

# Problems in Graph Theory With Applications to Topology and Modeling RNA

**Rayan Ibrahim**

Department of Mathematics and Applied Mathematics  
Virginia Commonwealth University

July 3, 2024

<https://raymaths.github.io/Defense.pdf>

Acknowledgement: Supported in part by The Thomas F. and Kate Miller Jeffress Memorial Trust, Bank of America, Trustee and by National Science Foundation DMS-2204148.

**1** Modeling RNA Via Chord Diagrams and Intersection Graphs

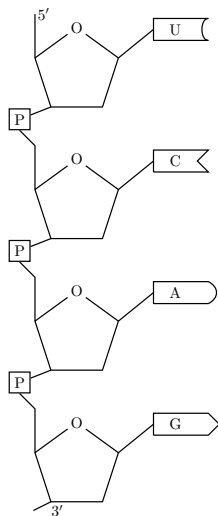
**2** 2-Neighbor Bootstrap Percolation

# Outline

**1** Modeling RNA Via Chord Diagrams and Intersection Graphs

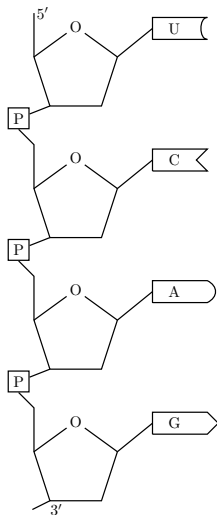
**2** 2-Neighbor Bootstrap Percolation

# What is RNA?



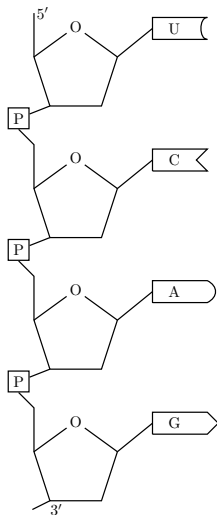
- Ribonucleic acid (RNA) is single stranded molecule comprising a phosphate-sugar backbone and to each sugar a base is attached.

# What is RNA?



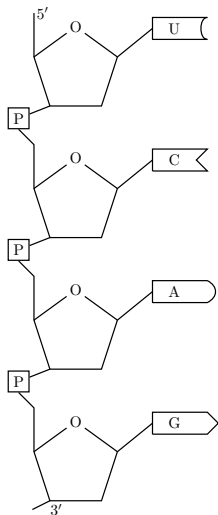
- Ribonucleic acid (RNA) is single stranded molecule comprising a phosphate-sugar backbone and to each sugar a base is attached.
- Bases: Adenine, Guanine, Cytosine, Uracil

# What is RNA?



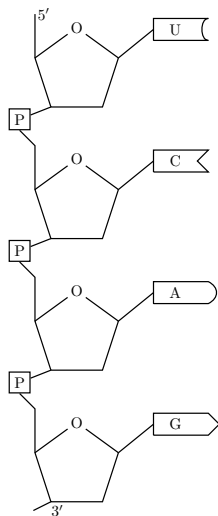
- Ribonucleic acid (RNA) is single stranded molecule comprising a phosphate-sugar backbone and to each sugar a base is attached.
- Bases: Adenine, Guanine, Cytosine, Uracil
- Strand orientation: 5' to 3'.

# What is RNA?



- Ribonucleic acid (RNA) is single stranded molecule comprising a phosphate-sugar backbone and to each sugar a base is attached.
- Bases: Adenine, Guanine, Cytosine, Uracil
- Strand orientation: 5' to 3'.
- Bases may form hydrogen bonds.  
Watson-Crick: A-U, G-C. Wobble: G-U

# What is RNA?



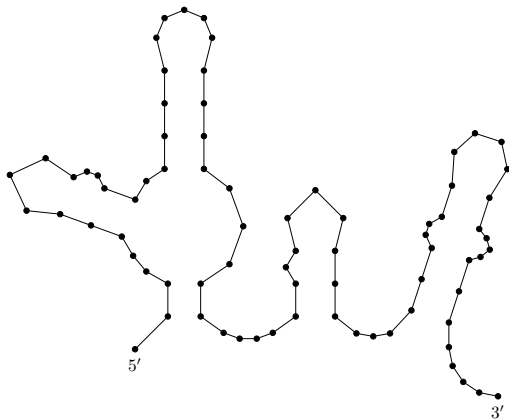
- Ribonucleic acid (RNA) is single stranded molecule comprising a phosphate-sugar backbone and to each sugar a base is attached.
- Bases: Adenine, Guanine, Cytosine, Uracil
- Strand orientation: 5' to 3'.
- Bases may form hydrogen bonds. Watson-Crick: A-U, G-C. Wobble: G-U
- Many functions essential to life, e.g. gene expression and cellular communication.



# RNA Structures

## Primary Structure

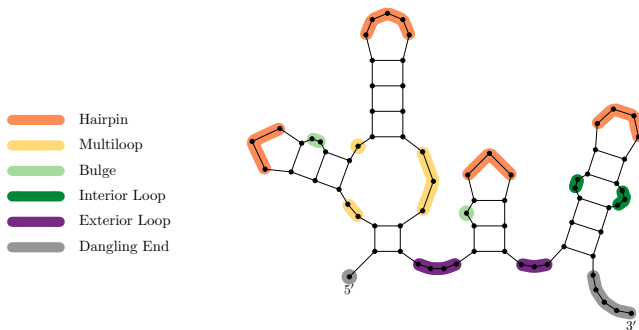
The *primary structure* can be thought of as the sequence of bases.



# RNA Structures

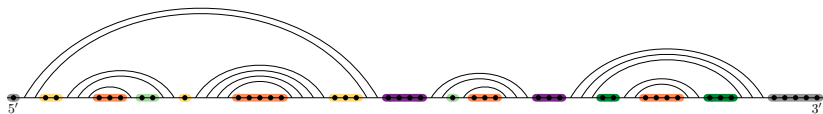
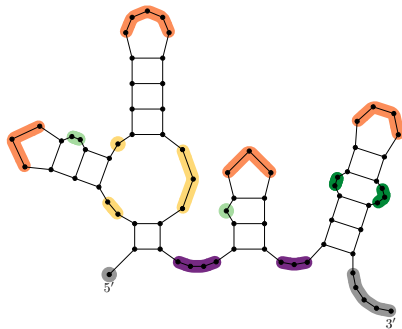
## Secondary Structure

RNA may coil in space, resulting in complementary bases bonding. With base pairing, we get higher order structure.

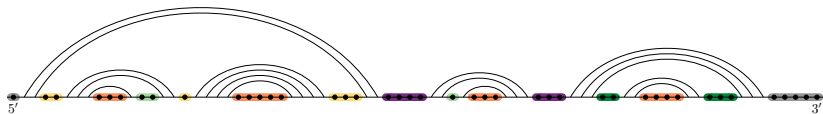


# RNA Structures

- Hairpin
- Multiloop
- Bulge
- Interior Loop
- Exterior Loop
- Dangling End



# RNA Structures



## Linear Chord Diagram

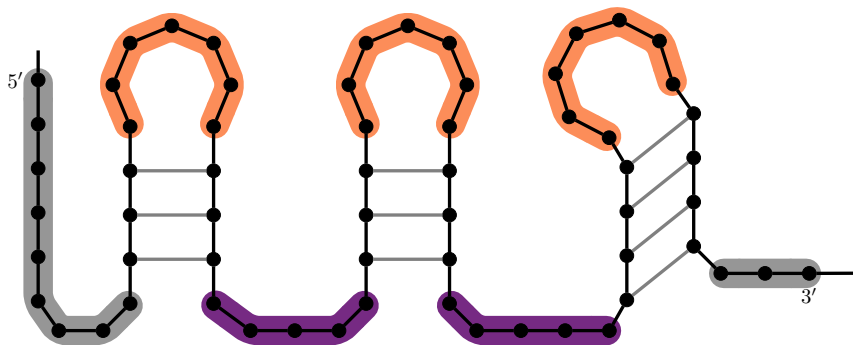
A *linear chord diagram* of size  $n$  is a graph with  $n$  points on an oriented line along with a matching of the points.

RNA sequences correspond to linear chord diagrams: each point on the line is a base in the base sequence and chords are edges in the matching.

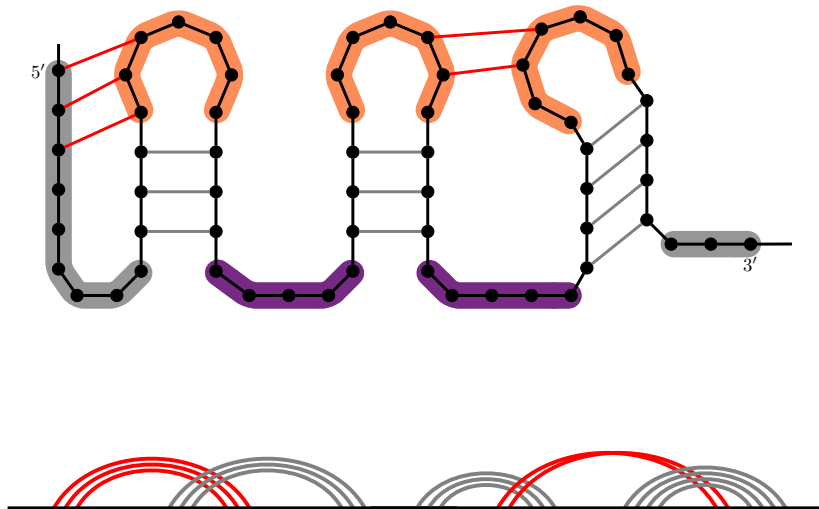
# RNA Structures – Pseudoknots

## A Different Structure

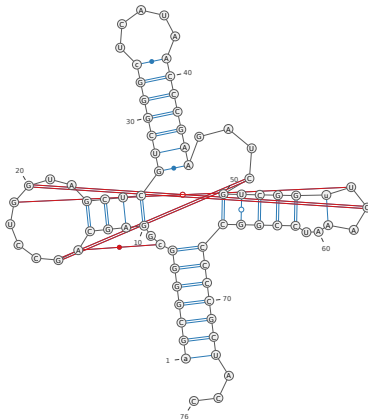
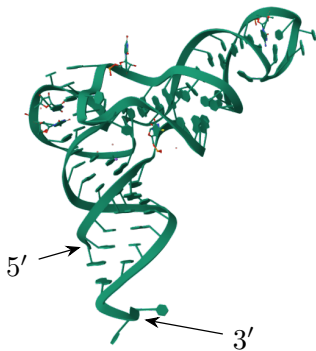
RNA may coil in space, resulting in complementary bases bonding. With base pairing, we get higher order structure. A pseudoknot arises when chords cross.



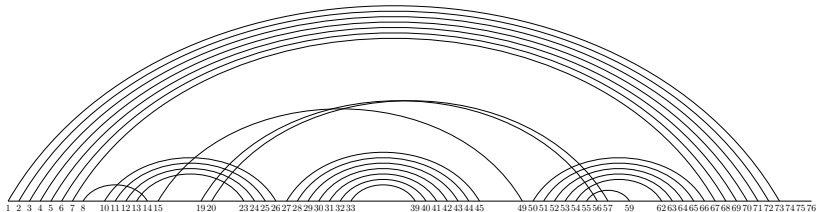
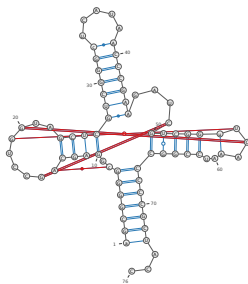
# RNA Structures – Pseudoknots



# Real Example (RCSB PDB/bpRNA-1m)



# Real Example (RCSB PDB/bpRNA-1m)





# The Goals

Our goals are to...

# The Goals

Our goals are to...

- Quantify pseudoknot complexity.

# The Goals

Our goals are to...

- Quantify pseudoknot complexity.
  - What is “a” pseudoknot?
  - How many pseudoknots does a structure have? (Is there a measure pseudoknottedness?)

# The Goals

Our goals are to...

- Quantify pseudoknot complexity.
  - What is “a” pseudoknot?
  - How many pseudoknots does a structure have? (Is there a measure pseudoknottedness?)
- Characterize pseudoknots by identifying the underlying secondary structure.

# The Goals

Our goals are to...

- Quantify pseudoknot complexity.
  - What is “a” pseudoknot?
  - How many pseudoknots does a structure have? (Is there a measure pseudoknottedness?)
- Characterize pseudoknots by identifying the underlying secondary structure.
  - We do not know the underlying secondary structure of a given RNA structure with pseudoknots.
  - How do we remove pseudoknots to analyze the “crossingless” structure?

# The Goals

Our goals are to...

- Quantify pseudoknot complexity.
  - What is “a” pseudoknot?
  - How many pseudoknots does a structure have? (Is there a measure pseudoknottedness?)
- Characterize pseudoknots by identifying the underlying secondary structure.
  - We do not know the underlying secondary structure of a given RNA structure with pseudoknots.
  - How do we remove pseudoknots to analyze the “crossingless” structure?

We will use... chord diagrams, intersection graphs, and graph theoretic invariants.

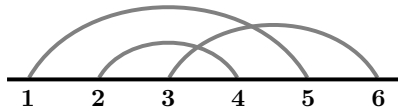
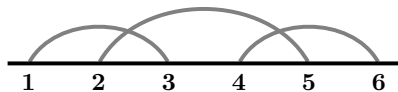
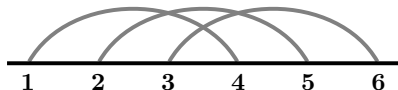
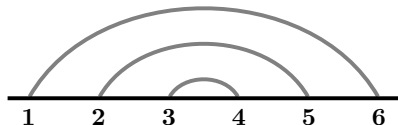
# Nestings, Crossings, Segments, and the Intersection Graph

## Nestings and Crossings

For any two chords  $c_1$  and  $c_2$ , there are three possibilities:

- 1  $c_1$  and  $c_2$  form a *crossing*:  $\ell_1 < \ell_2 < r_1 < r_2$ .
- 2  $c_1$  and  $c_2$  form a *nesting*:  $\ell_1 < \ell_2 < r_2 < r_1$ .
- 3  $c_1$  and  $c_2$  are *independent*:  $\ell_1 < r_1 < \ell_2 < r_2$ .

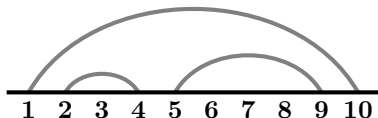
(Can be extended to  $k$ -crossings and  $k$ -nestings.)



# Nestings, Crossings, Segments, and the Intersection Graph

## Chord Obstructed

Let  $c$  and  $c'$  be chords and let  $S = [\ell, \ell'] \cup [r, r']$ , i.e.  $S$  can be thought of as the set of bases between the left endpoints and right endpoints of  $c$  and  $c'$ . We say  $c$  and  $c'$  are *chord obstructed* if there is some chord  $c''$  such that either  $\ell'' \in S$  or  $r'' \in S$ .





# Nestings, Crossings, Segments, and the Intersection Graph

## Segments

A *segment*  $S$  of a chord diagram is a maximal nonempty set of chords  $\{c_1, c_2, \dots, c_k\}$  forming a  $k$ -nesting such that  $c_i$  and  $c_{i+1}$  are not chord obstructed for  $1 \leq i \leq k - 1$ .

# Nestings, Crossings, Segments, and the Intersection Graph

## Segments

A *segment*  $S$  of a chord diagram is a maximal nonempty set of chords  $\{c_1, c_2, \dots, c_k\}$  forming a  $k$ -nesting such that  $c_i$  and  $c_{i+1}$  are not chord obstructed for  $1 \leq i \leq k - 1$ .

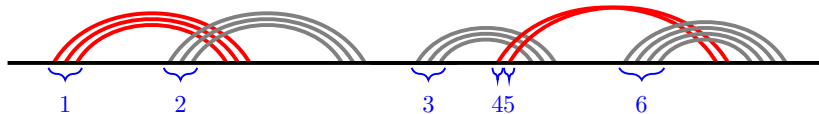
The set of segments partitions the set of chords, and two segments can cross in a natural way.

# Nestings, Crossings, Segments, and the Intersection Graph

## Segments

A *segment*  $S$  of a chord diagram is a maximal nonempty set of chords  $\{c_1, c_2, \dots, c_k\}$  forming a  $k$ -nesting such that  $c_i$  and  $c_{i+1}$  are not chord obstructed for  $1 \leq i \leq k - 1$ .

The set of segments partitions the set of chords, and two segments can cross in a natural way.

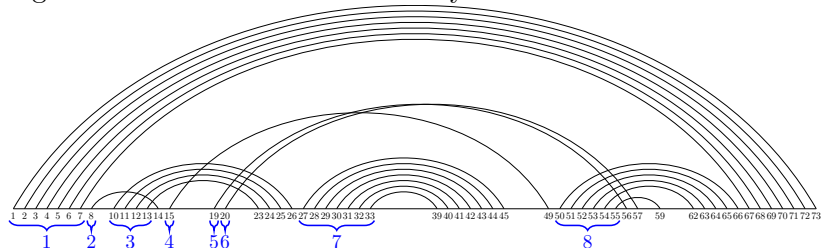


# Nestings, Crossings, Segments, and the Intersection Graph

## Segments

A *segment*  $S$  of a chord diagram is a maximal nonempty set of chords  $\{c_1, c_2, \dots, c_k\}$  forming a  $k$ -nesting such that  $c_i$  and  $c_{i+1}$  are not chord obstructed for  $1 \leq i \leq k - 1$ .

The set of segments partitions the set of chords, and two segments can cross in a natural way.



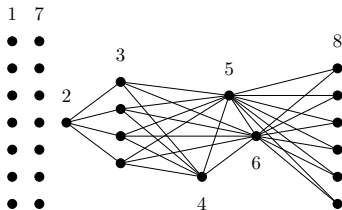
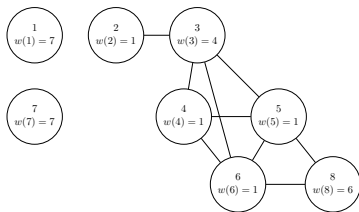
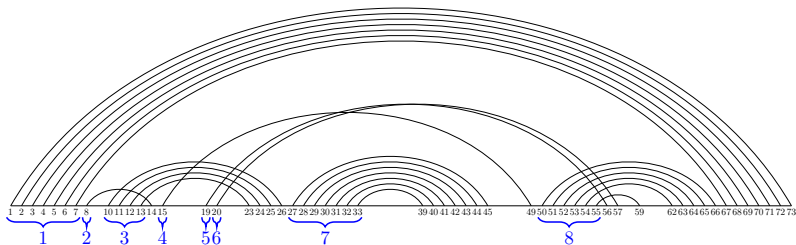
# Segments Example

## Intersection and Segment Graph

The *intersection graph*  $G$  of a chord diagram  $D$  is the graph whose vertices are the chords of  $D$ , and two vertices in  $G$  are adjacent if their corresponding chords in  $D$  form a crossing.

The *segment graph* of a chord diagram is similar: vertices are segments and weighted, and two vertices are adjacent if their segments cross.

# Segments Example



# bpRNA – Quantifying Pseudoknots

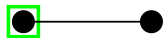
## Vertex Covers

A *vertex cover* of a graph  $G$  is a set  $A \subseteq V(G)$  such that for every edge  $xy \in E(G)$ , either  $x \in A$  or  $y \in A$ . The *vertex cover number*  $\beta(G)$  is the size of a minimum vertex cover. The *weight* of a vertex cover  $A$  in a vertex-weighted graph  $G$  is  $\sum_{v \in A} w(v)$ .

# bpRNA – Quantifying Pseudoknots

## Vertex Covers

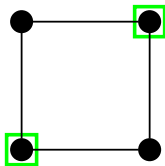
A *vertex cover* of a graph  $G$  is a set  $A \subseteq V(G)$  such that for every edge  $xy \in E(G)$ , either  $x \in A$  or  $y \in A$ . The *vertex cover number*  $\beta(G)$  is the size of a minimum vertex cover. The *weight* of a vertex cover  $A$  in a vertex-weighted graph  $G$  is  $\sum_{v \in A} w(v)$ .



$$\beta = 1$$



$$\beta = 1$$



$$\beta = 2$$



# bpRNA – Quantifying Pseudoknots

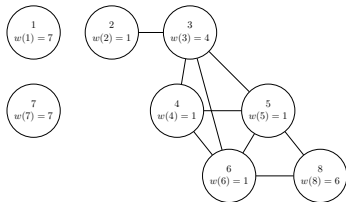
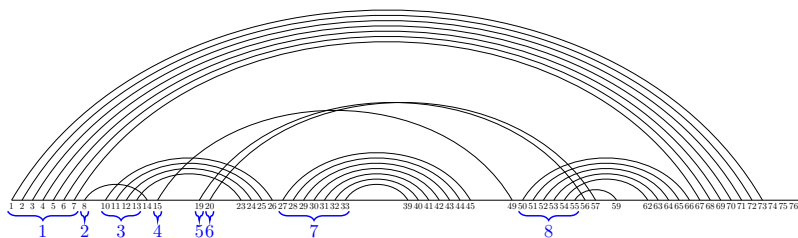
## Vertex Covers

A *vertex cover* of a graph  $G$  is a set  $A \subseteq V(G)$  such that for every edge  $xy \in E(G)$ , either  $x \in A$  or  $y \in A$ . The *vertex cover number*  $\beta(G)$  is the size of a minimum vertex cover. The *weight* of a vertex cover  $A$  in a vertex-weighted graph  $G$  is  $\sum_{v \in A} w(v)$ .

## Pseudoknotted Structures (Danaee et al. '18)

Let  $D$  be the chord diagram of an RNA structure and  $G_S$  be the corresponding segment graph. Then the number of pseudoknots in  $D$  is the minimum cardinality over all vertex covers of minimum weight of  $G_S$ . A vertex in such a vertex cover corresponds to a pseudoknot.

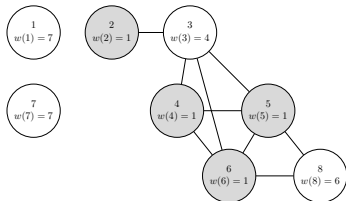
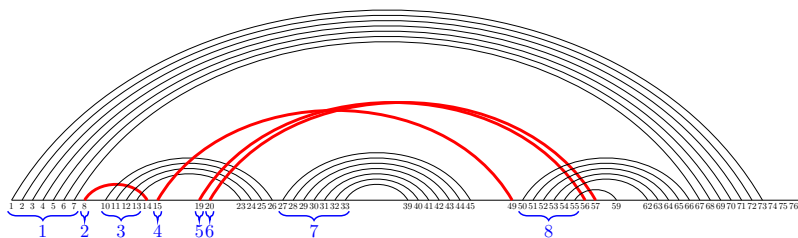
# bpRNA – Quantifying Pseudoknots - Example



Minimum Vertex Cover:  $\{3, 5, 6\}$

Minimum Weight Vertex Cover:  $\{2, 4, 5, 6\}$

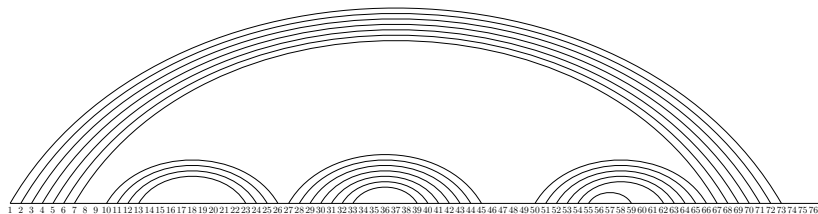
# bpRNA – Quantifying Pseudoknots - Example



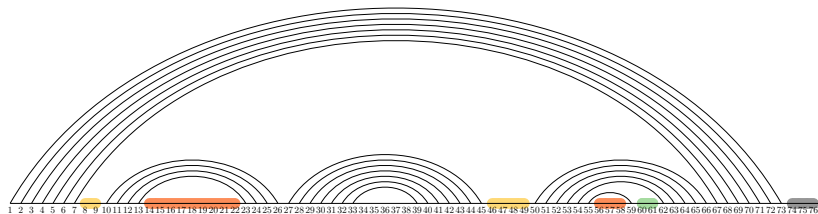
Minimum Vertex Cover:  $\{3, 5, 6\}$

Minimum Weight Vertex Cover:  $\{2, 4, 5, 6\}$

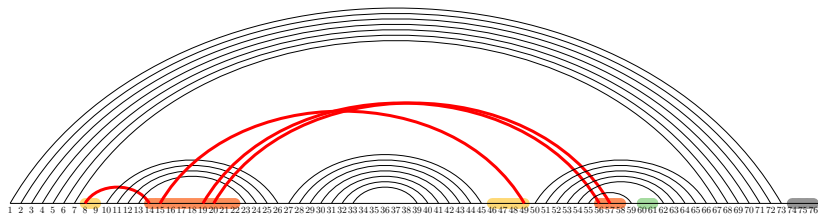
# bpRNA – Quantifying Pseudoknots - Example



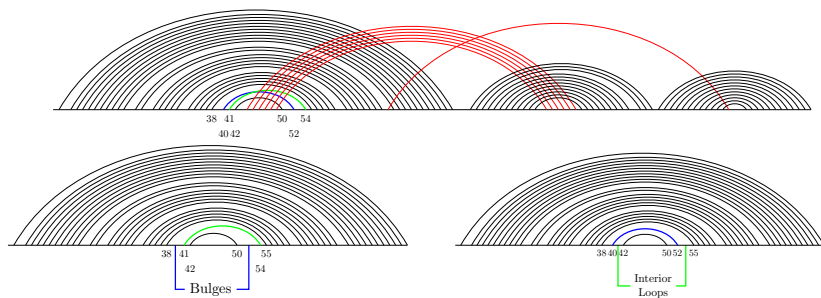
# bpRNA – Quantifying Pseudoknots - Example



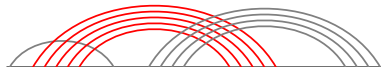
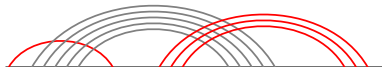
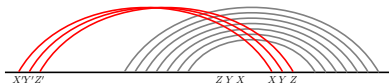
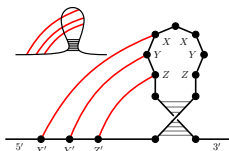
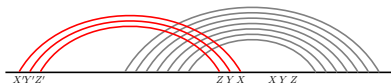
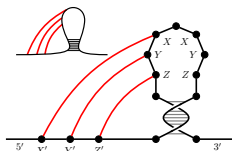
# bpRNA – Quantifying Pseudoknots - Example



# bpRNA – Quantifying Pseudoknots - Choice Example



# bpRNA - Discrepancies Due to $k$ -crossings and Weights





# Chord Distance and $\tau$ -Segments

## Chord Distance

Let  $c_1 = (\ell_i, r_i)$  and  $c_2 = (\ell_j, r_j)$  be two chords. We define a chord distance  $d(c_i, c_j) = \max\{|\ell_i - \ell_j|, |r_i - r_j|\}$

# Chord Distance and $\tau$ -Segments

## Chord Distance

Let  $c_1 = (\ell_i, r_i)$  and  $c_2 = (\ell_j, r_j)$  be two chords. We define a chord distance  $d(c_i, c_j) = \max\{|\ell_i - \ell_j|, |r_i - r_j|\}$

## $\tau$ -Segment

A  $\tau$ -segment of a chord diagram is a maximal nonempty set of chords  $\{c_1, c_2, \dots, c_k\}$  forming a  $k$ -nesting or  $k$ -crossing such that  $d(c_i, c_{i+1}) \leq \tau$  and  $c_i$  and  $c_{i+1}$  are not chord obstructed. (When  $\tau = 0$ , we use bpRNA segments).

# Chord Distance and $\tau$ -Segments

## Chord Distance

Let  $c_1 = (\ell_i, r_i)$  and  $c_2 = (\ell_j, r_j)$  be two chords. We define a chord distance  $d(c_i, c_j) = \max\{|\ell_i - \ell_j|, |r_i - r_j|\}$

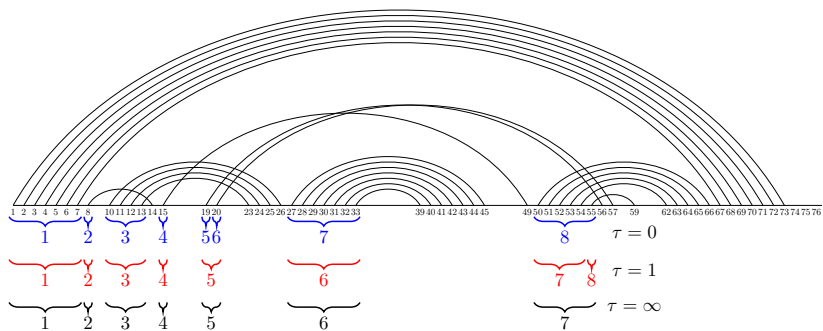
## $\tau$ -Segment

A  $\tau$ -segment of a chord diagram is a maximal nonempty set of chords  $\{c_1, c_2, \dots, c_k\}$  forming a  $k$ -nesting or  $k$ -crossing such that  $d(c_i, c_{i+1}) \leq \tau$  and  $c_i$  and  $c_{i+1}$  are not chord obstructed. (When  $\tau = 0$ , we use bpRNA segments).

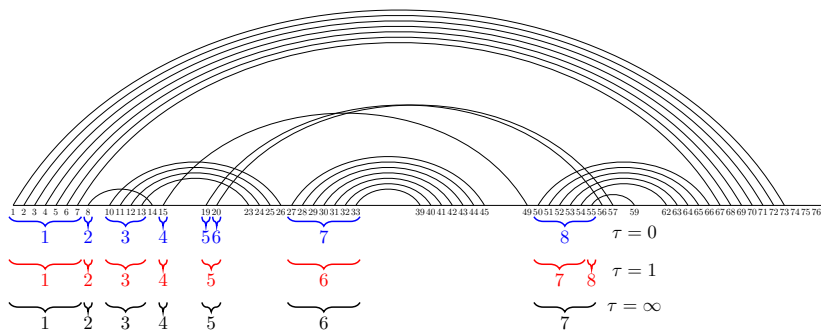
## Augmented Segments

An augmented segment  $S$  of a chord diagram is a maximal nonempty set of chords  $\{c_1, c_2, \dots, c_k\}$  forming an unobstructed  $k$ -nesting or an unobstructed  $k$ -crossing. ( $\tau = \infty$ ).

# $\tau$ -Segment Partitions



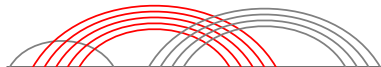
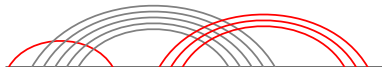
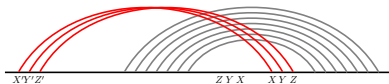
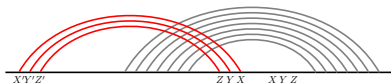
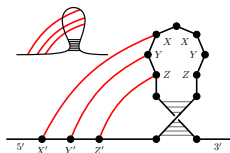
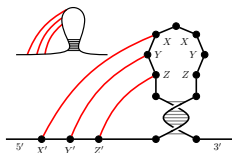
# $\tau$ -Segment Partitions



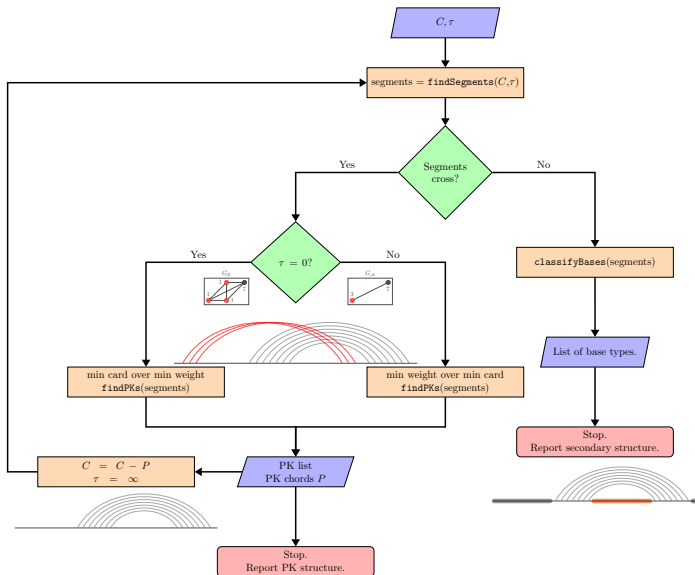
## Pseudoknotted Structures

Let  $D$  be the chord diagram of an RNA structure and  $G_\tau$  be the corresponding  $\tau$ -segment graph. The number of pseudoknots is the minimum cardinality of a vertex cover of  $G_\tau$ , i.e.  $\beta(G_\tau)$ .

# $\tau$ -Segment Partitions



# The Workflow

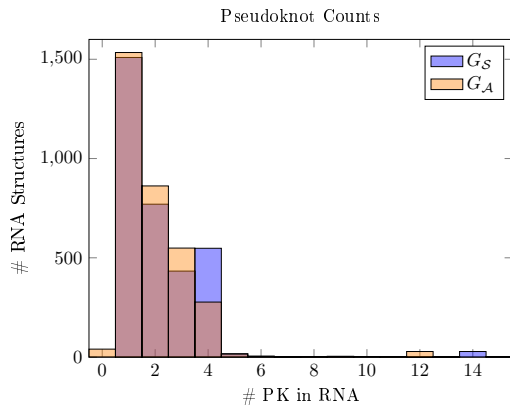


# Analysis

- The genus of a chord diagram is  $\text{rank}_2(A)$  where  $A$  is the adjacency matrix of the intersection graph.
- Genus bubble plot
- $\tau$ -max

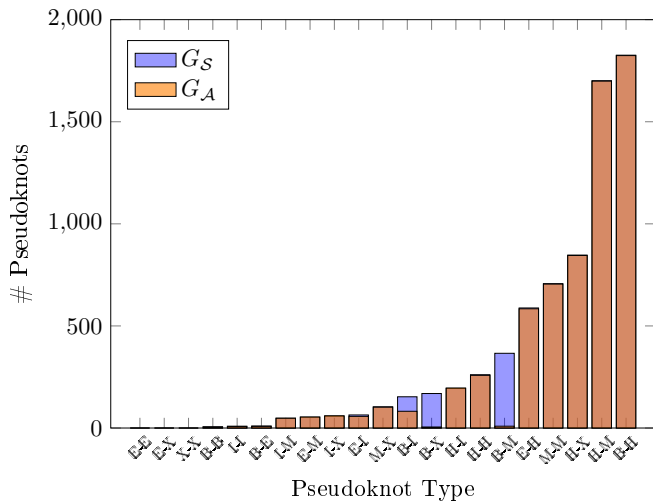


# Analysis – Number of Pseudoknots



# PK	$G_S$	$G_A$
0	0	40
1	1509	1534
2	770	862
3	433	549
4	548	277
5	17	15
6	3	5
7	0	1
8	2	2
9	2	4
10	2	0
11	2	2
12	1	28
13	2	1
14	28	0
15	1	0

# Analysis – Pseudoknot Types



# Analysis – Genus

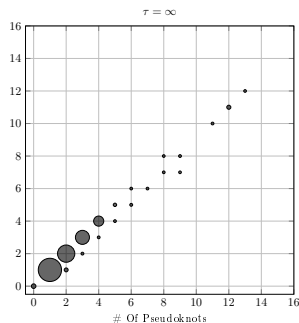
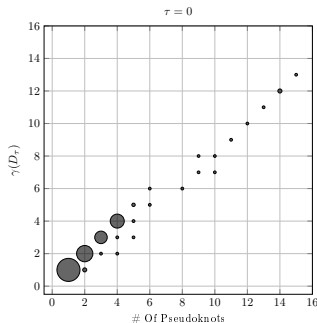
Genus (Bon et al. '08)

The *genus* of a chord diagram  $D$  is  $\gamma(D) = (1/2) \text{rank}_2(A)$ , where  $A$  is the adjacency matrix of the intersection graph of  $D$ .

# Analysis – Genus

## Genus (Bon et al. '08)

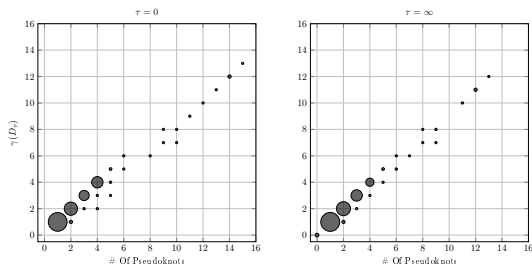
The *genus* of a chord diagram  $D$  is  $\gamma(D) = (1/2) \text{rank}_2(A)$ , where  $A$  is the adjacency matrix of the intersection graph of  $D$ .



# Analysis – Genus

## Genus (Bon et al. '08)

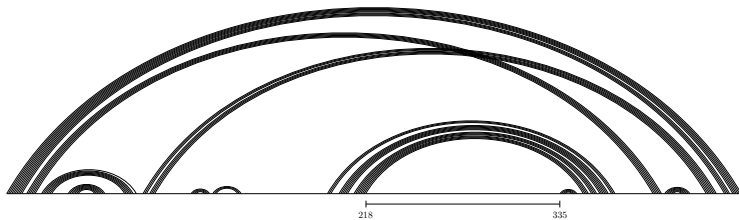
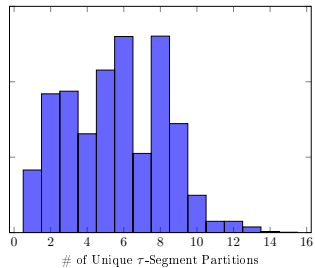
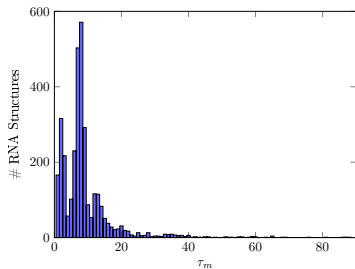
The *genus* of a chord diagram  $D$  is  $\gamma(D) = (1/2) \text{rank}_2(A)$ , where  $A$  is the adjacency matrix of the intersection graph of  $D$ .



## Theorem (Ibrahim '24)

If  $G$  is acyclic with adjacency matrix  $A$ , then  $\beta(G) = (1/2) \text{rank}_2(A)$ .

# Analysis – $\tau_m$



# Outline

- 1 Modeling RNA Via Chord Diagrams and Intersection Graphs
- 2 2-Neighbor Bootstrap Percolation

# Bootstrap Percolation Process

## $r$ -Neighbor Bootstrap Percolation

- Begin with an initial set of ‘infected’ vertices,  $A_0 \subseteq V(G)$ .



# Bootstrap Percolation Process

## $r$ -Neighbor Bootstrap Percolation

- Begin with an initial set of ‘infected’ vertices,  $A_0 \subseteq V(G)$ .
- In each round, an uninfected vertex  $v$  becomes infected if  $v$  is adjacent to at least  $r$  infected vertices.

# Bootstrap Percolation Process

## $r$ -Neighbor Bootstrap Percolation

- Begin with an initial set of ‘infected’ vertices,  $A_0 \subseteq V(G)$ .
- In each round, an uninfected vertex  $v$  becomes infected if  $v$  is adjacent to at least  $r$  infected vertices.
- Once infected, vertices remain infected.

# Bootstrap Percolation Process

## $r$ -Neighbor Bootstrap Percolation

- Begin with an initial set of ‘infected’ vertices,  $A_0 \subseteq V(G)$ .
- In each round, an uninfected vertex  $v$  becomes infected if  $v$  is adjacent to at least  $r$  infected vertices.
- Once infected, vertices remain infected.
- The process is finite – the *closure* of  $A_0$ , denoted  $\langle A_0 \rangle$ , is the set of infected vertices when the process finishes.

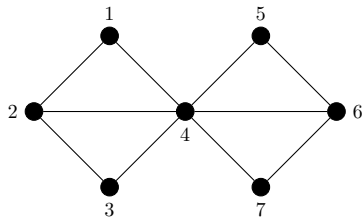
# Bootstrap Percolation Process

## $r$ -Neighbor Bootstrap Percolation

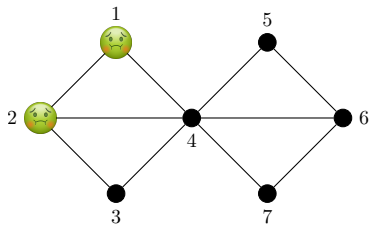
- Begin with an initial set of ‘infected’ vertices,  $A_0 \subseteq V(G)$ .
- In each round, an uninfected vertex  $v$  becomes infected if  $v$  is adjacent to at least  $r$  infected vertices.
- Once infected, vertices remain infected.
- The process is finite – the *closure* of  $A_0$ , denoted  $\langle A_0 \rangle$ , is the set of infected vertices when the process finishes.

Cellular automaton – Conway’s Game of Life

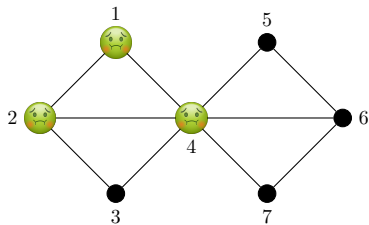
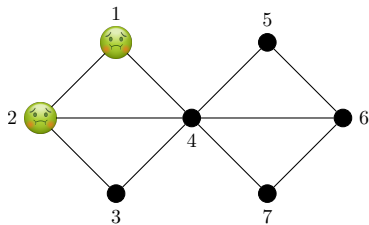
## Example 1 ( $r = 2$ )



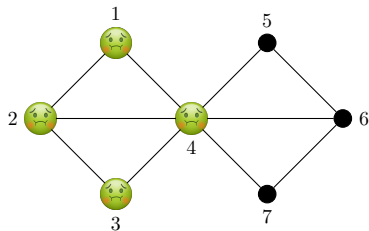
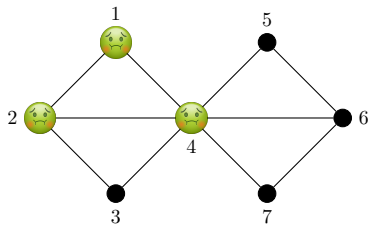
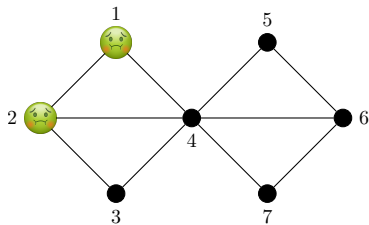
## Example 1 ( $r = 2$ )



# Example 1 ( $r = 2$ )



# Example 1 ( $r = 2$ )

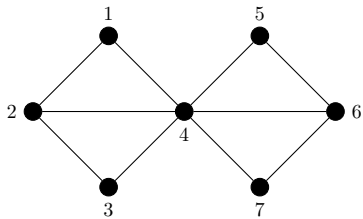


$$A_0 = \{1, 2\}$$

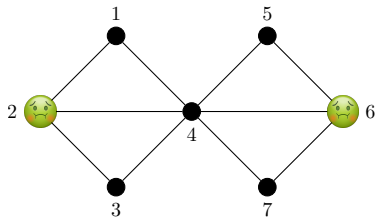
$$\langle A_0 \rangle = \{1, 2, 3, 4\}$$



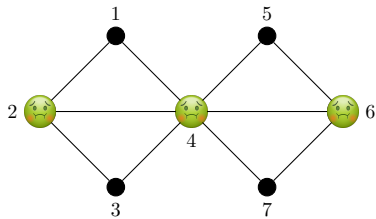
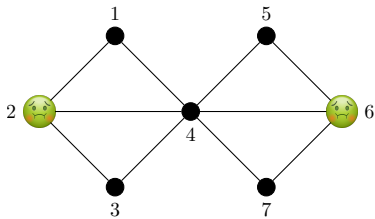
## Example 2 ( $r = 2$ )



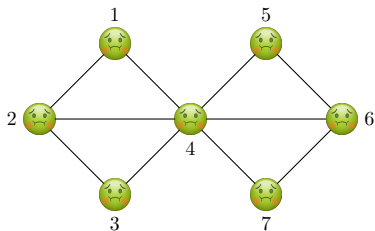
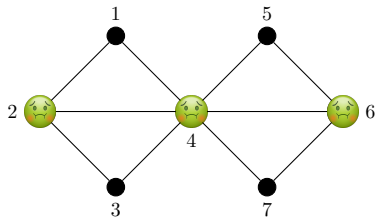
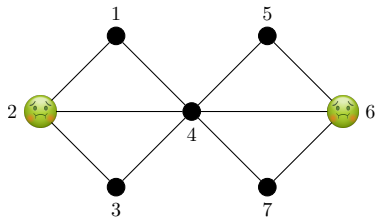
## Example 2 ( $r = 2$ )



## Example 2 ( $r = 2$ )



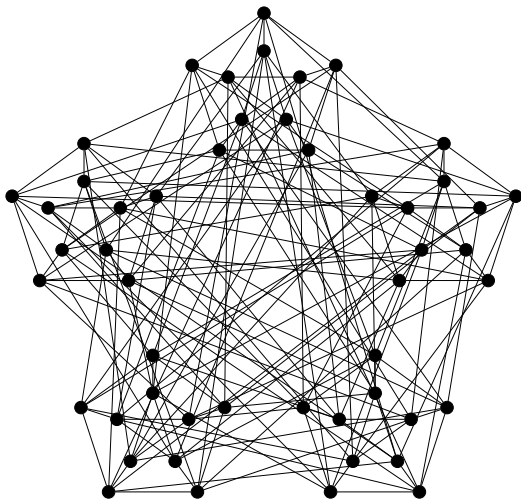
## Example 2 ( $r = 2$ )



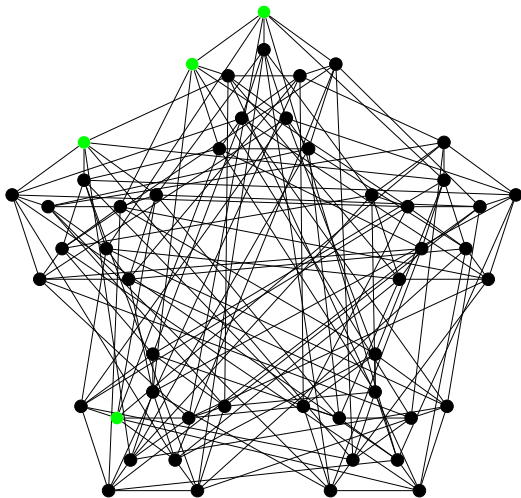
$$A_0 = \{2, 6\}$$

$$\langle A_0 \rangle = V(G)$$

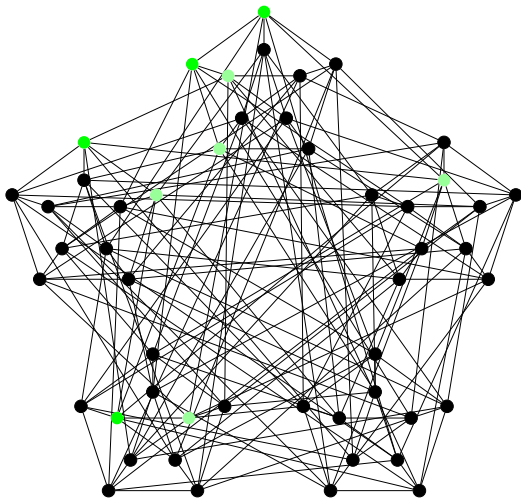
### Example 3 ( $r = 2$ ) (Hoffman-Singleton)



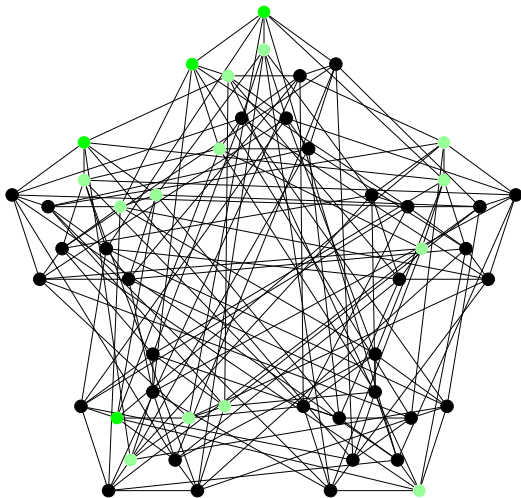
### Example 3 ( $r = 2$ ) (Hoffman-Singleton)



### Example 3 ( $r = 2$ ) (Hoffman-Singleton)

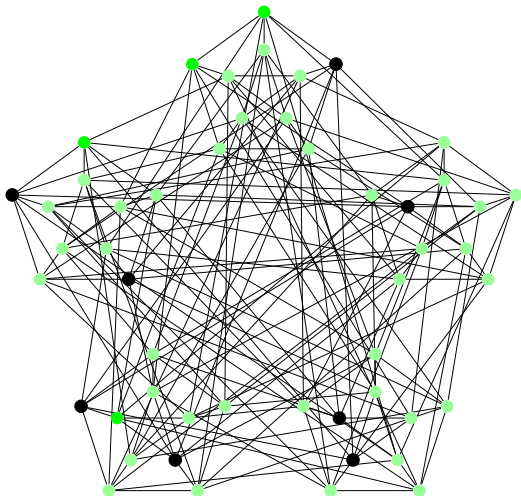


### Example 3 ( $r = 2$ ) (Hoffman-Singleton)

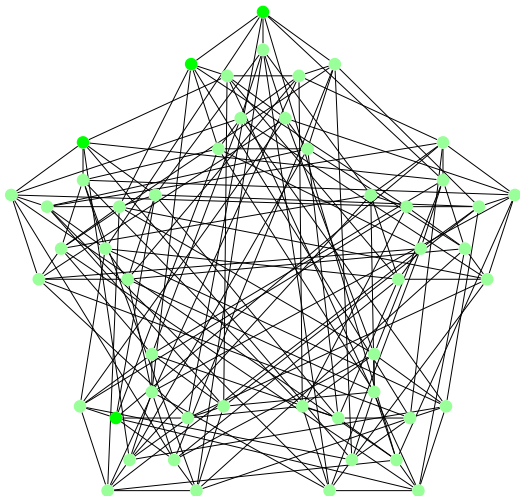




### Example 3 ( $r = 2$ ) (Hoffman-Singleton)



### Example 3 ( $r = 2$ ) (Hoffman-Singleton)



# Bootstrap Percolation Process

## $r$ -Neighbor Bootstrap Percolation

- Begin with an initial set of ‘infected’ vertices,  $A_0 \subseteq V(G)$ .
- In each round, an uninfected vertex  $v$  becomes infected if  $v$  is adjacent to at least  $r$  infected vertices.
- Once infected, vertices remain infected.
- The process is finite – the *closure* of  $A_0$ , denoted  $\langle A_0 \rangle$ , is the set of infected vertices when the process finishes.

# Bootstrap Percolation Process

## $r$ -Neighbor Bootstrap Percolation

- Begin with an initial set of ‘infected’ vertices,  $A_0 \subseteq V(G)$ .
- In each round, an uninfected vertex  $v$  becomes infected if  $v$  is adjacent to at least  $r$  infected vertices.
- Once infected, vertices remain infected.
- The process is finite – the *closure* of  $A_0$ , denoted  $\langle A_0 \rangle$ , is the set of infected vertices when the process finishes.
- If  $\langle A_0 \rangle = V(G)$ , we say  $A_0$  *percolates*.

# Bootstrap Percolation Process

## $r$ -Neighbor Bootstrap Percolation

- Begin with an initial set of ‘infected’ vertices,  $A_0 \subseteq V(G)$ .
- In each round, an uninfected vertex  $v$  becomes infected if  $v$  is adjacent to at least  $r$  infected vertices.
- Once infected, vertices remain infected.
- The process is finite – the *closure* of  $A_0$ , denoted  $\langle A_0 \rangle$ , is the set of infected vertices when the process finishes.
- If  $\langle A_0 \rangle = V(G)$ , we say  $A_0$  *percolates*.

## Choosing $A_0$

Early models incorporate randomness; initial infected vertices are selected with probability  $p$ .

# Bootstrap Percolation Process

## $r$ -Bootstrap-Good

- $m(G, r)$  : the minimum size of an  $r$ -percolating set of  $G$ .

# Bootstrap Percolation Process

## $r$ -Bootstrap-Good

- $m(G, r)$  : the minimum size of an  $r$ -percolating set of  $G$ .
  - If  $|V(G)| \leq r$  then  $m(G, r) = |V(G)|$ , otherwise  
 $r \leq m(G, r) \leq |V(G)|$ .

# Bootstrap Percolation Process

## $r$ -Bootstrap-Good

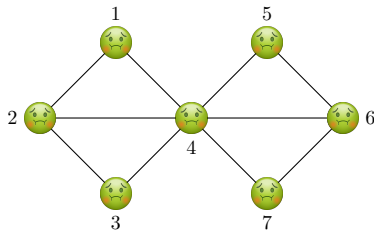
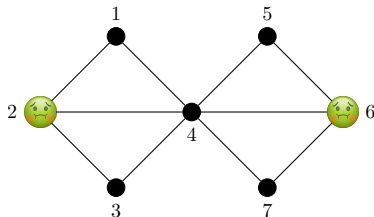
- $m(G, r)$  : the minimum size of an  $r$ -percolating set of  $G$ .
  - If  $|V(G)| \leq r$  then  $m(G, r) = |V(G)|$ , otherwise  
 $r \leq m(G, r) \leq |V(G)|$ .
- If  $m(G, r) = r$  then we say  $G$  is  $r$ -Bootstrap-Good, or  $r$ -BG.



# Bootstrap Percolation Process

## $r$ -Bootstrap-Good

- $m(G, r)$  : the minimum size of an  $r$ -percolating set of  $G$ .
  - If  $|V(G)| \leq r$  then  $m(G, r) = |V(G)|$ , otherwise  $r \leq m(G, r) \leq |V(G)|$ .
- If  $m(G, r) = r$  then we say  $G$  is  $r$ -Bootstrap-Good, or  $r$ -BG.



# Paths to Explore

Lots of questions!!

# Paths to Explore

Lots of questions!!

- What is the structure of  $r$ -BG graphs?
  - Necessary and sufficient conditions for when  $m(G, 2) = 2$ .

# Paths to Explore

Lots of questions!!

- What is the structure of  $r$ -BG graphs?
  - Necessary and sufficient conditions for when  $m(G, 2) = 2$ .
- Bounds on  $m(G, r)$  for particular graph classes.
  - For some class of graphs  $\mathcal{G}$ , is there a constant  $k$  such that for all  $G \in \mathcal{G}$ , we have  $m(G, 2) \leq k$ ?

# Paths to Explore

Lots of questions!!

- What is the structure of  $r$ -BG graphs?
  - Necessary and sufficient conditions for when  $m(G, 2) = 2$ .
- Bounds on  $m(G, r)$  for particular graph classes.
  - For some class of graphs  $\mathcal{G}$ , is there a constant  $k$  such that for all  $G \in \mathcal{G}$ , we have  $m(G, 2) \leq k$ ?
- What are the minimum and maximum number of rounds to percolate?
  - Looking at all percolating sets of a fixed size (minimum), which set takes the most rounds to percolate? The fewest?

# Paths to Explore

Lots of questions!!

- What is the structure of  $r$ -BG graphs?
  - Necessary and sufficient conditions for when  $m(G, 2) = 2$ .
- Bounds on  $m(G, r)$  for particular graph classes.
  - For some class of graphs  $\mathcal{G}$ , is there a constant  $k$  such that for all  $G \in \mathcal{G}$ , we have  $m(G, 2) \leq k$ ?
- What are the minimum and maximum number of rounds to percolate?
  - Looking at all percolating sets of a fixed size (minimum), which set takes the most rounds to percolate? The fewest?

# Our Graph Class - Diameter

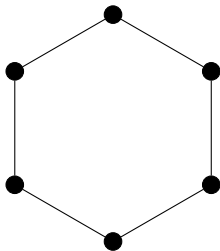
## Diameter

The *diameter* of a graph  $G$  is  $\max_{u,v \in V(G)} d(u,v)$ , where  $d(u,v)$  is the length of a shortest path from  $u$  to  $v$ .

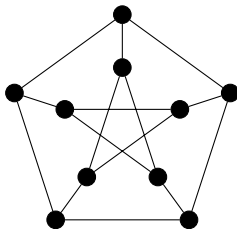
# Our Graph Class - Diameter

## Diameter

The *diameter* of a graph  $G$  is  $\max_{u,v \in V(G)} d(u,v)$ , where  $d(u,v)$  is the length of a shortest path from  $u$  to  $v$ .



$$\text{diam}(C_6) = 3$$



$$\text{diam}(\text{Pete}) = 2$$



# Our Graph Class - Connectivity

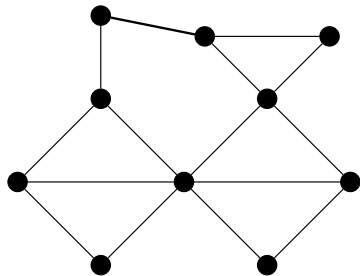
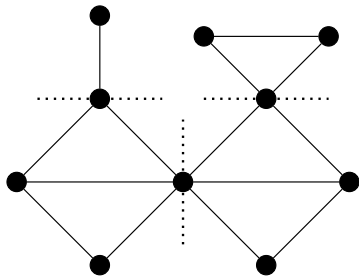
## Connectivity

A connected graph  $G$  is 2-connected if for every vertex  $x \in V(G)$ , the graph  $G - x$  is connected. (No cut-vertex in  $G$ ).

# Our Graph Class - Connectivity

## Connectivity

A connected graph  $G$  is 2-connected if for every vertex  $x \in V(G)$ , the graph  $G - x$  is connected. (No cut-vertex in  $G$ ).

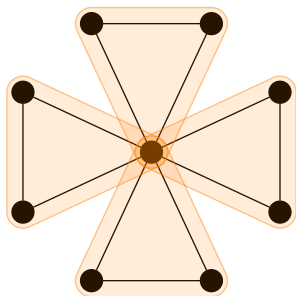


# Our Graph Class - Diameter 2 and 2-connected

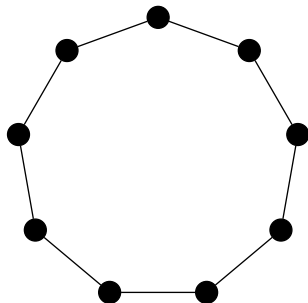
Graphs of interest: Diameter 2 and 2-connected

# Our Graph Class - Diameter 2 and 2-connected

Graphs of interest: Diameter 2 and 2-connected



Diameter 2, ~~2-connected~~



~~Diameter 2~~, 2-connected

# A 2-BG Theorem

Theorem (Ibrahim, LaFayette, McCall '24)

*Let  $G$  be a 2-connected graph with diameter 2. If  $G$  is  $C_5$ -free, then  $G$  is 2-BG.*

# A 2-BG Theorem

Theorem (Ibrahim, LaFayette, McCall '24)

*Let  $G$  be a 2-connected graph with diameter 2. If  $G$  is  $C_5$ -free, then  $G$  is 2-BG.*

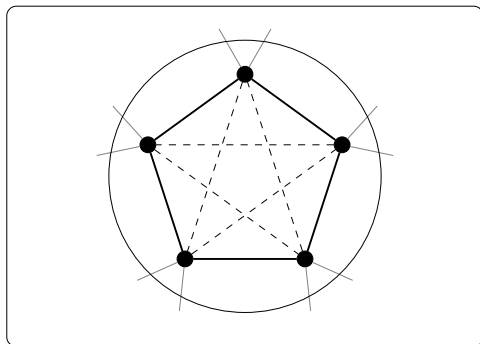
By  $C_5$ -free, we mean...

# A 2-BG Theorem

Theorem (Ibrahim, LaFayette, McCall '24)

*Let  $G$  be a 2-connected graph with diameter 2. If  $G$  is  $C_5$ -free, then  $G$  is 2-BG.*

By  $C_5$ -free, we mean...



$G$

# A Constant Upper Bound

Theorem (Cappelle et al. '22)

*Let  $G$  be a 2-connected graph with diameter 2. If  $G$  is  $C_6$ -free, then  $m(G, 2) \leq 4$ .*



# A Constant Upper Bound

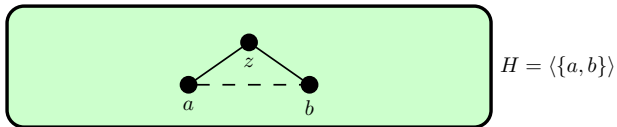
## Theorem (Cappelle et al. '22)

*Let  $G$  be a 2-connected graph with diameter 2. If  $G$  is  $C_6$ -free, then  $m(G, 2) \leq 4$ .*

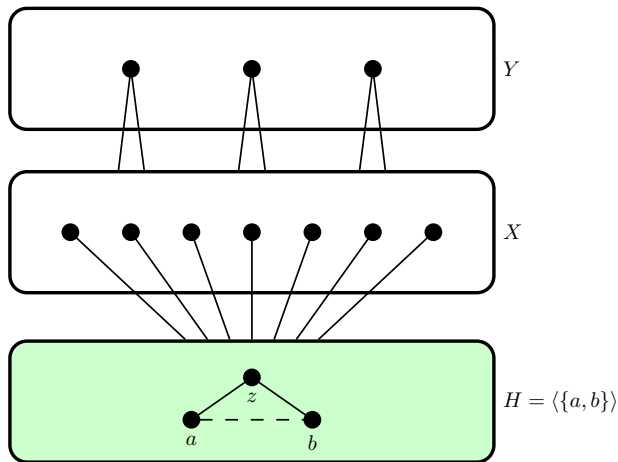
## Theorem (Ibrahim '24)

*Let  $G$  be a 2-connected graph with diameter 2. If  $G$  is  $C_6$ -free or  $C_7$ -free, then  $m(G, 2) \leq 3$ .*

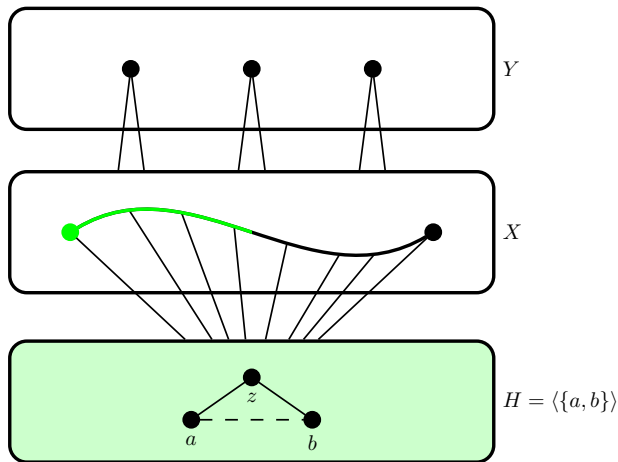
A Constant Upper Bound:  $C_6$ -free  $\implies m(G, 2) \leq 3$



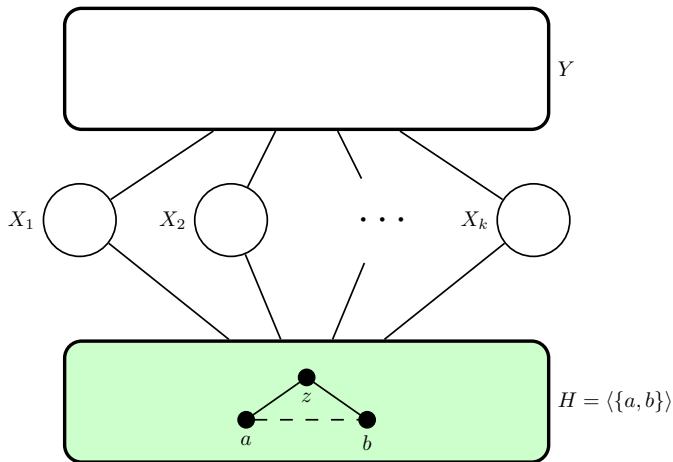
A Constant Upper Bound:  $C_6$ -free  $\implies m(G, 2) \leq 3$



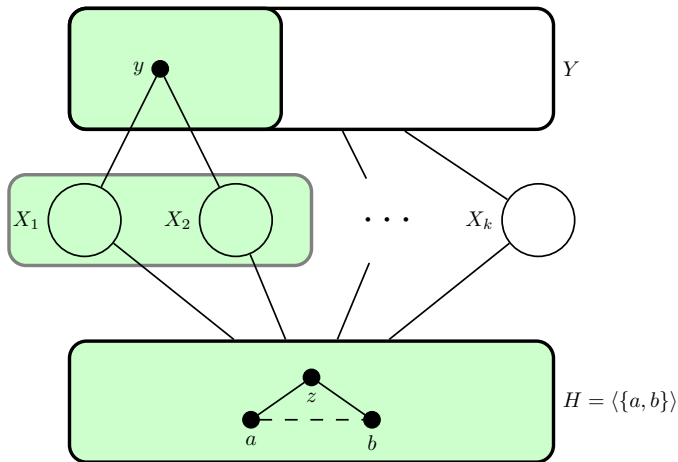
A Constant Upper Bound:  $C_6$ -free  $\implies m(G, 2) \leq 3$



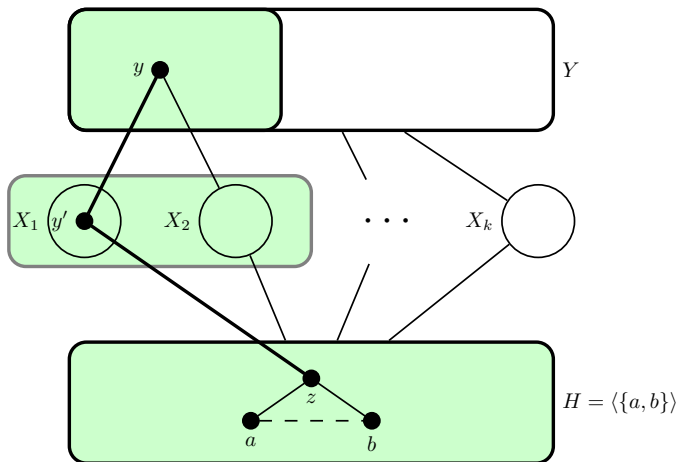
A Constant Upper Bound:  $C_6$ -free  $\implies m(G, 2) \leq 3$



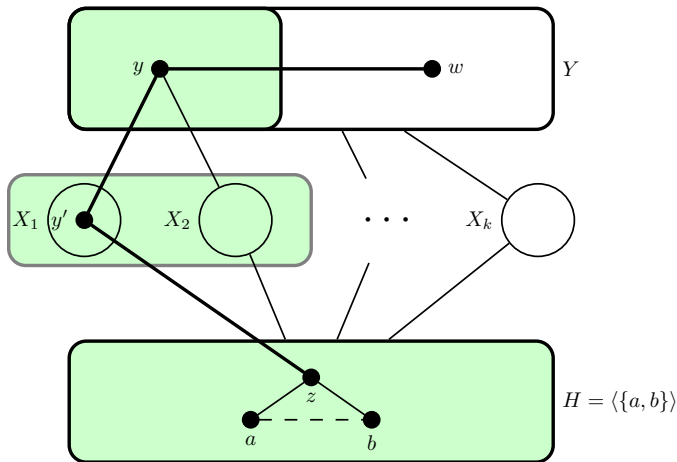
A Constant Upper Bound:  $C_6$ -free  $\implies m(G, 2) \leq 3$



A Constant Upper Bound:  $C_6$ -free  $\implies m(G, 2) \leq 3$

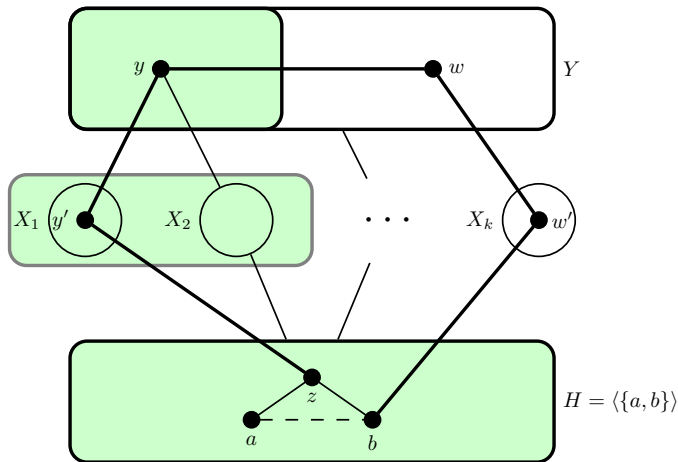


A Constant Upper Bound:  $C_6$ -free  $\implies m(G, 2) \leq 3$





A Constant Upper Bound:  $C_6$ -free  $\implies m(G, 2) \leq 3$



The End

Thank you!

# Extra

The next few slides are extra slides...

# A 2-BG Theorem (Generalization)

## Theorem (Ibrahim '24)

*Let  $G$  be a 2-connected graph with diameter 2. If  $G$  is  $C_k$ -free, then  $m(G, 2) \leq \lceil (k-3)/2 \rceil$ .*

