

E.HOP: A Bacterial Computer to Solve the Pancake Problem




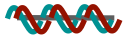












Davidson College iGEM Team

Lance Harden, Sabriya Rosemond,
Samantha Simpson, Erin Zwack

Faculty advisors: Malcolm Campbell,
Karmella Haynes, Laurie Heyer




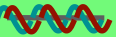
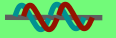









Basic Pancake Parts

| Name | Icon | Description |
|--------|---|----------------------------|
| J31009 |  | pSB1A7 (insulated plasmid) |
| J31000 |  | Hin Recombinase |
| J31001 |  | Hin Recombinase-LVA |
| J31011 |  | Recombination Enhancer |
| J44000 |  | HixC |
| J44001 |  | RBS reverse |
| J31003 |  | Kan ^R forward |
| J31002 |  | Kan ^R reverse |
| J31005 |  | Chl ^R forward |
| J31004 |  | Chl ^R reverse |
| J31007 |  | Tet ^R forward |
| J31006 |  | Tet ^R reverse |
| J44002 |  | pBAD reverse |
| J31011 |  | RFP and RBS reverse |

Davidson

Missouri
Western

Basic Pancake Parts

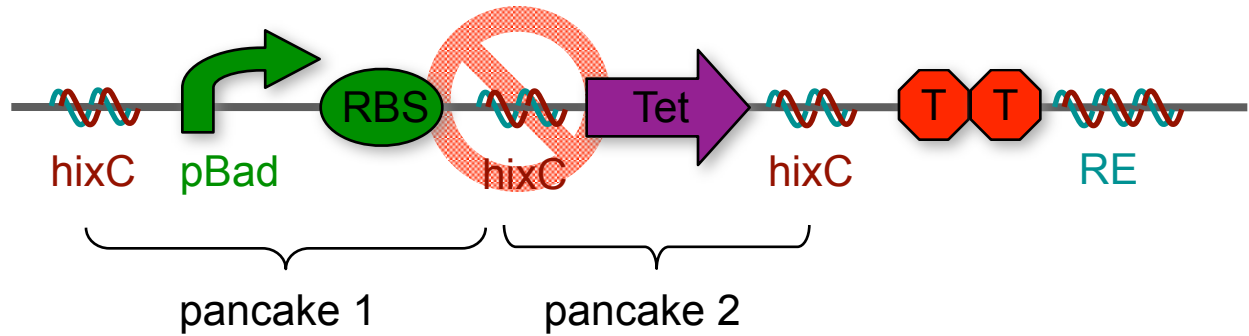
| Available | Tested | Included in Devices |
|-----------|---|----------------------------|
| J31009 |  | pSB1A7 (insulated plasmid) |
| J31000 |  | Hin Recombinase |
| J31001 |  | Hin Recombinase-LVA |
| J31011 |  | Recombination Enhancer |
| J44000 |  | HixC |
| J44001 |  | RBS reverse |
| J31003 |  | Kan ^R forward |
| J31002 |  | Kan ^R reverse |
| J31005 |  | Chl ^R forward |
| J31004 |  | Chl ^R reverse |
| J31007 |  | Tet ^R forward |
| J31006 |  | Tet ^R reverse |
| J44002 |  | pBAD reverse |
| J31011 |  | RFP and RBS reverse |

Davidson

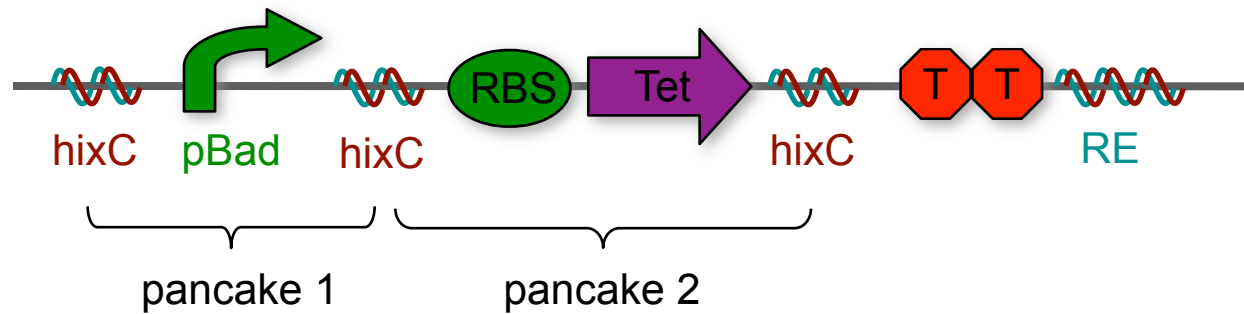
Missouri
Western

Define a Biological Pancake

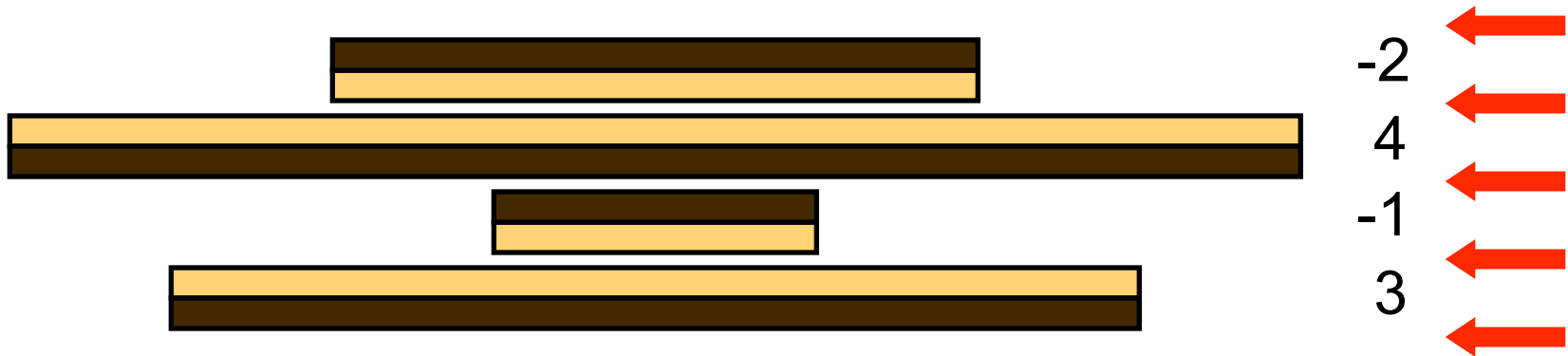
Nonfunctional



Functional



Modeling Pancake Flipping



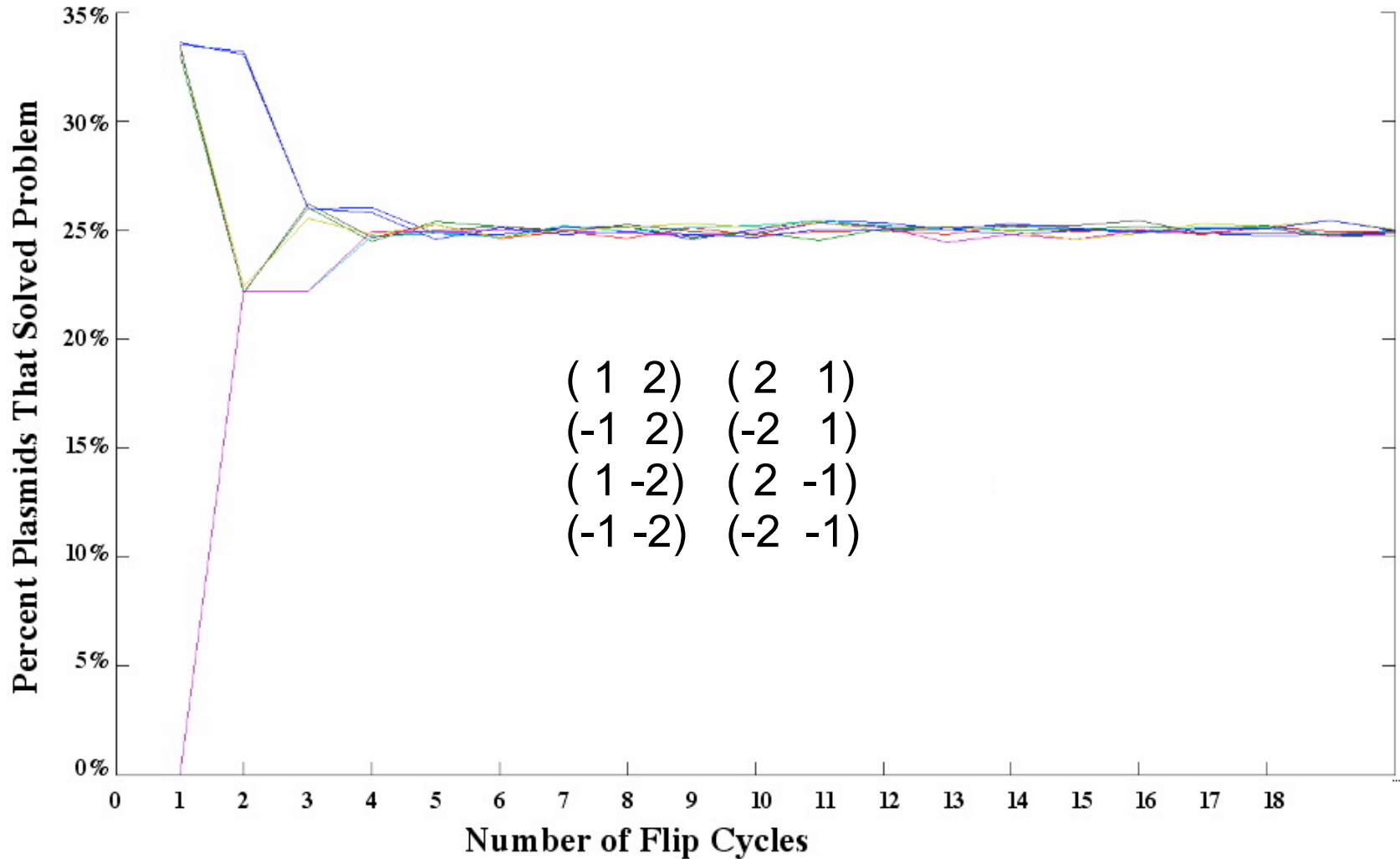
${}_5C_2 = 10$ ways to choose two spatula positions

Assume all 10 ways are equally likely

Simulate random flipping process with MATLAB

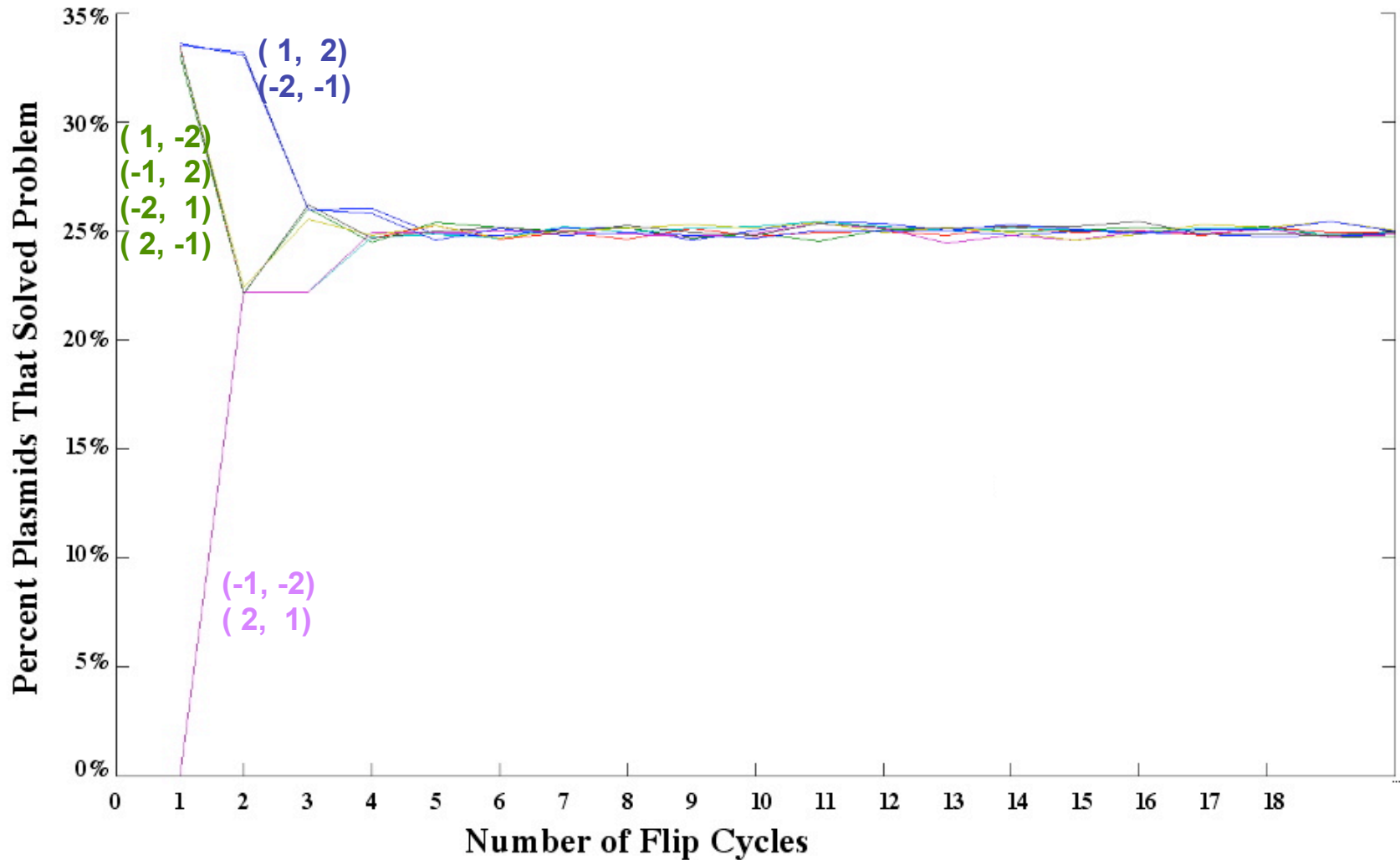
Modeling 8 Stacks of 2 Pancakes

*Solution is (1, 2) or (-2, -1)



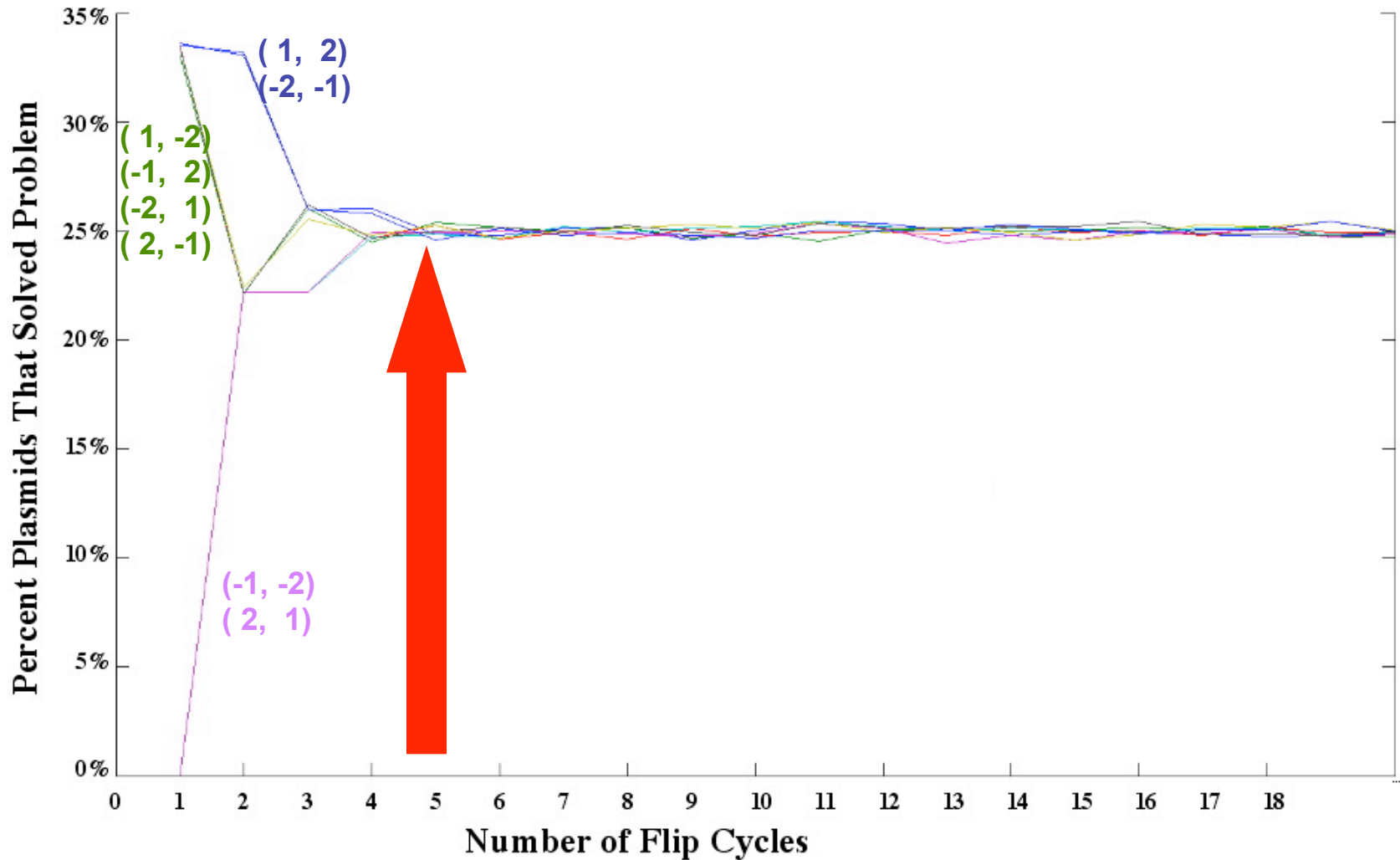
Modeling 8 Stacks of 2 Pancakes

*Solution is (1, 2) or (-2, -1)

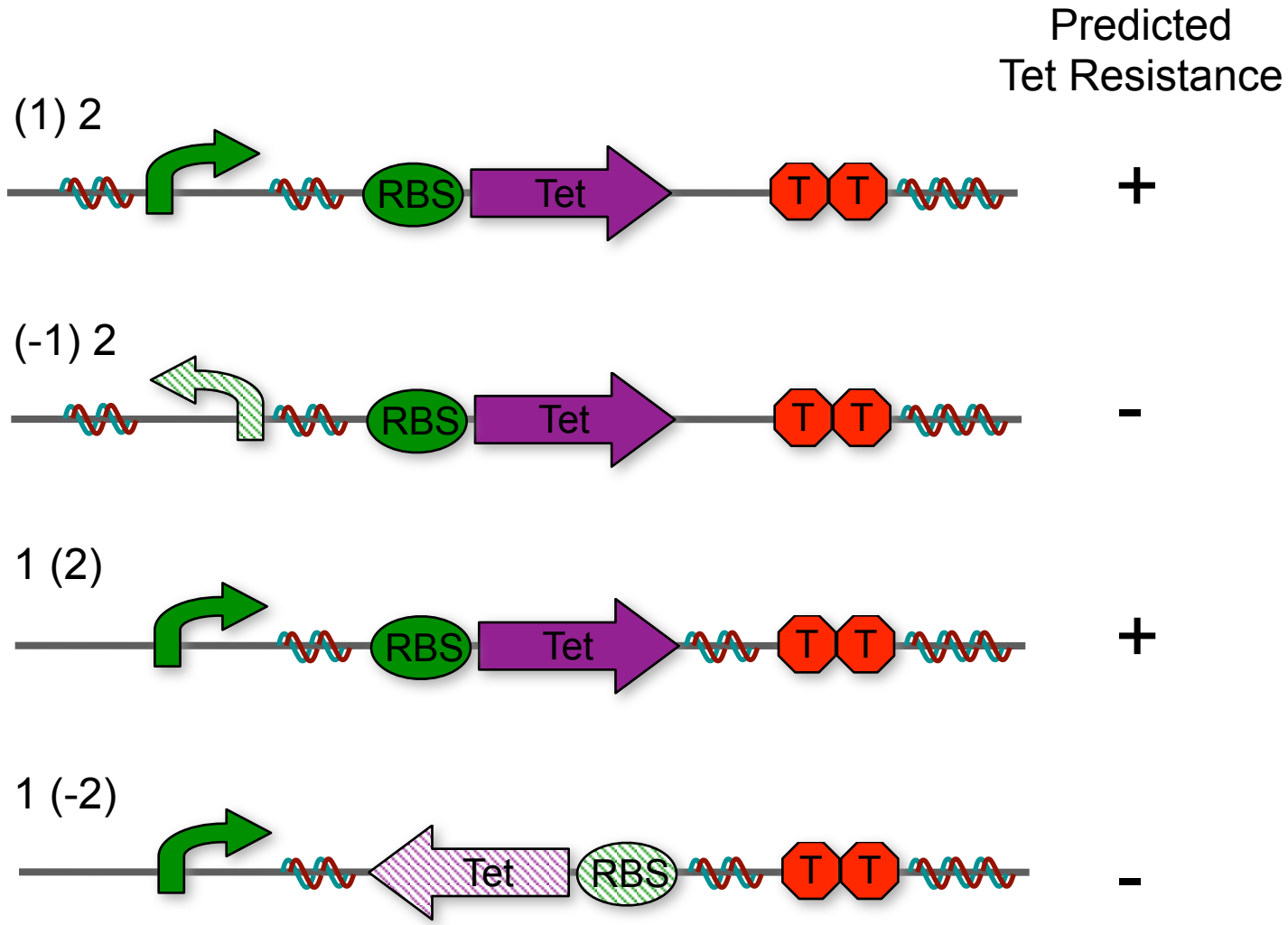


Modeling 8 Stacks of 2 Pancakes

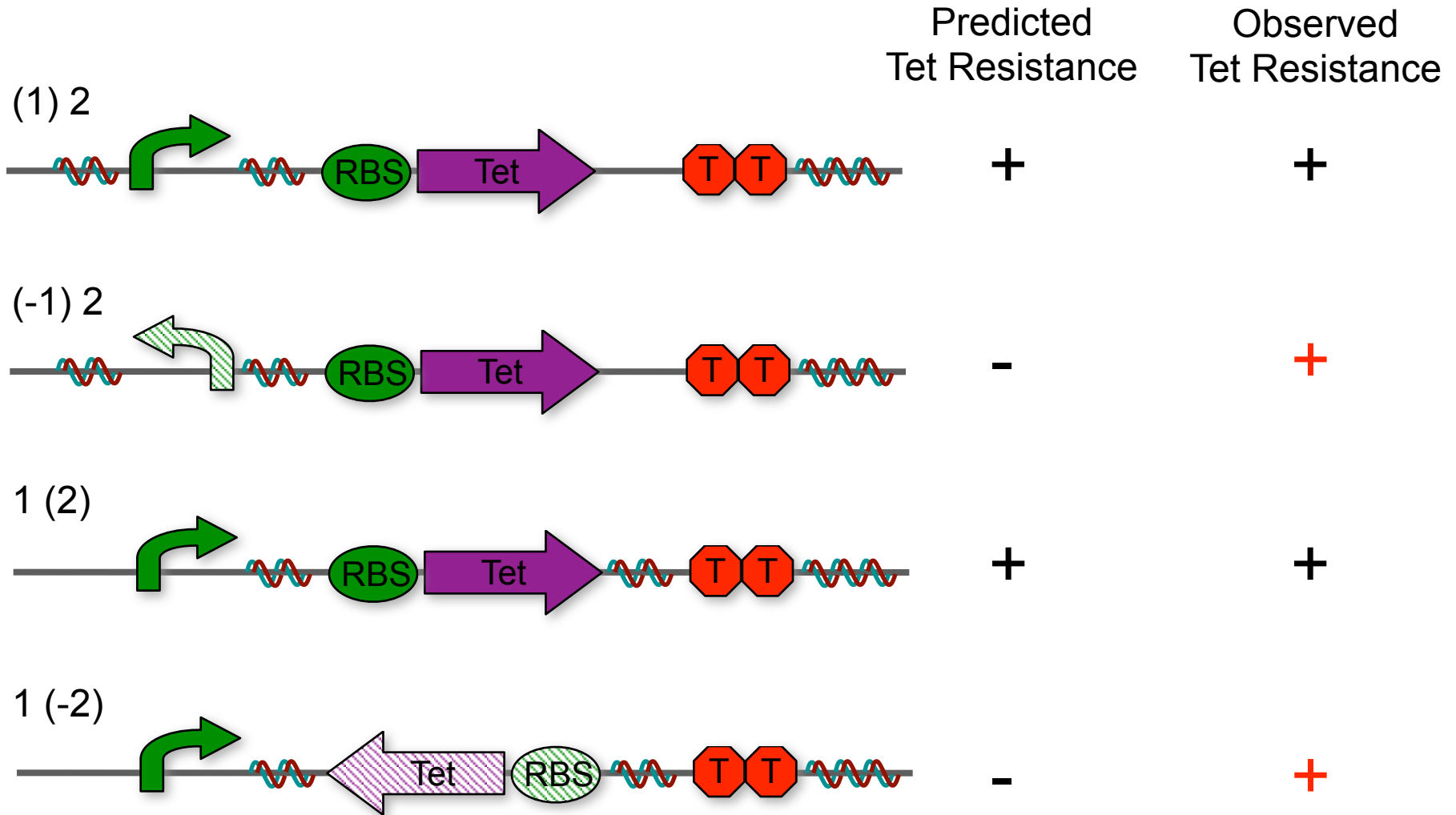
*Solution is (1, 2) or (-2, -1)



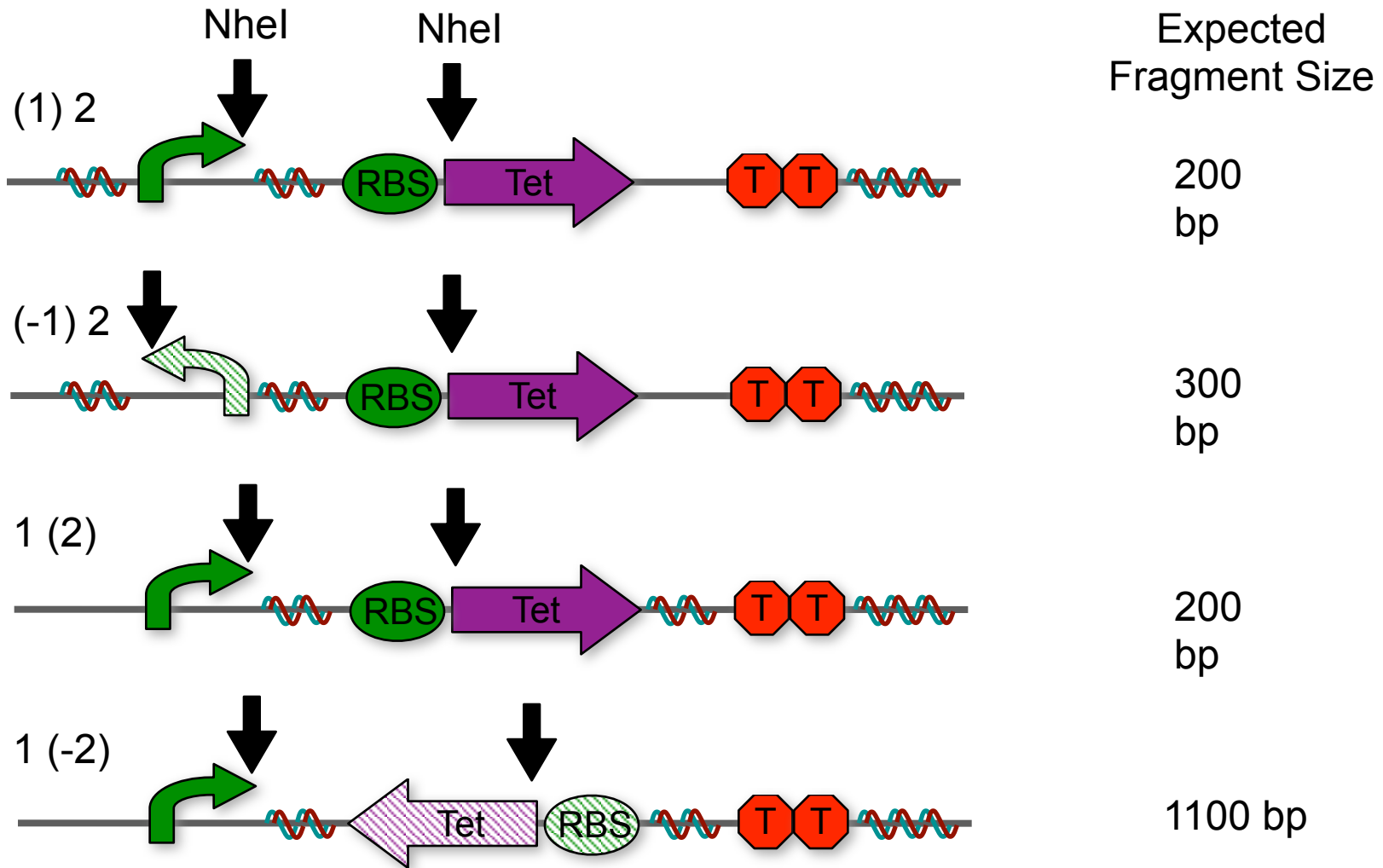
Four One-Pancake Constructs



Four One-Pancake Constructs

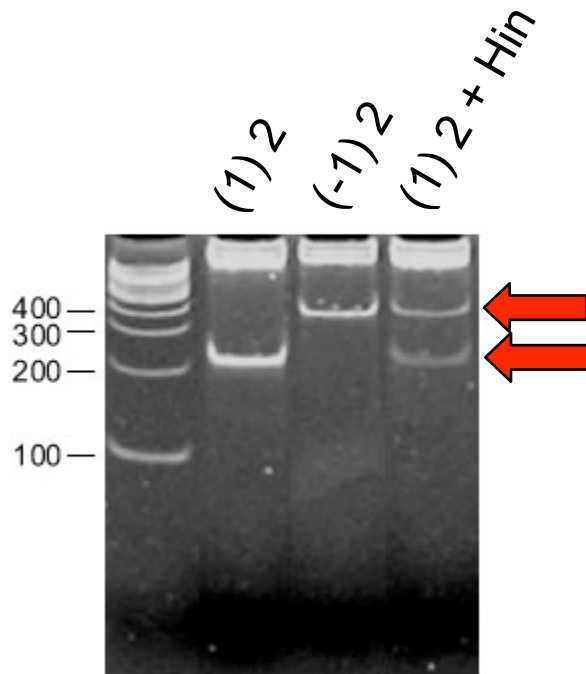


How to Detect Flipping via Restriction Digest

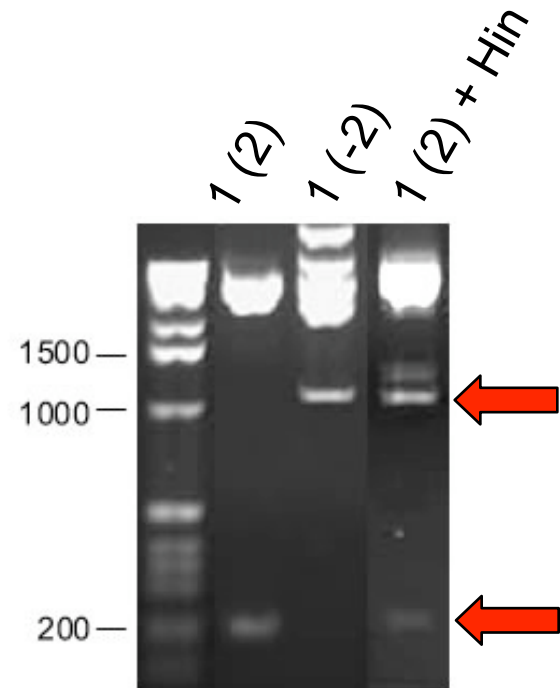


Hin-mediated Flipping

pBad Pancake Flipping



Tet Pancake Flipping



Read-Through Transcription Blocked by pSB1A7

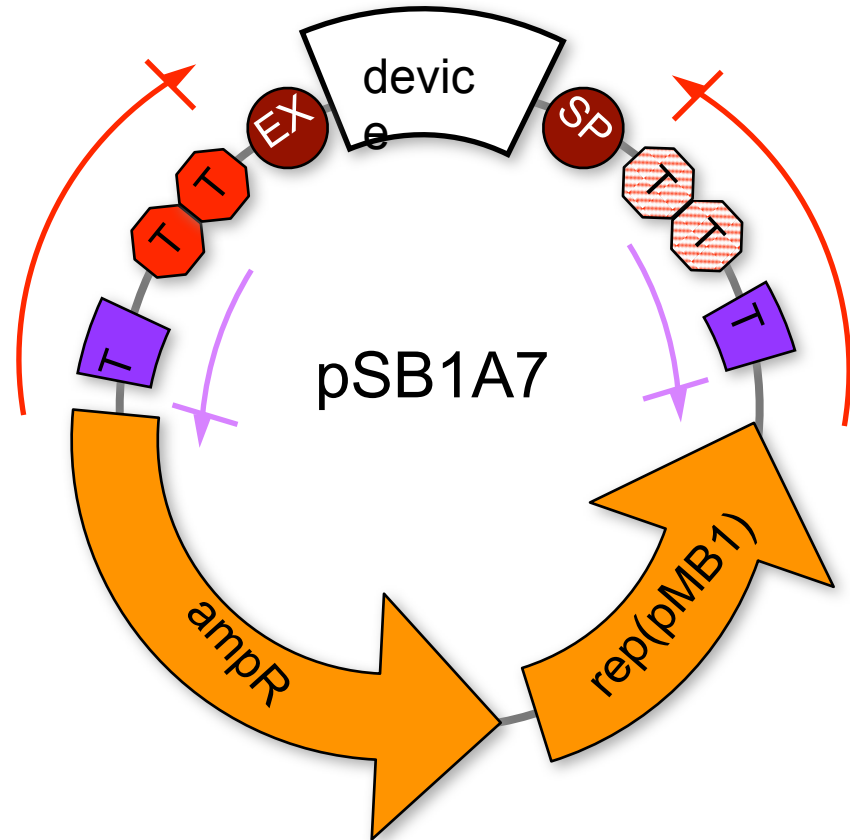
Observed Tet Resistance
in pSB1A3

+ (1) 2

+ (-1) 2

+ 1 (2)

+ 1(-2)



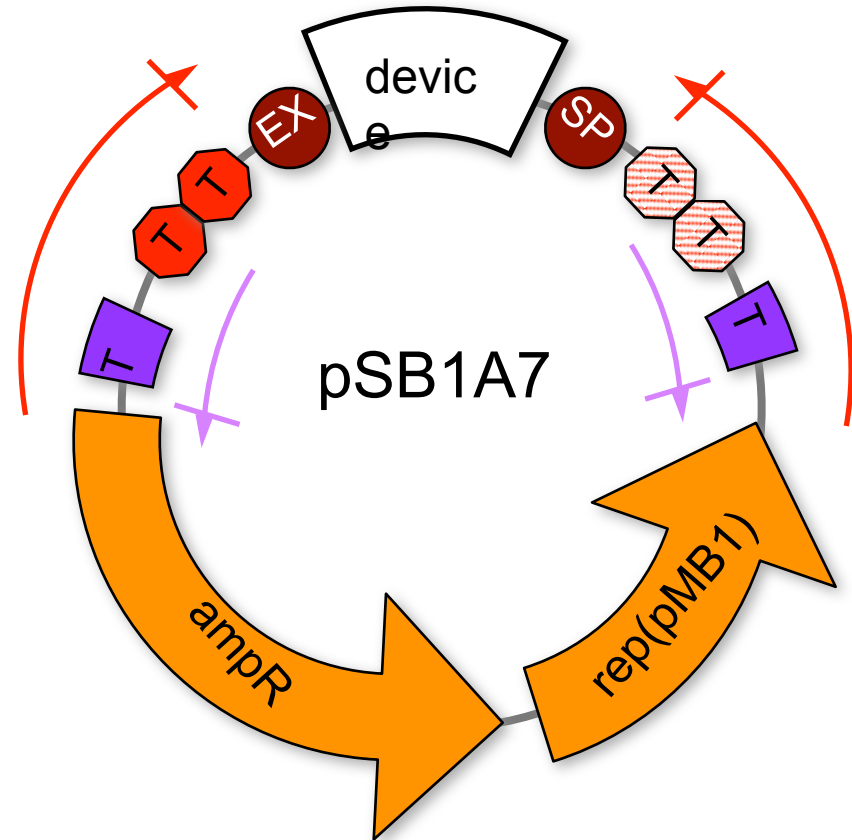
Read-Through Transcription Blocked by pSB1A7

Observed Tet Resistance
in pSB1A3

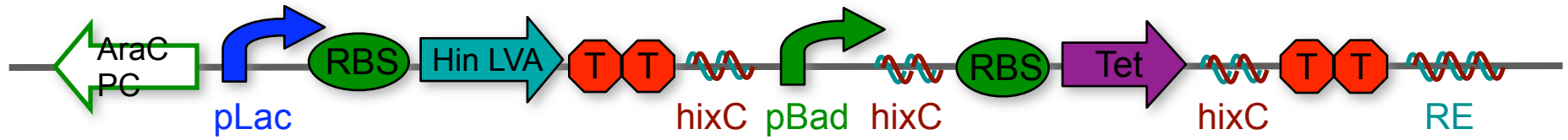
- + (1) 2
- + (-1) 2
- + 1 (2)
- + 1(-2)

Observed Tet Resistance
in *pSB1A7*

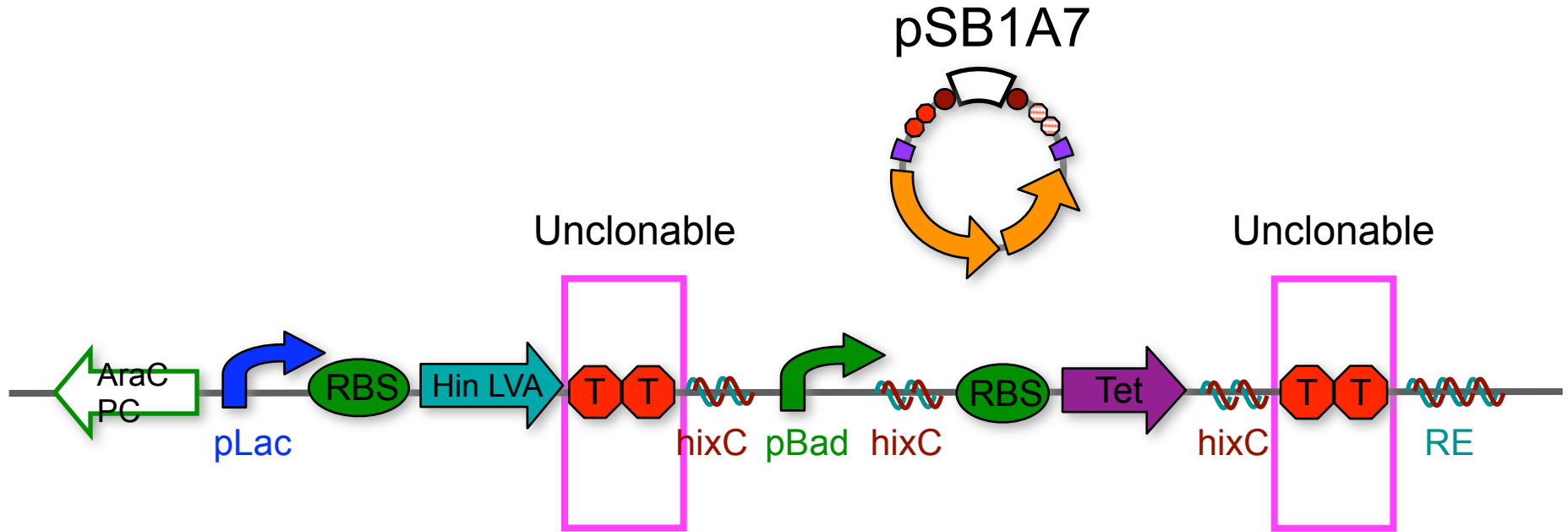
- + (1) 2
- (-1) 2
- + 1 (2)
- 1(-2)



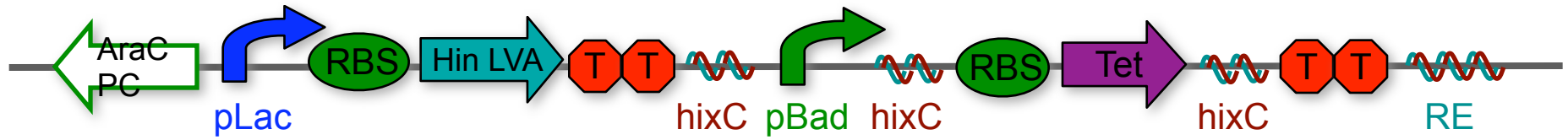
Original Design Problems



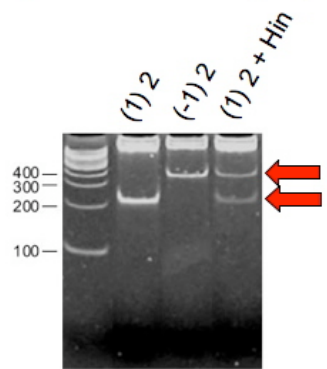
Original Design Problems



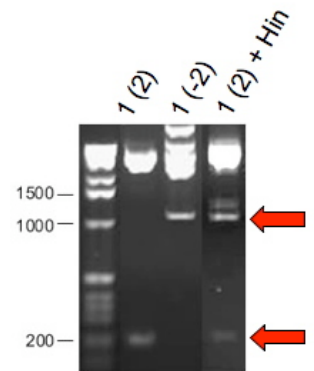
Original Design Problems



pBad Pancake Flipping



Tet Pancake Flipping

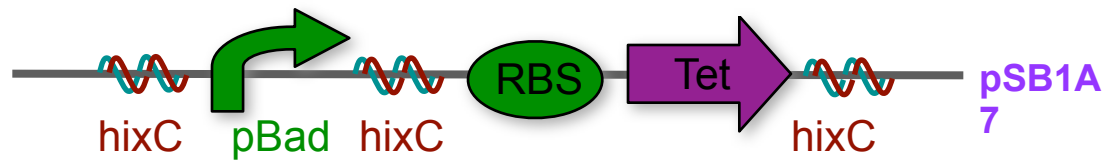


Flips too fast

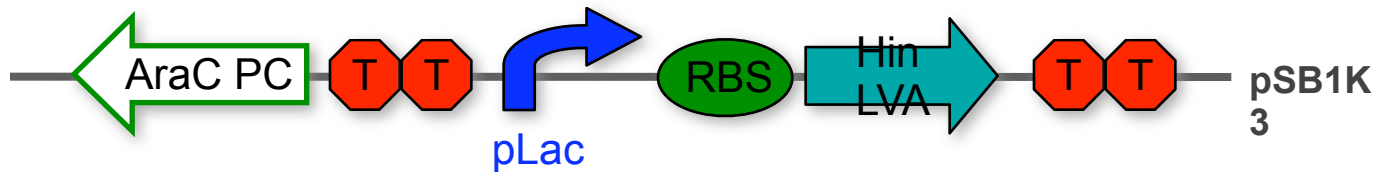


Two-Plasmid Solution

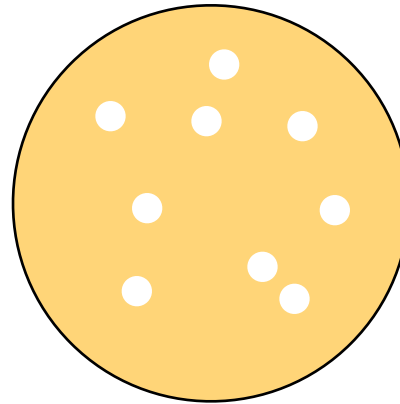
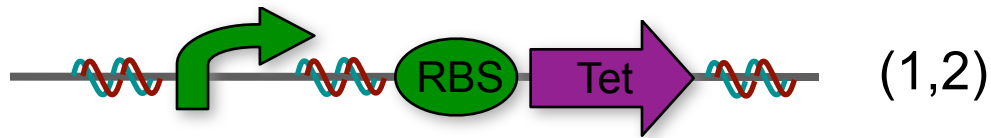
#1: Two pancakes (Amp vector)



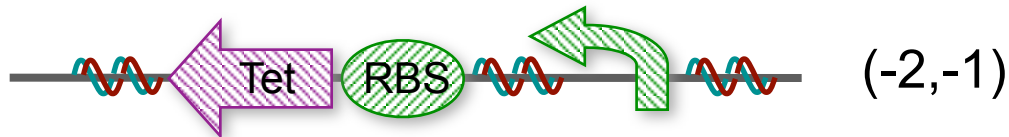
#2: AraC/Hin generator (Kan vector)



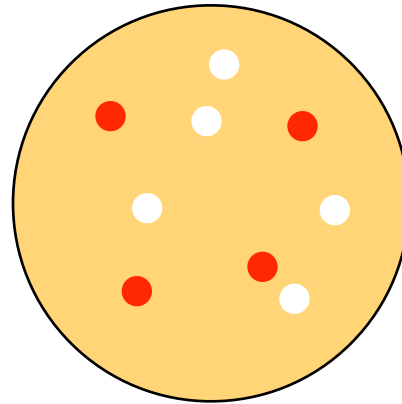
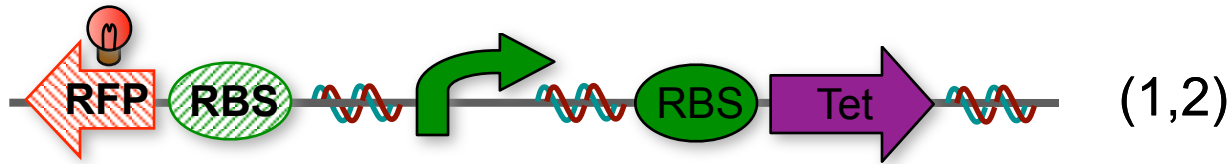
Biological Equivalence Problem



LB Amp + Kan+ Tet



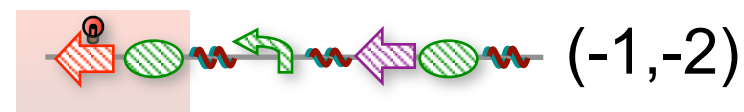
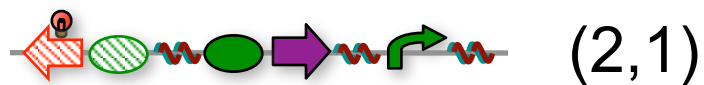
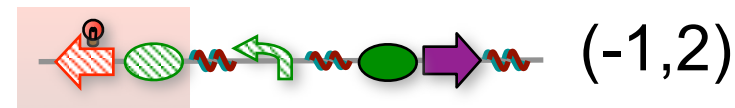
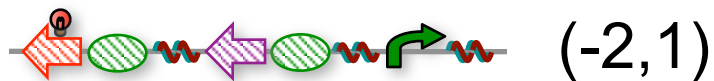
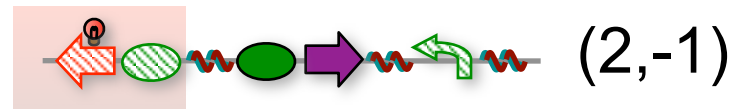
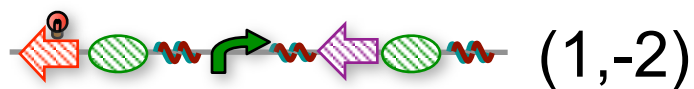
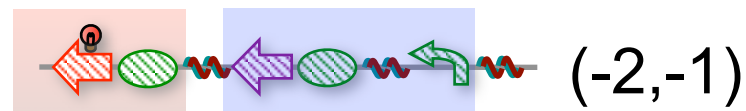
Biological Equivalence Solution



LB Amp + Kan+ Tet



Eight Two-Pancake Stacks



Intermediate Results

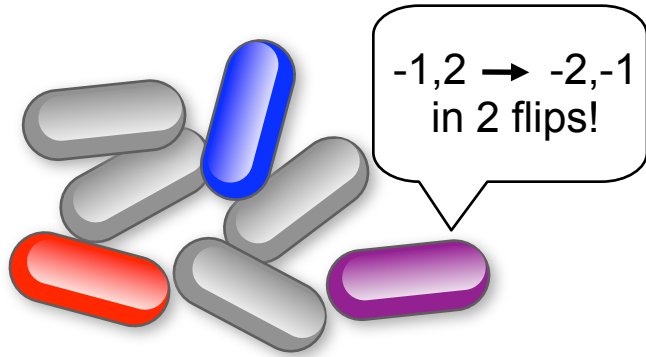
| Amp, Kan, Tet + AraC/ Hin Tet Resistance | Hin Vector + IPTG Hin Induction | Amp + Tet Tet Resistance |
|--|---|--|
| (1, 2) - | Hin-LVA + | (1, 2) + |
| (1,-2) (-1,-2) - | AraC/Hin-LVA + | (1,-2) (-1,-2) - |

Repress
pBad-Tet

Activate
pLac-Hin

Screen for
Orientation

CONCLUSIONS: Consequences of DNA Flipping Devices



PRACTICAL

Proof-of-concept for bacterial computers

Data storage

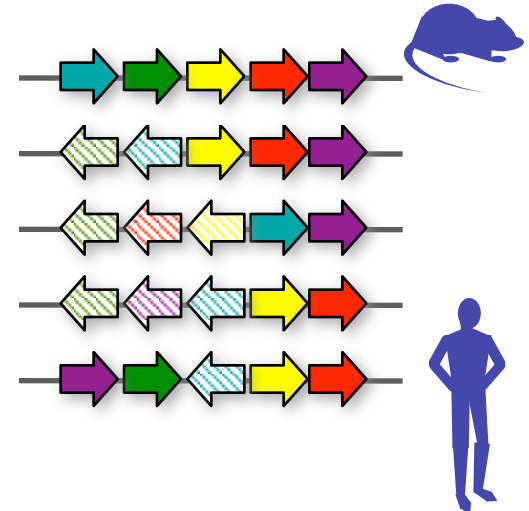
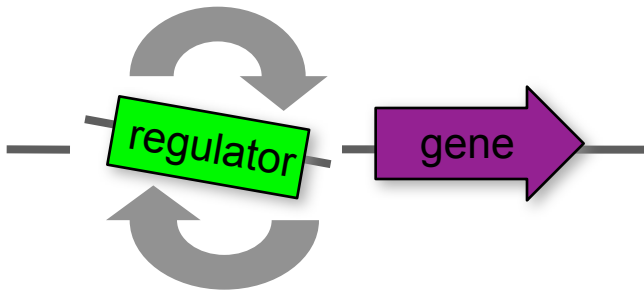
n units gives $2^n(n!)$ combinations



BASIC BIOLOGY RESEARCH

Improved transgenes *in vivo*

Evolutionary insights



Next Steps

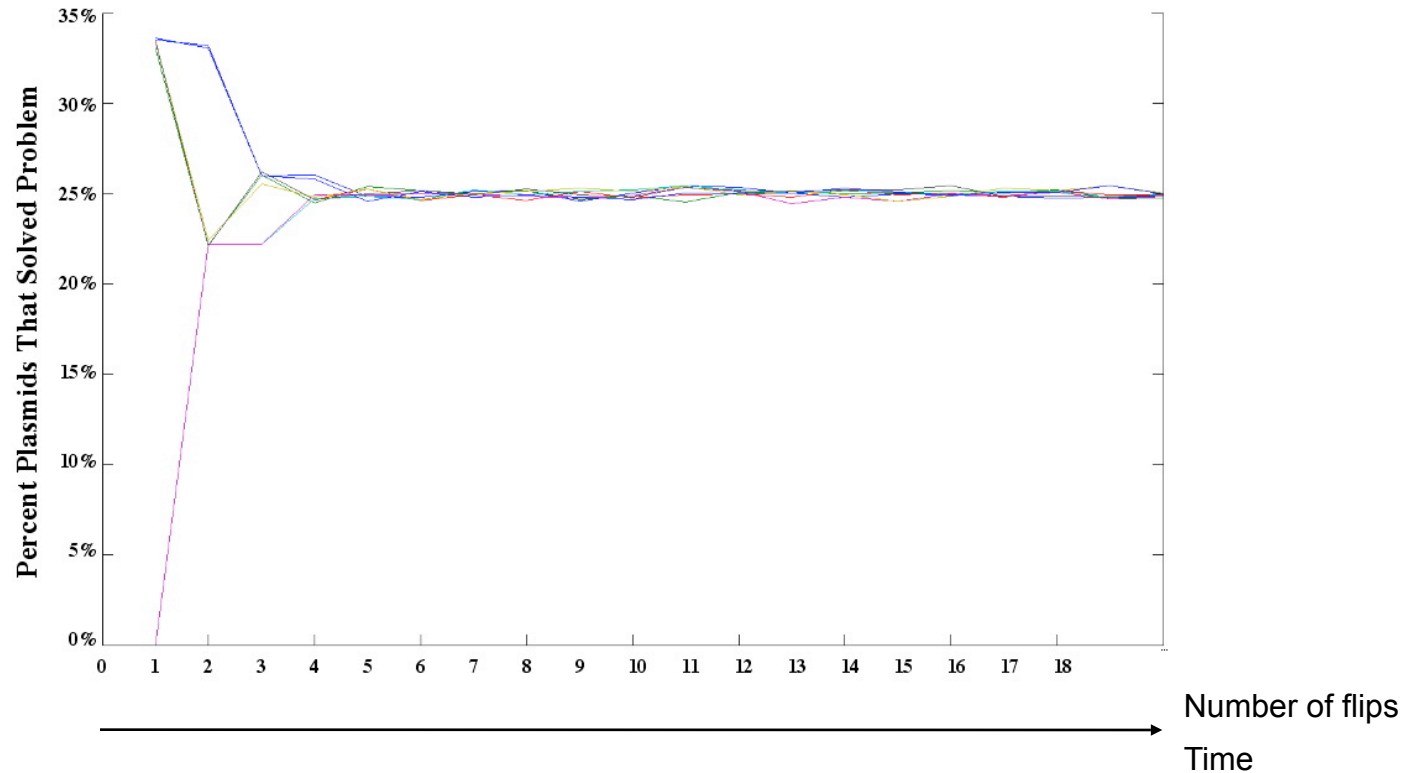
Better control of kinetics

Slow down Hin activity

Determine x flips second⁻¹

Size bias?

Number of flips vs. Time?



Modeling Plasmid Copy Number

$$P(\text{cell survives}) = 1 - \sum_{j=0}^{x-1} \binom{n}{j} p^j (1-p)^{n-j}$$

n = plasmid copy number

p = probability of a pancake stack being sorted

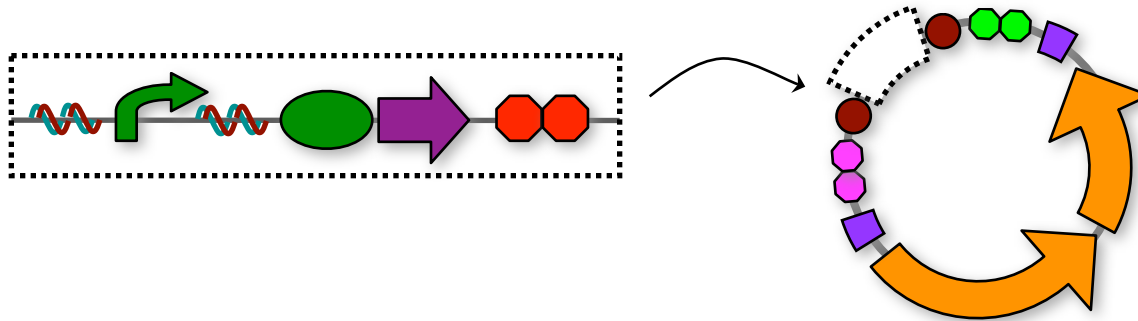
x = number of sorted stacks sufficient for cell survival

| Number of Pancakes | Number of Plasmids | Number Solutions Required | P (solution) | P (cell survives) |
|--------------------|--------------------|---------------------------|--------------|--------------------------------------|
| 2-stack | 200 | 1 | 0.25 | $1 - (1 - 0.25)^{200} \approx 1$ |
| 4-stack | 200 | 1 | 0.003 | $1 - (1 - 0.003)^{200} \approx 0.45$ |
| different 4-stack | 200 | 1 | 0.01 | $1 - (1 - 0.01)^{200} \approx 0.87$ |

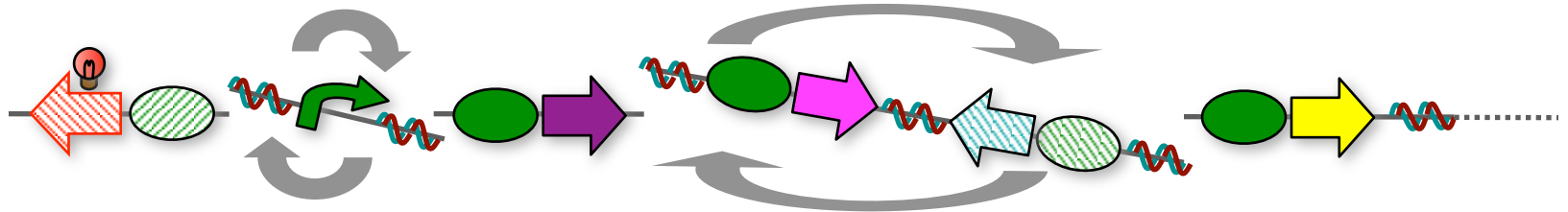
Next Steps

Improved insulating vector

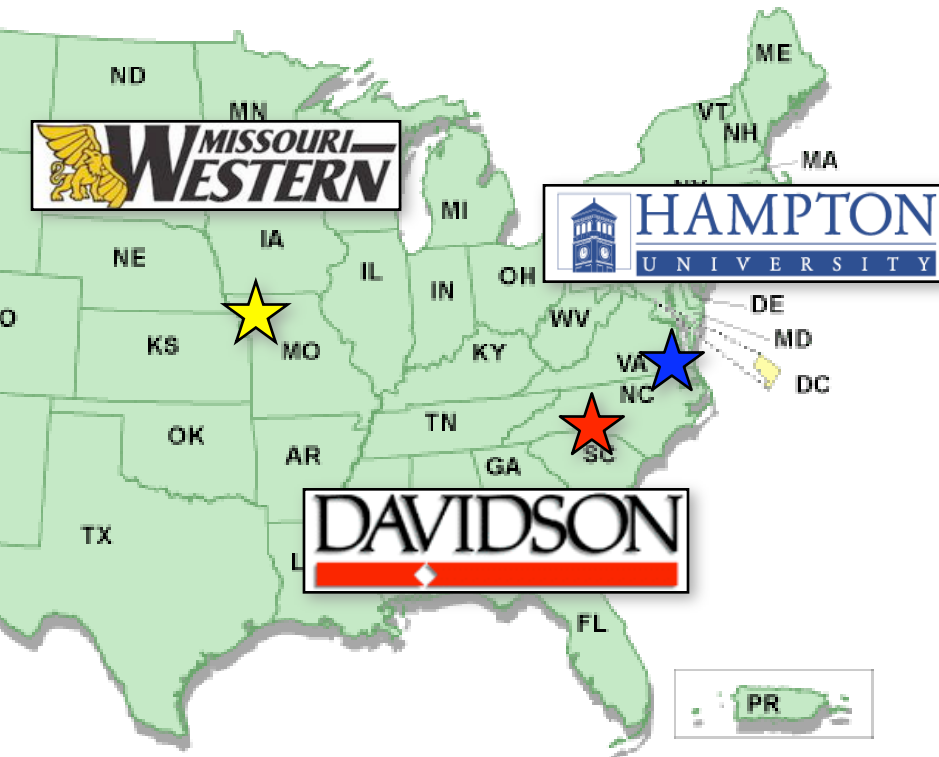
Accepts B0015 double terminator



Bigger pancake stacks



Summary: What We Learned



Multiple campuses increase capacity with parallel processing

Collective Troubleshooting

Primarily Undergraduate Institutions can iGEM collaboratively

Summary: What We Learned

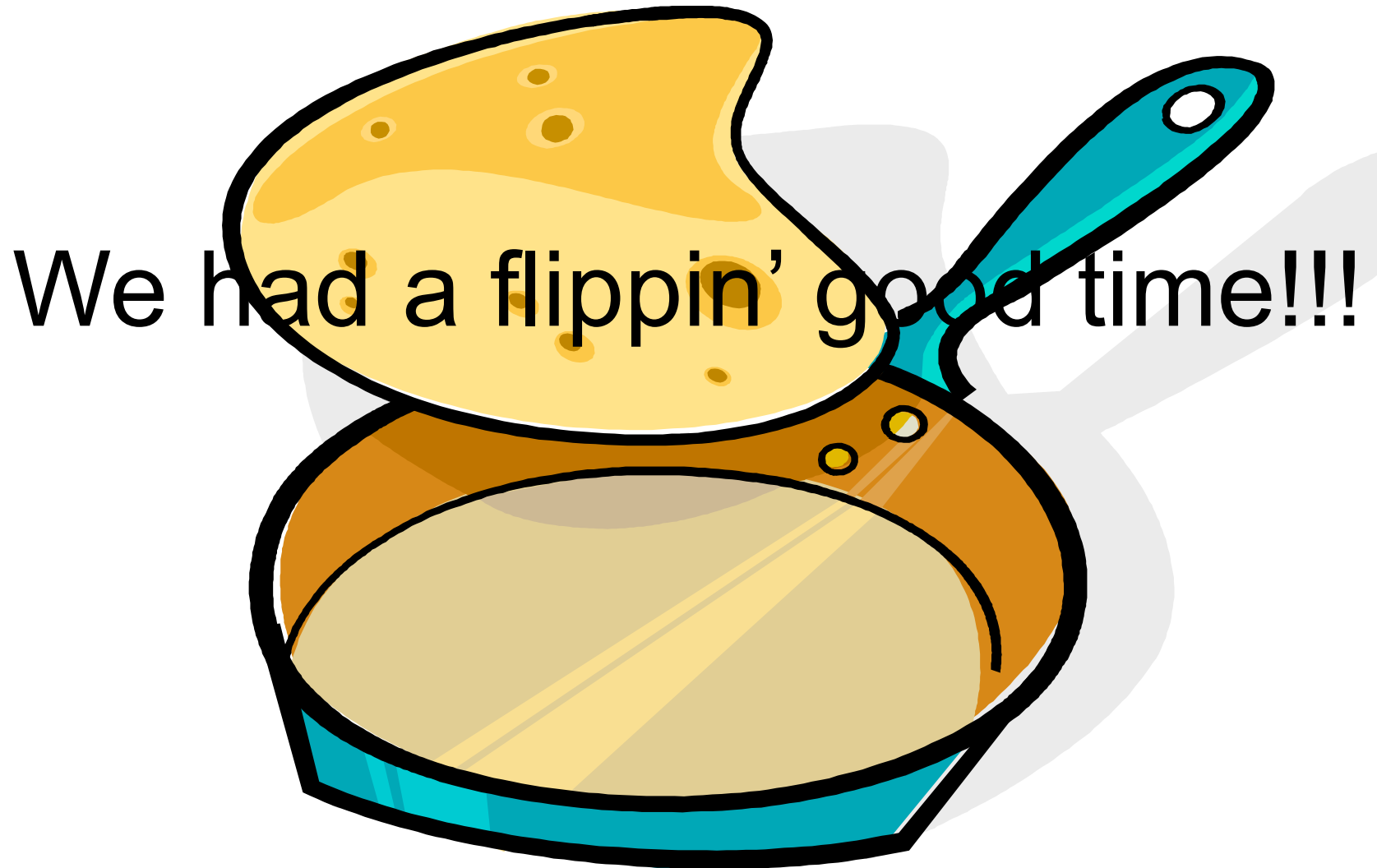


Math and Biology mesh really well

Math modeling
We proved a new theorem!

Challenges of biological components

But above all...



E.HOP: A Bacterial Computer to Solve the Pancake Problem

Collaborators at MWSU:

Marian Broderick, Adam Brown,
Trevor Butner, Lane Heard, Eric
Jessen, Kelly Malloy, Brad Ogden



Support: Davidson College

HHMI

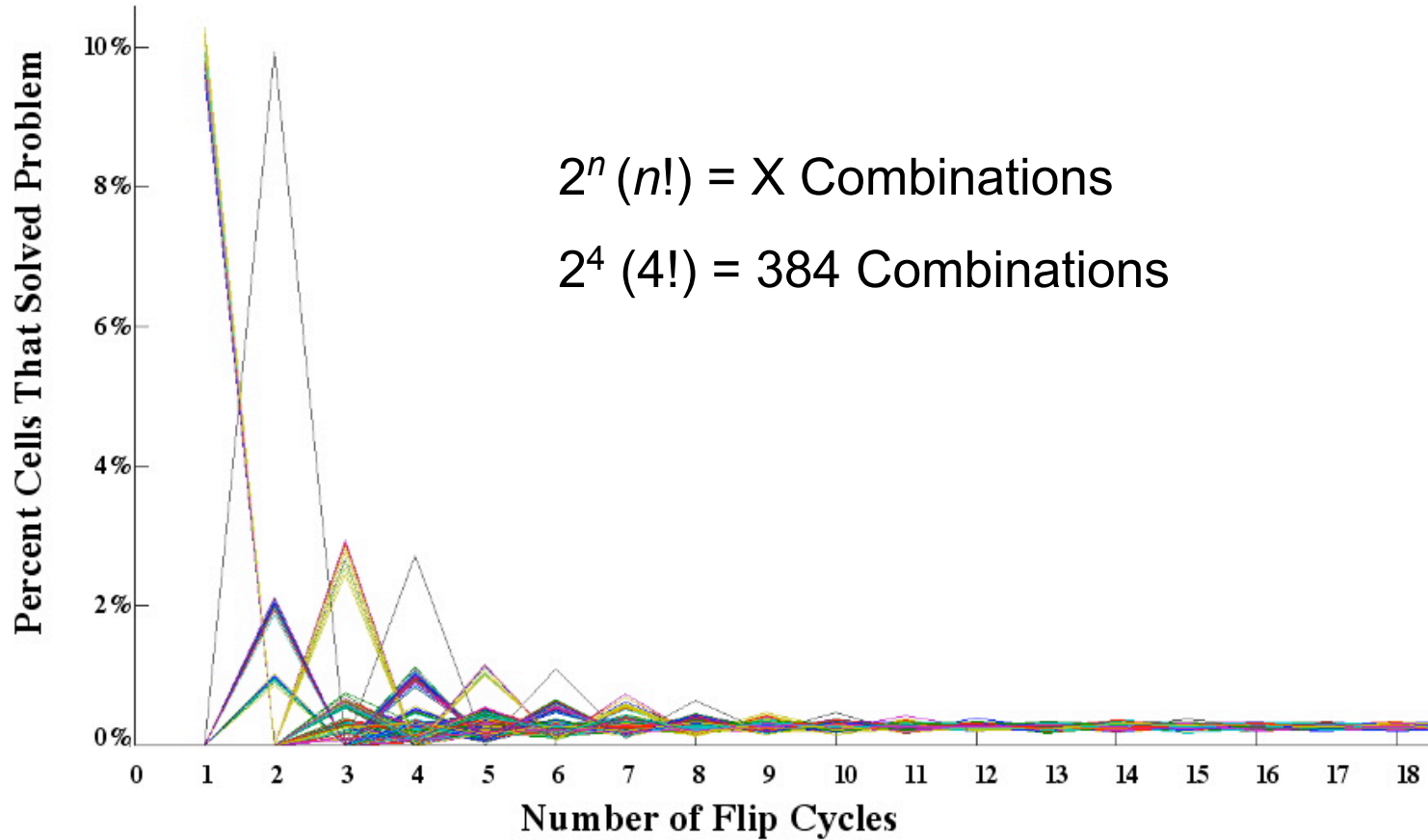
NSF

Genome Consortium for Active Teaching

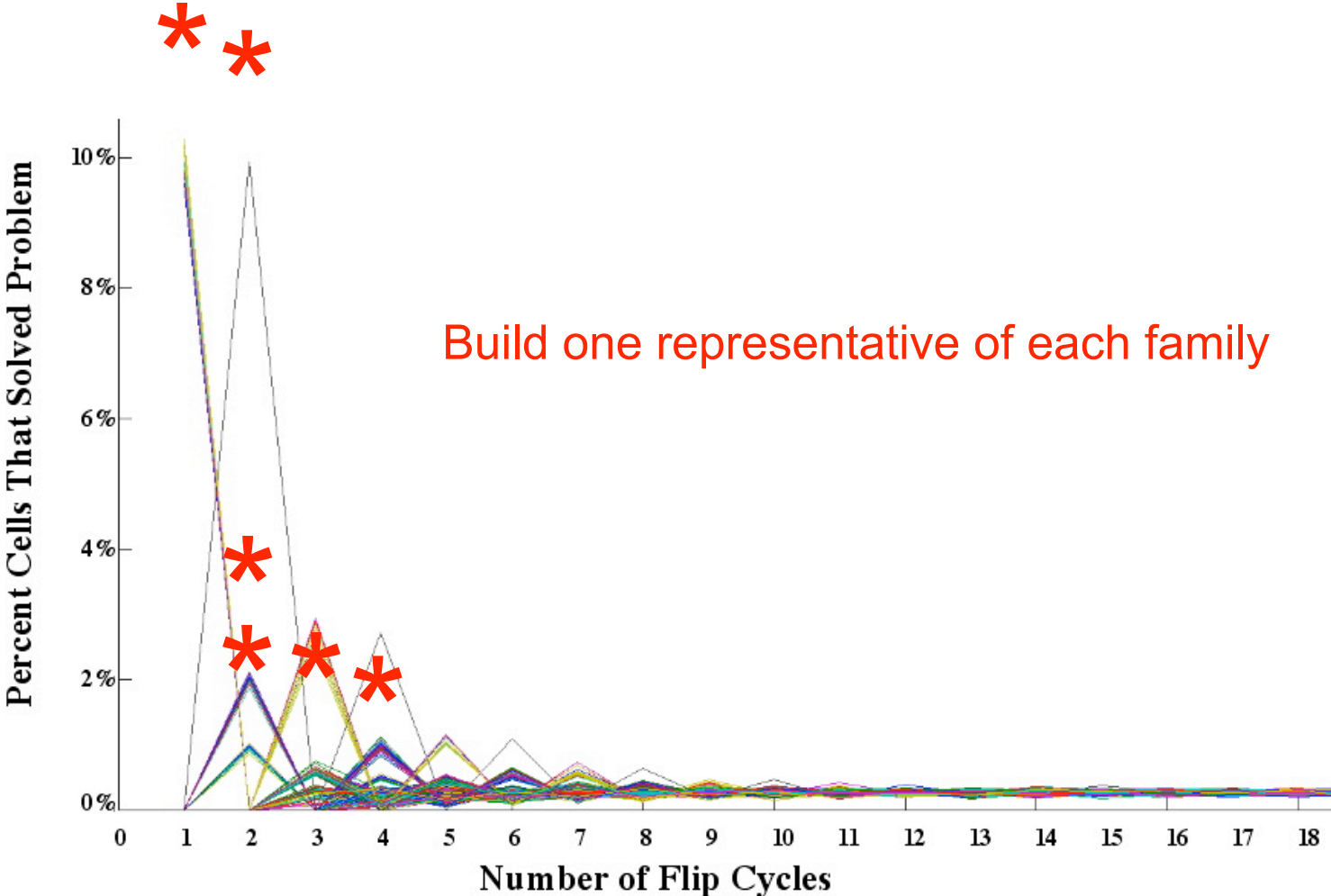
James G. Martin Genomics Program

Extra Slides >>>

Modeling 384 Stacks of 4 Pancakes

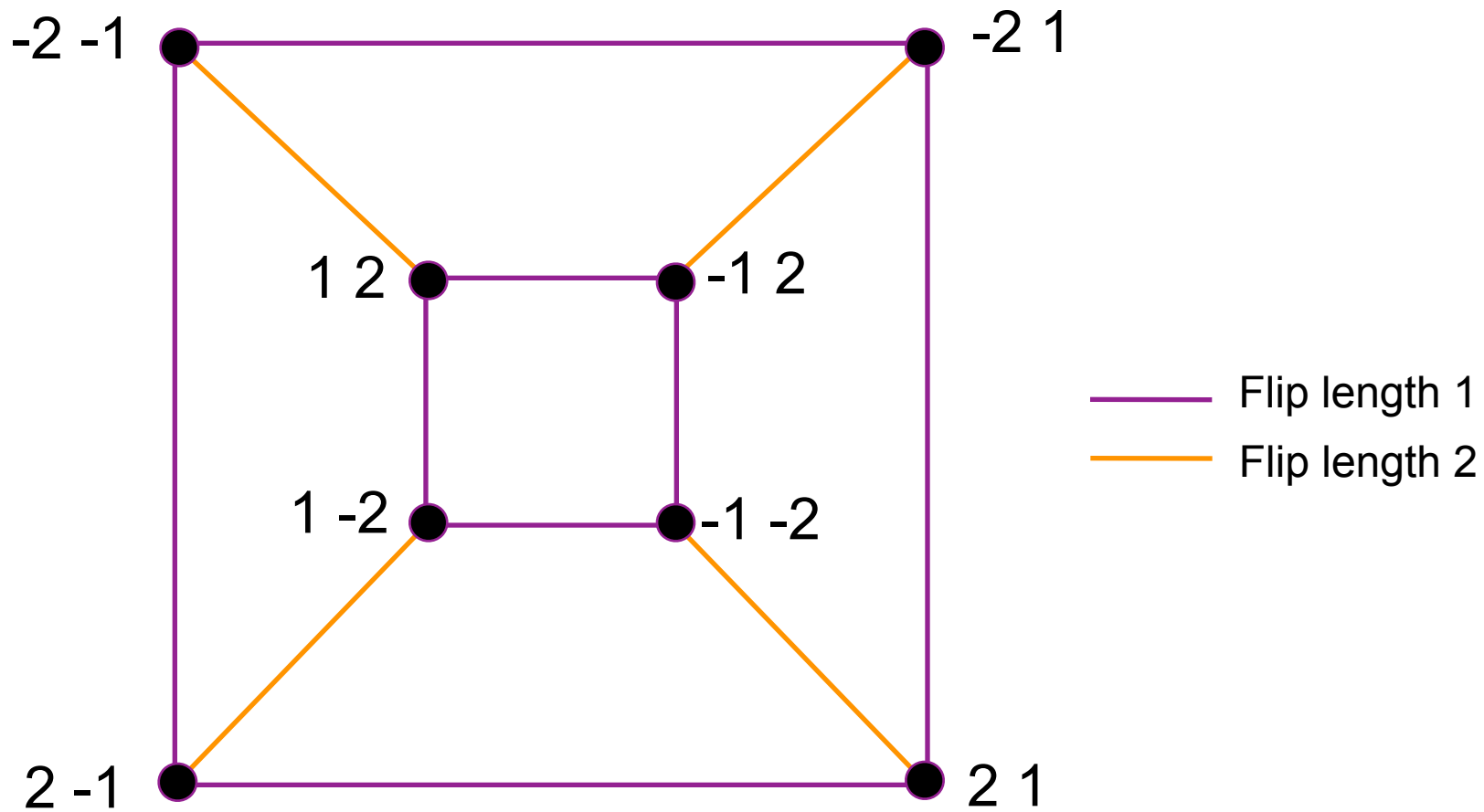


Modeling 384 Stacks of 4 Pancakes

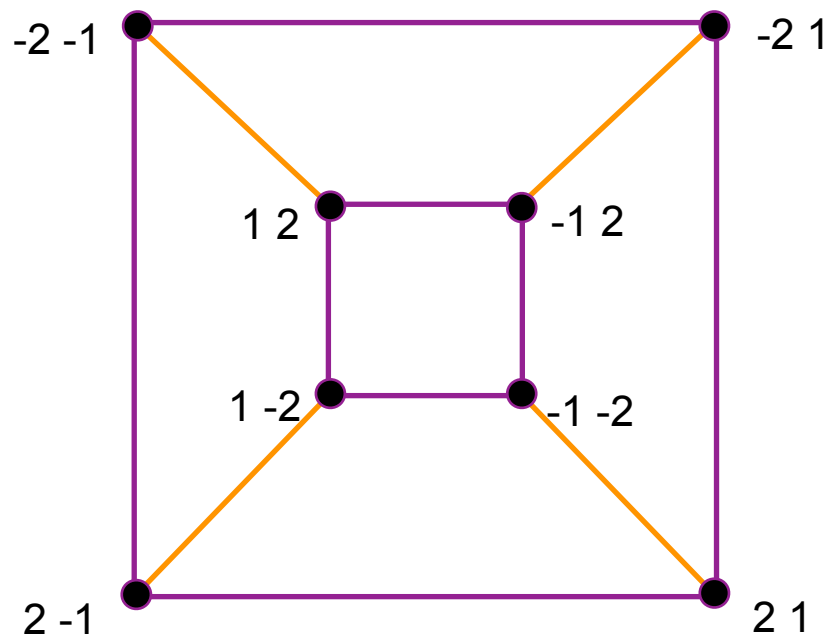


Modeling Random Flipping

```
for trial = 1 to n trials
    stack = input_stack
    flips = 0
    while stack ~= 1:k
        flips = flips + 1
        int = choose_random_interval
        stack = [stack(1:(int(1)-1))
                -1 * fliplr(stack(int(1):int(2)))
                stack(int(2)+1:k)]
    end
end
```

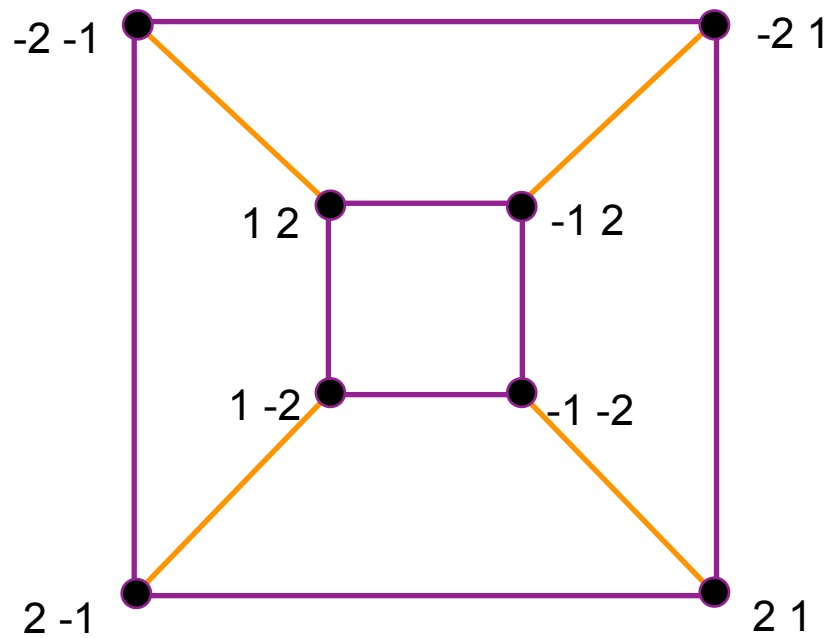


P_2 : 2 Burnt Pancakes
 8 vertices, each with degree 3



P_2 : 2 Burnt Pancakes
8 vertices, each with degree 3

| | | | | | | | | |
|----------|--------|---------|----------|---------|----------|---------|--------|---------|
| A | 1 2 | -1 2 | -1 -2 | 1 -2 | -2 -1 | -2 1 | 2 1 | 2 -1 |
| 1 2 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| -1 2 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| -1 -2 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 -2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| -2 -1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| -2 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 2 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 2 -1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |



B_2 : 2 Burnt Pancakes
8 vertices, each with degree 3

| | | | | | | | | |
|-------|--------|---------|----------|---------|----------|---------|--------|---------|
| A^2 | 1 2 | -1 2 | -1 -2 | 1 -2 | -2 -1 | -2 1 | 2 1 | 2 -1 |
| 1 2 | 3 | 0 | 2 | 0 | 0 | 2 | 0 | 2 |
| -1 2 | 0 | 3 | 0 | 2 | 2 | 0 | 2 | 0 |
| -1 -2 | 2 | 0 | 3 | 0 | 0 | 2 | 0 | 2 |
| 1 -2 | 0 | 2 | 0 | 3 | 2 | 0 | 2 | 0 |
| -2 -1 | 0 | 2 | 0 | 2 | 3 | 0 | 2 | 0 |
| -2 1 | 2 | 0 | 2 | 0 | 0 | 3 | 0 | 2 |
| 2 1 | 0 | 2 | 0 | 2 | 2 | 0 | 3 | 0 |
| 2 -1 | 2 | 0 | 2 | 0 | 0 | 2 | 0 | 3 |