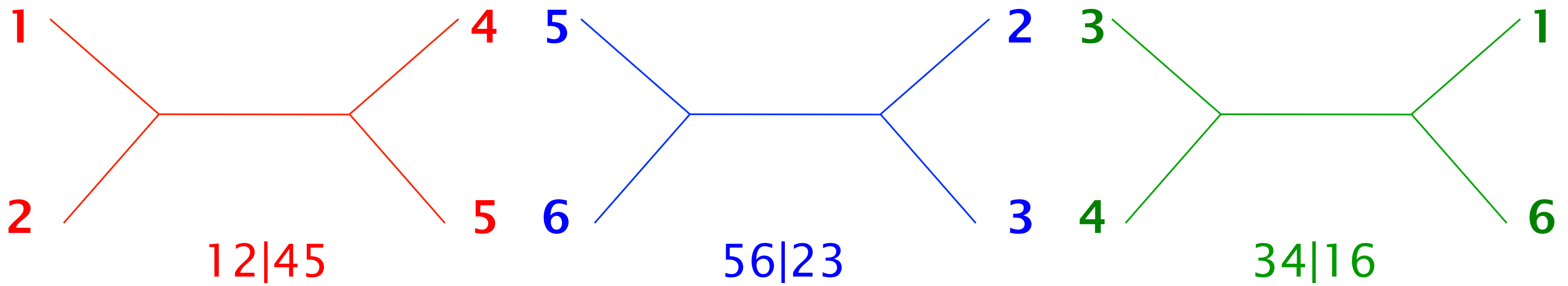


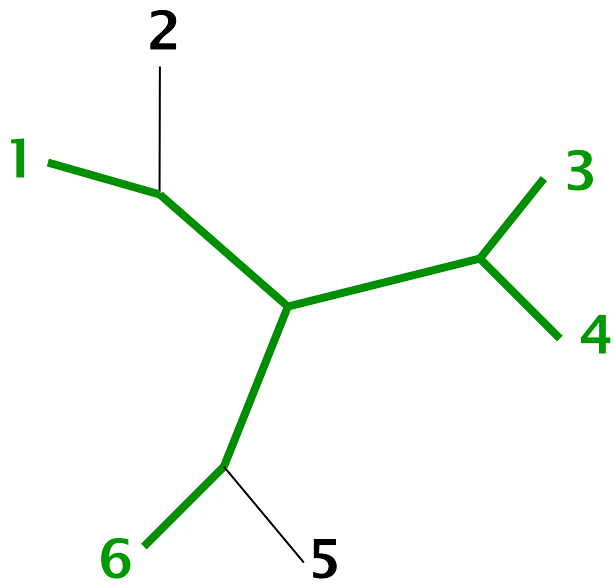
Compatibility



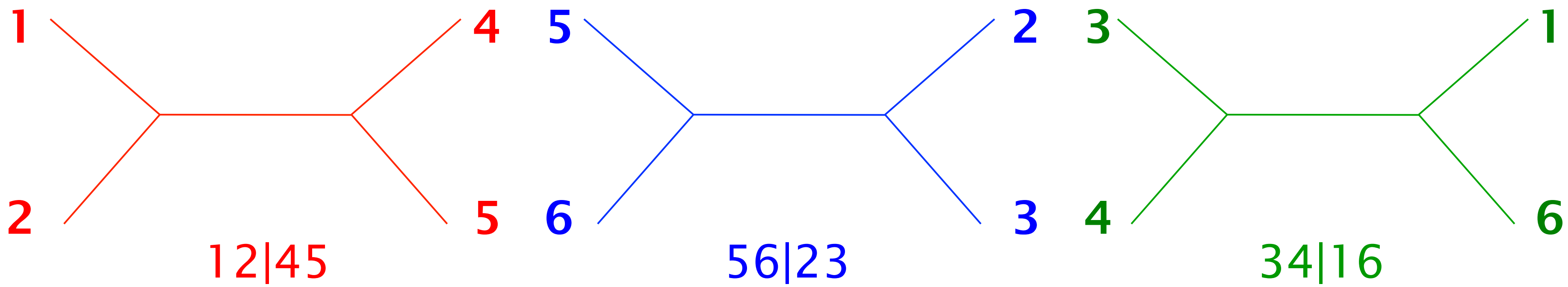


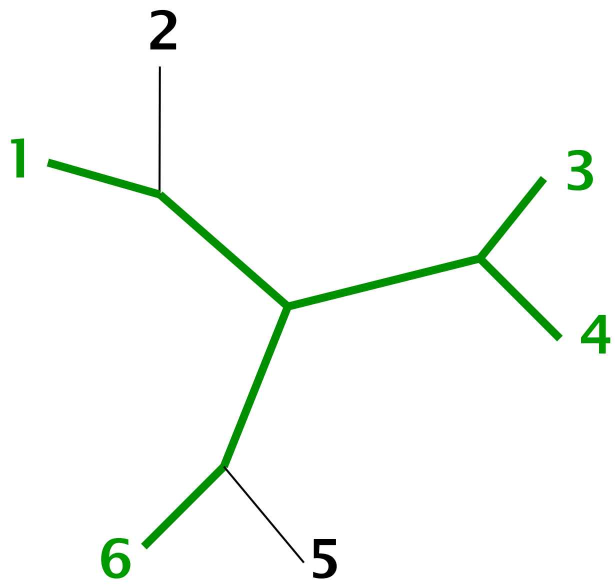
Compatibility





Compatibility





The three quartets are *compatible*.

The tree *displays* the quartets.

<u>QUARTET COMPATIBILITY</u>

Instance. A set Q of quartets (binary unrooted trees on four taxa).

Question. Is Q compatible?

QUARTET COMPATIBILITY

Instance. A set Q of quartets (binary unrooted trees on four taxa).

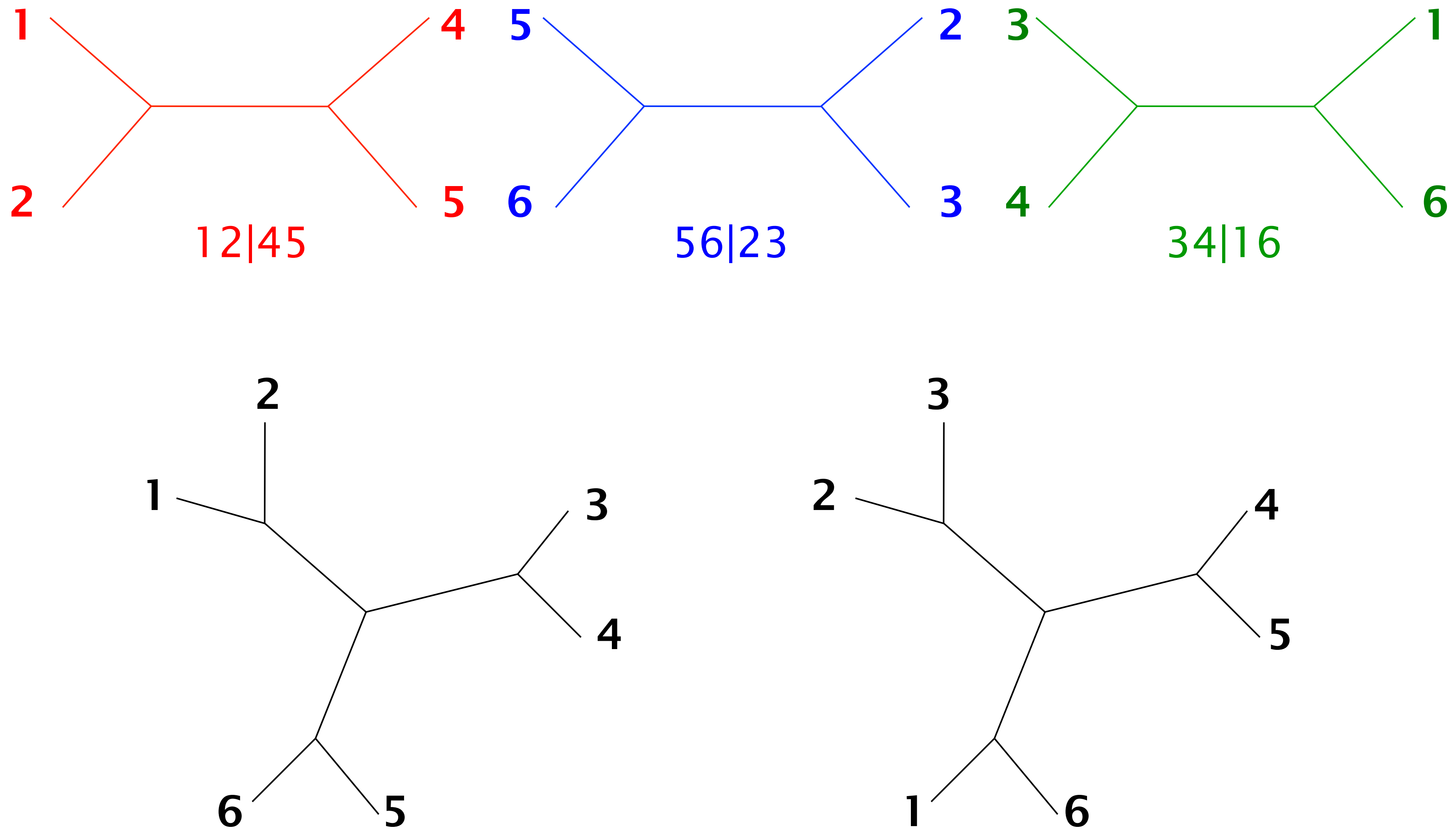
Question. Is Q compatible?

Theorem. (Steel, 1992)

QUARTET COMPATIBILITY is NP-complete.

Follow-up question:

What about uniqueness?


1
2
4
5
12|45
5
6
2
3
56|23
3
4
1
6
34|16
2
1
3
4
6
5
3
2
4
5
1
6

Is the following question NP-hard?

QUARTET CHALLENGE

Instance. A binary phylogenetic ***X***-tree ***T*** and a set ***Q*** of quartets on ***X*** such that ***T*** displays ***Q***.

Question. Is ***T*** the unique phylogenetic ***X***-tree that displays ***Q*?**

Is the following question NP-hard?

QUARTET CHALLENGE

Instance. A binary phylogenetic ***X***-tree ***T*** and a set ***Q*** of quartets on ***X*** such that ***T*** displays ***Q***.

Question. Is ***T*** the unique phylogenetic ***X***-tree that displays ***Q*?**

Characterizations of the QUARTET CHALLENGE:

- **Chordal and intersection graphs (Semple, Steel, 2002).**
- **Quartet graphs and edge colorings (Grünewald, Humphries, Semple, 2008).**

Is the following question NP-hard?

QUARTET CHALLENGE

Instance. A binary phylogenetic ***X***-tree ***T*** and a set ***Q*** of quartets on ***X*** such that ***T*** displays ***Q***.

Question. Is ***T*** the unique phylogenetic ***X***-tree that displays ***Q?***

QUARTET CHALLENGE*

Instance. A binary phylogenetic ***X***-tree ***T*** and a set ***Q*** of quartets on ***X*** such that ***T*** displays ***Q***.

Question. Is there a binary phylogenetic ***X***-tree other than ***T*** that displays ***Q?***

Is the following question NP-hard?

QUARTET CHALLENGE

Instance. A binary phylogenetic ***X***-tree ***T*** and a set ***Q*** of quartets on ***X*** such that ***T*** displays ***Q***.

Question. Is ***T*** the unique phylogenetic ***X***-tree that displays ***Q***?

QUARTET CHALLENGE*

Instance. A binary phylogenetic ***X***-tree ***T*** and a set ***Q*** of quartets on ***X*** such that ***T*** displays ***Q***.

Question. Is there a binary phylogenetic ***X***-tree other than ***T*** that displays ***Q***?

QUARTET CHALLENGE and QUARTET CHALLENGE* are complementary.

Our Contribution

Theorem. (Bonet, L., St. John, 2011)

The QUARTET CHALLENGE* is NP-complete and, in particular, the QUARTET CHALLENGE is coNP-complete.

ASP-completeness
(Yato and Seta, 2003)

The **A**nother **S**olution **P**roblem (ASP) of a problem Π' is the following:

Input. Instance σ' of Π' and a solution s' to σ'.
Question. Does there exists a solution to σ' other than s'?

ASP-completeness

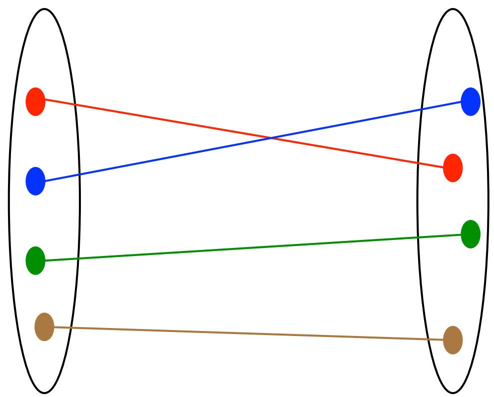
(Yato and Seta, 2003)

The **A**nother **S**olution **P**roblem (ASP) of a problem ***Π’*** is the following:

> **Input.** Instance ***σ’*** of ***Π’*** and a solution ***s’*** to ***σ’***.
> **Question.** Does there exists a solution to ***Π’*** other than ***s’***?

Definition. A polynomial–time reduction ***f*** from ***Π*** to ***Π’*** is an ***ASP–reduction*** precisely if for any instance ***σ*** of ***Π*** there is a bijection from the solutions of ***σ*** to the solutions of ***f(σ)*** (an instance of ***Π’*** obtained under ***f***).

Space of solutions for *σ*.



Space of solutions for *f(σ)*.

ASP-completeness
(Yato and Seta, 2003)

The **A**nother **S**olution **P**roblem (ASP) of a problem ***Π'*** is the following:

Input. Instance ***σ'*** of ***Π'*** and a solution ***s'*** to ***σ'***.
Question. Does there exists a solution to ***Π*** other than ***s***?

Definition. A polynomial-time reduction ***f*** from ***Π*** to ***Π'*** is an ***ASP-reduction*** precisely if for any instance ***σ*** of ***Π*** there is a bijection from the solutions of ***σ*** to the solutions of ***f(σ)*** (an instance of ***Π'*** obtained under ***f***).

Informally, a problem ***Π'*** is ***ASP-complete*** if and only if there is an ASP-reduction from ***Π*** to ***Π'*** for an ASP-complete problem ***Π***.

ASP-completeness
(Yato and Seta, 2003)

The **A**nother **S**olution **P**roblem (ASP) of a problem ***Π'*** is the following:

> **Input.** Instance ***σ'*** of ***Π'*** and a solution ***s'*** to ***σ'***.
> **Question.** Does there exists a solution to ***Π'*** other than ***s'***?

Definition. A polynomial-time reduction ***f*** from ***Π*** to ***Π'*** is an ***ASP-reduction*** precisely if for any instance ***σ*** of ***Π*** there is a bijection from the solutions of ***σ*** to the solutions of ***f(σ)*** (an instance of ***Π'*** obtained under ***f***).

Informally, a problem ***Π'*** is ***ASP-complete*** if and only if there is an ASP-reduction from ***Π*** to ***Π'*** for an ASP-complete problem ***Π***.

Theorem. If a problem is ASP-complete, then it is NP-complete to decide if, given a solution, another solution exists (unless P=NP).

<u>3SAT</u>

Instance. A conjunction of clauses, each clause with 3 literals over a set of variables.

Question. Does there exist a truth assignment such that each clause contains at least one literal assigned to ***true***?

Example.

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4)$$

3SAT

Instance. A conjunction of clauses, each clause with 3 literals over a set of variables.

Question. Does there exist a truth assignment such that each clause contains at least one literal assigned to ***true***?

Example.

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4)$$

$$x_1 = x_2 = x_3 = x_4 = true$$

$$x_3 = \bar{x}_1 = \bar{x}_2 = \bar{x}_4 = true$$

3SAT

Instance. A conjunction of clauses, each clause with 3 literals over a set of variables.

Question. Does there exist a truth assignment such that each clause contains at least one literal assigned to ***true***?

Theorem. (Yato and Seta, 2003)

3SAT is ASP-complete.

NOT-ALL-EQUAL-3SAT (with constants)

Instance. A conjunction of clauses, each clause with 3 literals over a set of variables and the two constants ***T*** and ***F***.

Question. Does there exist a truth assignment with ***T=true*** and ***F=false*** such that each clause contains at least one literal assigned to ***true*** and one literal assigned to ***false***?

Example.

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4)$$

NOT-ALL-EQUAL-3SAT (with constants)

Instance. A conjunction of clauses, each clause with 3 literals over a set of variables and the two constants ***T*** and ***F***.

Question. Does there exist a truth assignment with ***T=true*** and ***F=false*** such that each clause contains at least one literal assigned to ***true*** and one literal assigned to ***false***?

Example.

$$(x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4)$$

~~$x_1 = x_2 = x_3 = x_4 = true$~~

$$x_3 = \bar{x}_1 = \bar{x}_2 = \bar{x}_4 = true$$

NOT-ALL-EQUAL-3SAT WITH CONSTANTS

Instance. A conjunction of clauses, each clause with 3 literals over a set of variables and the two constants ***T*** and ***F***.

Question. Does there exist a truth assignment with ***T=true*** and ***F=false*** such that each clause contains at least one literal assigned to ***true*** and one literal assigned to ***false***?

Theorem. (Bonet, L., St. John, 2011)

NOT-ALL-EQUAL-3SAT WITH CONSTANTS is ASP-complete.

(Proof: ASP-reduction from 3SAT.)

BETWEENNESS
used to show that QUARTET COMPATIBILITY is hard

Instance. A collection of ordered triples ***(a,b,c)*** over a set ***A***.

Question. Does there exist an ordering ***f: A → {0,1,…,|A|-1}*** such that for each triple either ***f(a)<f(b)<f(c)*** or ***f(c)<f(b)<f(a)***.

BETWEENNESS WITH CONSTANTS

Instance. A collection of ordered triples ***(a,b,c)*** over a set ***A*** ∪ ***{M,m}***.

Question. Does there exist a betweenness ordering ***f: A*** ∪ ***{M,m} → {0,1,…,|A|+1}*** such that for each triple either ***f(a)<f(b)<f(c)*** or ***f(c)<f(b)<f(a)***, and ***f(m)=0*** and ***f(M)=|A|+1***.

Example.

Input: *(m,a,b), (M,c,b), (c,a,m), (m,b,M)*.

The ordering ***m<a<b<c<M*** satisfies all four constraints.

BETWEENNESS WITH CONSTANTS

Instance. A collection of ordered triples ***(a,b,c)*** over a set ***A*** ∪ ***{M,m}***.

Question. Does there exist a betweenness ordering ***f: A*** ∪ ***{M,m}*** → ***{0,1,…,|A|+1}*** such that for each triple either ***f(a)<f(b)<f(c)*** or ***f(c)<f(b)<f(a)***, and ***f(m)=0*** and ***f(M)=|A|+1***.

Example.

Input: *(m,a,b)*, *(M,c,b)*, *(c,a,m)*, *(m,b,M)*.

The ordering ***m<a<b<c<M*** satisfies all four constraints.

BETWEENNESS WITH CONSTANTS

Instance. A collection of ordered triples ***(a,b,c)*** over a set ***A*** ∪ ***{M,m}***.

Question. Does there exist a betweenness ordering ***f: A*** ∪ ***{M,m}*** → ***{0,1,…,|A|+1}*** such that for each triple either ***f(a)<f(b)<f(c)*** or ***f(c)<f(b)<f(a)***, and ***f(m)=0*** and ***f(M)=|A|+1***.

Example.

Input: *(m,a,b)*, *(M,c,b)*, *(c,a,m)*, *(m,b,M)*.

The ordering ***m<a<b<c<M*** satisfies all four constraints.

BETWEENNESS WITH CONSTANTS

Instance. A collection of ordered triples ***(a,b,c)*** over a set ***A*** ∪ ***{M,m}***.

Question. Does there exist a betweenness ordering ***f: A*** ∪ ***{M,m}*** → ***{0,1,…,|A|+1}*** such that for each triple either ***f(a)<f(b)<f(c)*** or ***f(c)<f(b)<f(a)***, and ***f(m)=0*** and ***f(M)=|A|+1***.

Example.

Input: *(m,a,b)*, *(M,c,b)*, *(c,a,m)*, *(m,b,M)*.

The ordering ***m<a<b<c<M*** satisfies all four constraints.

<u>BETWEENNESS WITH CONSTANTS</u>

Instance. A collection of ordered triples ***(a,b,c)*** over a set $A \cup \{M,m\}$.

Question. Does there exist a betweenness ordering $f: A \cup \{M,m\} \rightarrow \{0,1,\ldots,|A|+1\}$ such that for each triple either $f(a)<f(b)<f(c)$ or $f(c)<f(b)<f(a)$, and $f(m)=0$ and $f(M)=|A|+1$.

Example.

Input: *(m,a,b), (M,c,b), (c,a,m), (m,b,M)*.

The ordering $m<a<b<c<M$ satisfies all four constraints.

BETWEENNESS WITH CONSTANTS

Instance. A collection of ordered triples (a,b,c) over a set $A \cup \{M,m\}$.

Question. Does there exist a betweenness ordering $f: A \cup \{M,m\} \rightarrow \{0,1,\ldots,|A|+1\}$ such that for each triple either $f(a)<f(b)<f(c)$ or $f(c)<f(b)<f(a)$, and $f(m)=0$ and $f(M)=|A|+1$.

Theorem. (Bonet, L., St. John, 2011)

BETWEENNESS WITH CONSTANTS is ASP-complete.

(Proof: ASP-reduction from NOT-ALL-EQUAL-3SAT WITH CONSTANTS.)

QUARTET CHALLENGE*

Instance. A binary phylogenetic ***X***-tree ***T*** and a set ***Q*** of quartets on ***X*** such that ***T*** displays ***Q***.

Question. Is there a binary phylogenetic ***X***-tree other than ***T*** that displays ***Q*?**

Theorem. (Bonet, L., St. John, 2011)

The QUARTET CHALLENGE* is ASP-complete.

Sketch of Proof (I)

ASP-reduction from BETWEENNESS WITH CONSTANTS.

For each BETWEENNESS constraint $\pi_i=(a_i,b_i,c_i)$, add a set S_i of *6* quartets (Steel, 1992):

- $p_i p_i' | a_i b_i$
- $p_i a_i | b_i c_i$
- $p_i b_i | c_i q_i$
- $p_i c_i | q_i q_i'$
- $\alpha p_i | p_i' \beta$
- $\alpha q_i | q_i' \beta$.

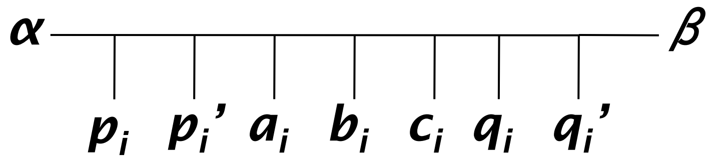
Sketch of Proof (I)

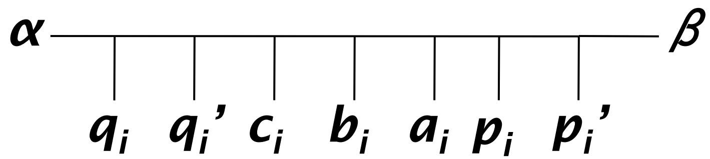
ASP-reduction from BETWEENNESS WITH CONSTANTS.

For each BETWEENNESS constraint $\pi_i=(a_i,b_i,c_i)$, add a set S_i of *6* quartets (Steel, 1992):

- $p_i p_i'|a_i b_i$
- $p_i a_i|b_i c_i$
- $p_i b_i|c_i q_i$
- $p_i c_i|q_i q_i'$
- $\alpha p_i|p_i'\beta$
- $\alpha q_i|q_i'\beta$.

There are exactly two phylogenetic trees that display all 6 quartets:





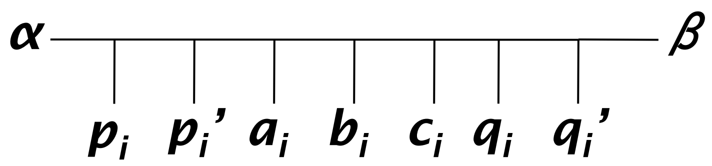
Sketch of Proof (I)

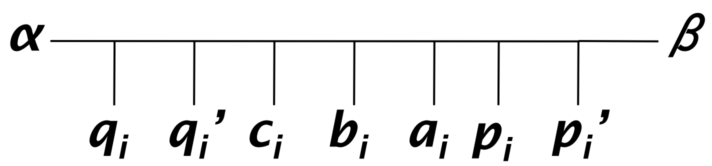
ASP-reduction from BETWEENNESS WITH CONSTANTS.

For each BETWEENNESS constraint $\pi_i=(a_i,b_i,c_i)$, add a set S_i of *6* quartets (Steel, 1992):

- $p_i p_i'|a_i b_i$
- $p_i a_i|b_i c_i$
- $p_i b_i|c_i q_i$
- $p_i c_i|q_i q_i'$
- $\alpha p_i|p_i'\beta$
- $\alpha q_i|q_i'\beta$.

There are exactly two phylogenetic trees that display all 6 quartets:





Union of quartets over all sets S_i can be displayed by a caterpillar if and only if there exists a BETWEENNESS ordering.

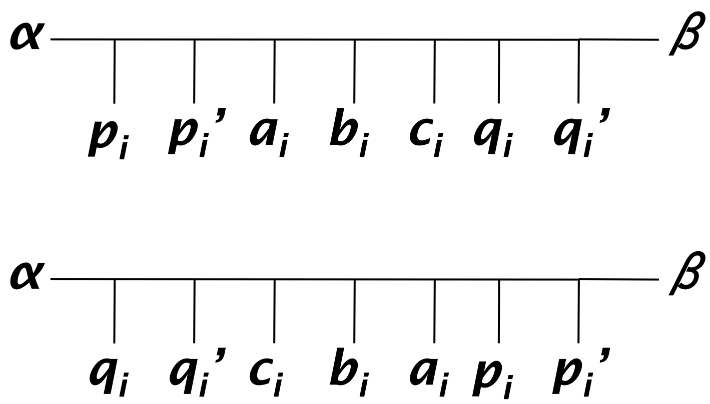
Sketch of Proof (I)

ASP-reduction from BETWEENNESS WITH CONSTANTS.

For each BETWEENNESS constraint $\pi_i=(a_i,b_i,c_i)$, add a set S_i of *6* quartets (Steel, 1992):

- $p_ip_i'|a_ib_i$
- $p_ia_i|b_ic_i$
- $p_ib_i|c_iq_i$
- $p_ic_i|q_iq_i'$
- $\alpha p_i|p_i'\beta$
- $\alpha q_i|q_i'\beta$.

There are exactly two phylogenetic trees that display all 6 quartets:



However, this caterpillar is not unique since the union over all sets S_i does not impose an ordering on the ***p***'s and ***q***'s.

Sketch of Proof (II)

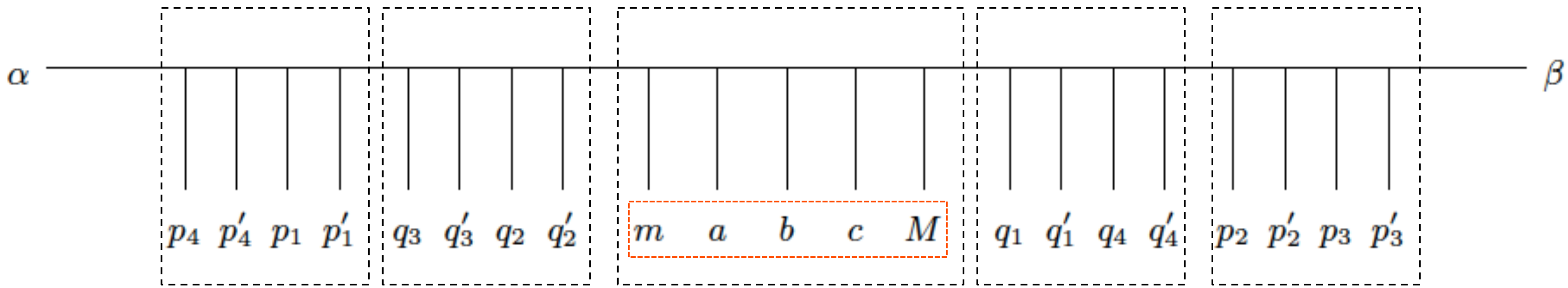
Thus, to get an ASP-reduction, we need a set Q of quartets that uniquely defines a phylogenetic tree for a given BETWEENNESS ordering.

Example. $\pi_1=(m,a,b)$, $\pi_2=(M,c,b)$, $\pi_3=(c,a,m)$, $\pi_4=(m,b,M)$, $A=\{m,M,a,b,c\}$.

Add in quartets such that the p's and q's are ordered among themselves on each side of the caterpillar.

Add in quartets such that all the p's are closer to α and β than all the q's.

…



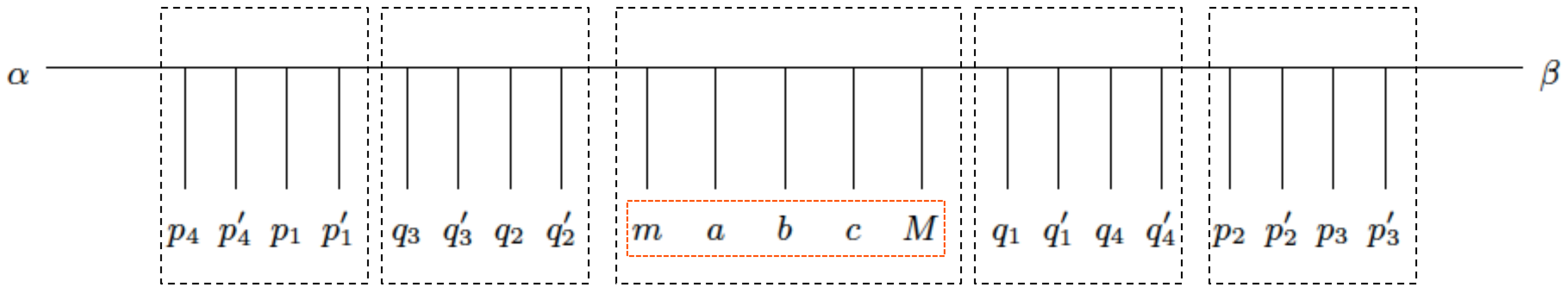
BETWEENNESS ordering for $\pi_1,\ldots,\pi_4$.

Sketch of Proof (II)

Thus, to get an ASP-reduction, we need a set ***Q*** of quartets that uniquely defines a phylogenetic tree for a given BETWEENNESS ordering.

Example. $\pi_1=(m,a,b)$, $\pi_2=(M,c,b)$, $\pi_3=(c,a,m)$, $\pi_4=(m,b,M)$, $A=\{m,M,a,b,c\}$.

Claim 1. Two phylogenetic trees ***T*** and ***T'*** that both display the set ***Q*** of quartets are isomorphic if and only if $T|(A \cup \{\alpha,\beta\})$ and $T'|(A \cup \{\alpha,\beta\})$ are isomorphic.



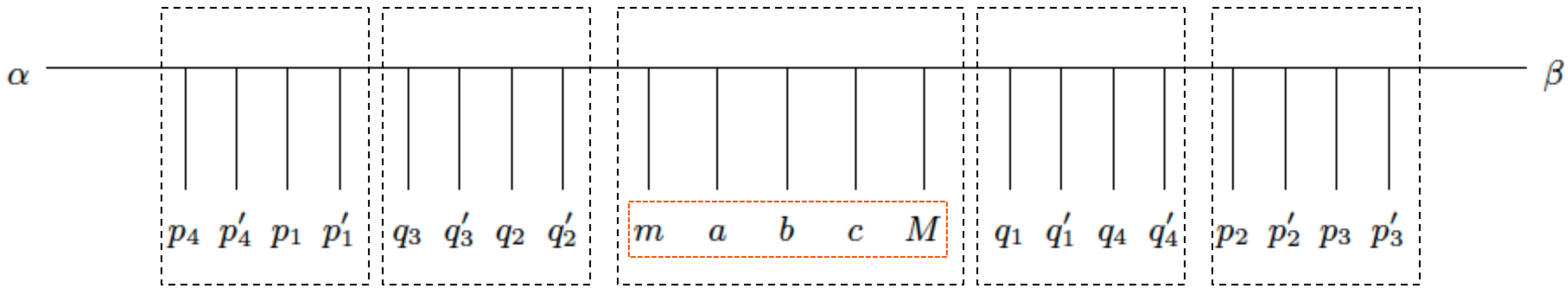
BETWEENNESS ordering for $\pi_1,\ldots,\pi_4$.

Sketch of Proof (II)

Thus, to get an ASP-reduction, we need a set ***Q*** of quartets that uniquely defines a phylogenetic tree for a given BETWEENNESS ordering.

Example. $\pi_1=(m,a,b)$, $\pi_2=(M,c,b)$, $\pi_3=(c,a,m)$, $\pi_4=(m,b,M)$, $A=\{m,M,a,b,c\}$.

Claim 2. The set ***Q*** of quartets is compatible if and only if ***A*** has a BETWEENNESS ordering ***f*** such that $f(m)=0$ and $f(M)=|A|-1$. In particular, there is a bijection from the solutions of a BETWEENNESS instance to the set of compatible trees that display ***Q***.



BETWEENNESS ordering for $\pi_1,\ldots,\pi_4$.

Theorem. (Bonet, L., St. John, 2011)

The QUARTET CHALLENGE* is ASP-complete.

In particular, the QUARTET CHALLENGE* is NP-complete and the QUARTET CHALLENGE is coNP-complete.

Acknowledgements

Mike Steel
for posing the QUARTET CHALLENGE.

Spanish Grant (TIN2007-68005-C04-03) and
the University of Tübingen
for funding.

INI / Organizers!

Aside: Radiation Hybrid Mapping

Method. Chromosomes are separated from one another and broken into several fragments using high doses of X-rays.

Goal. Construction of long-range maps of chromosomes, and determination not only of the distances between DNA markers but also of their order on the chromosomes.

Aside: Radiation Hybrid Mapping

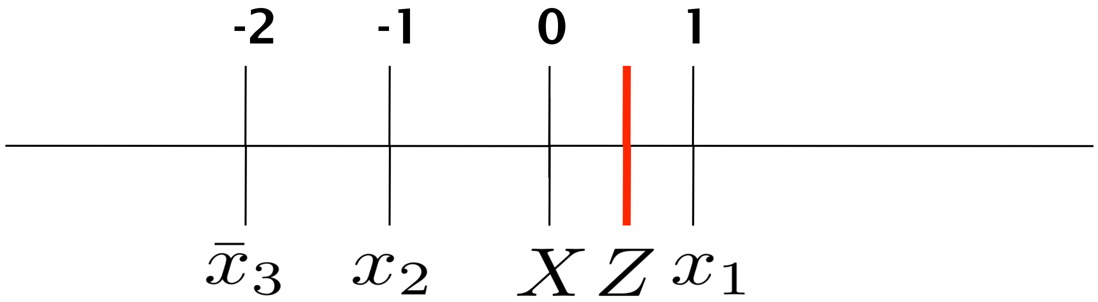
Method. Chromosomes are separated from one another and broken into several fragments using high doses of X-rays.

Goal. Construction of long-range maps of chromosomes, and determination not only of the distances between DNA markers but also their order on the chromosomes.

The problem of assembling the DNA fragments can be modeled by the decision problem BETWEENNESS (Chor and Sudan, 1998).

For each clause $(x_1 \vee x_2 \vee \bar{x}_3)$, add the two triples (x_1, Z, x_2) and $(\bar{x}_3, X, Z)$ and order all variables on a number line.

$$x_1 = true$$
$$x_2 = \bar{x}_3 = false$$



Note that more auxiliary variables are needed to order the ***Z***'s among themselves.

Radiation hybrid mapping: A problem that aims at constructing chromosome maps given DNA markers/ fragments can be modeled by the decision problem BETWEENNESS (Chor and Sudan, 1998).

Radiation hybrid mapping: A problem that aims at constructing chromosome maps given DNA markers/ fragments can be modeled by the decision problem <u>BETWEENNESS</u> (Chor and Sudan, 1998).

Since the 3’ and 5’ end of a DNA strand are frequently known, we study the following variation of BETWEENNESS: