Scouting the Heights of the Kreuzer-Skarke Landscape

Andres Rios-Tascon (Cornell)

String Phenomenology Seminar Series 2020

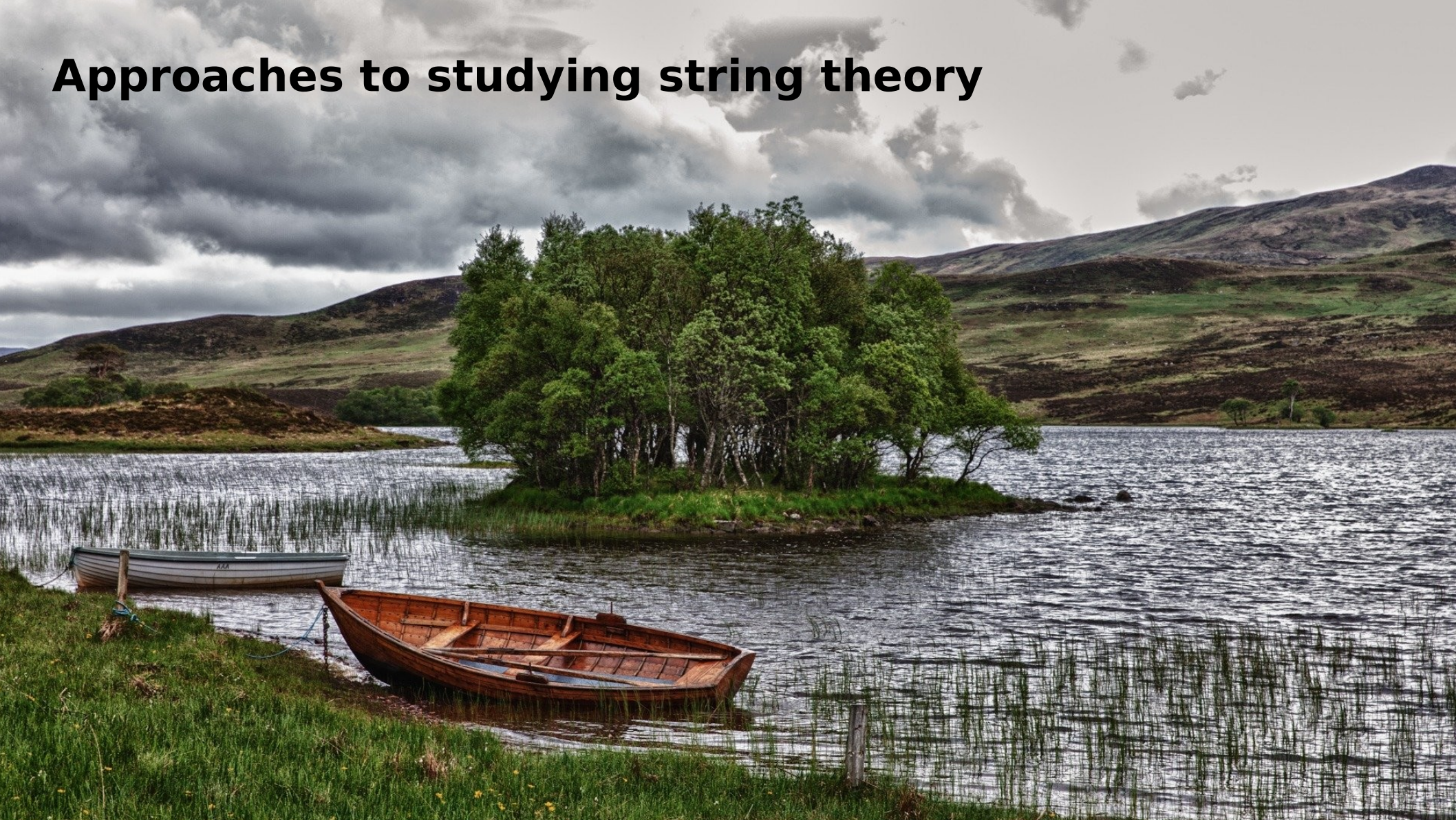
Outline

Goals: Explain our computational advancements for exploring KS.
Showcase our open-source package CYTools.

1. Motivation and background
2. Description of our new algorithms and computational advances
3. CYTools live demo
4. Conclusions

Based on: M. Demirtas, L. McAllister, A.R.T., 2008.01730, 201x.xxxxx

Approaches to studying string theory



Approaches to studying string theory

Approaches to studying string theory

Approaches to studying string theory

Constructing Calabi-Yau 3-folds

Batyrev construction: [alg-geom/9310003]

1) Take 4D reflexive polytope.

2) Find fine, regular, star triangulation.

3) Interpret triangulation as a fan and construct the toric variety.

4) Anti-canonical hypersurface is a smooth CY threefold.

Reflexive polytopes

- A lattice polytope is reflexive if its dual is also a lattice polytope.

Reflexive polytopes

- A lattice polytope is reflexive if its dual is also a lattice polytope.
- Kreuzer and Skarke found all 473,800,776 4D reflexive polytopes and constructed their celebrated database. [hep-th/0002240, math.AG/0001106]

Reflexive polytopes

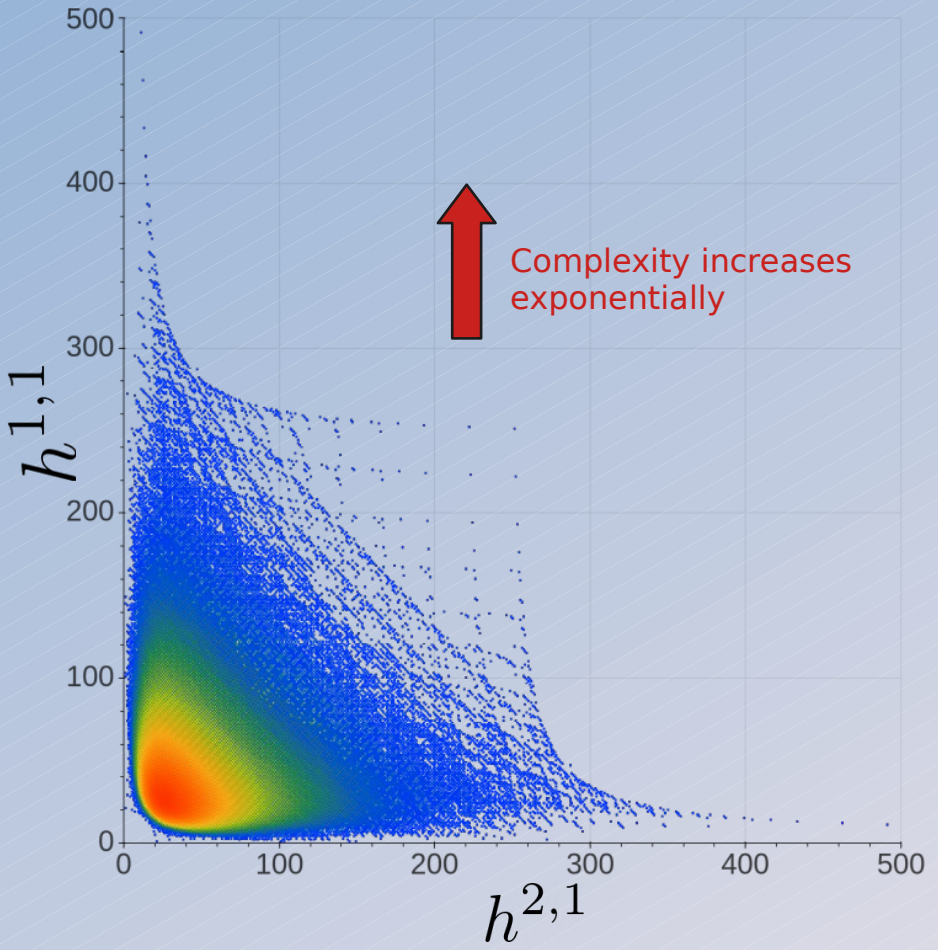
- A lattice polytope is reflexive if its dual is also a lattice polytope.
- Kreuzer and Skarke found all 473,800,776 4D reflexive polytopes and constructed their celebrated database. [hep-th/0002240, math.AG/0001106]
- They also provided “PALP: a Package for Analyzing Lattice Polytopes” to handle polytope computations. [math/0204356, arXiv:1205.4147]

Reflexive polytopes

- A lattice polytope is reflexive if its dual is also a lattice polytope.
- Kreuzer and Skarke found all 473,800,776 4D reflexive polytopes and constructed their celebrated database. [hep-th/0002240, math.AG/0001106]
- They also provided “PALP: a Package for Analyzing Lattice Polytopes” to handle polytope computations. [math/0204356, arXiv:1205.4147]
- For the rest of the CY construction we are left on our own, but mathematical software like Sage can be used.

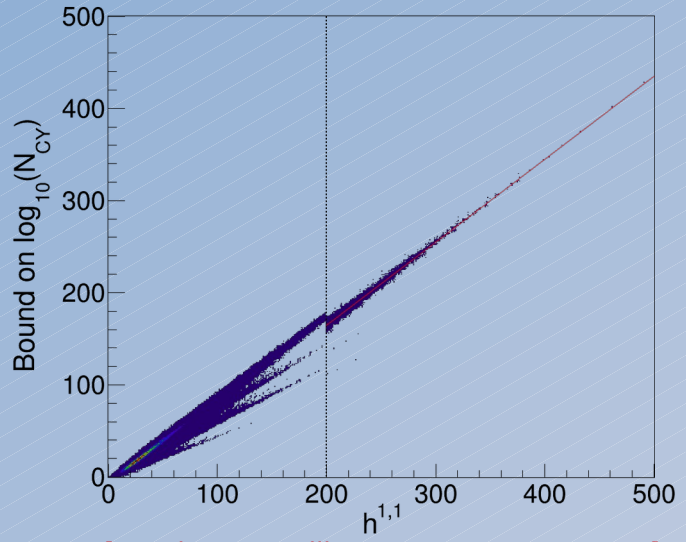
Exploring the KS database

- The complexity of the database grows exponentially with the number of points on the polytope ($\sim h^{1,1}$).

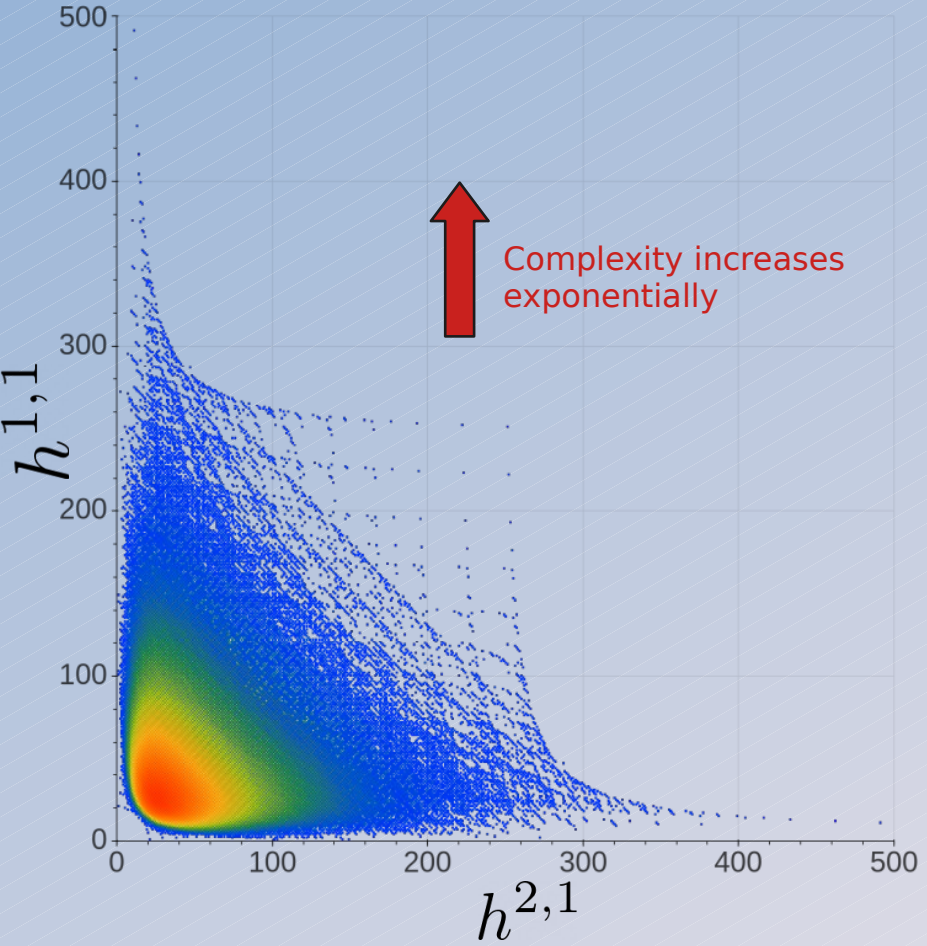


Exploring the KS database

- The complexity of the database grows exponentially with the number of points on the polytope ($\sim h^{1,1}$).
- The $h^{1,1} \gg 1$ region potentially contains most of the KS landscape.

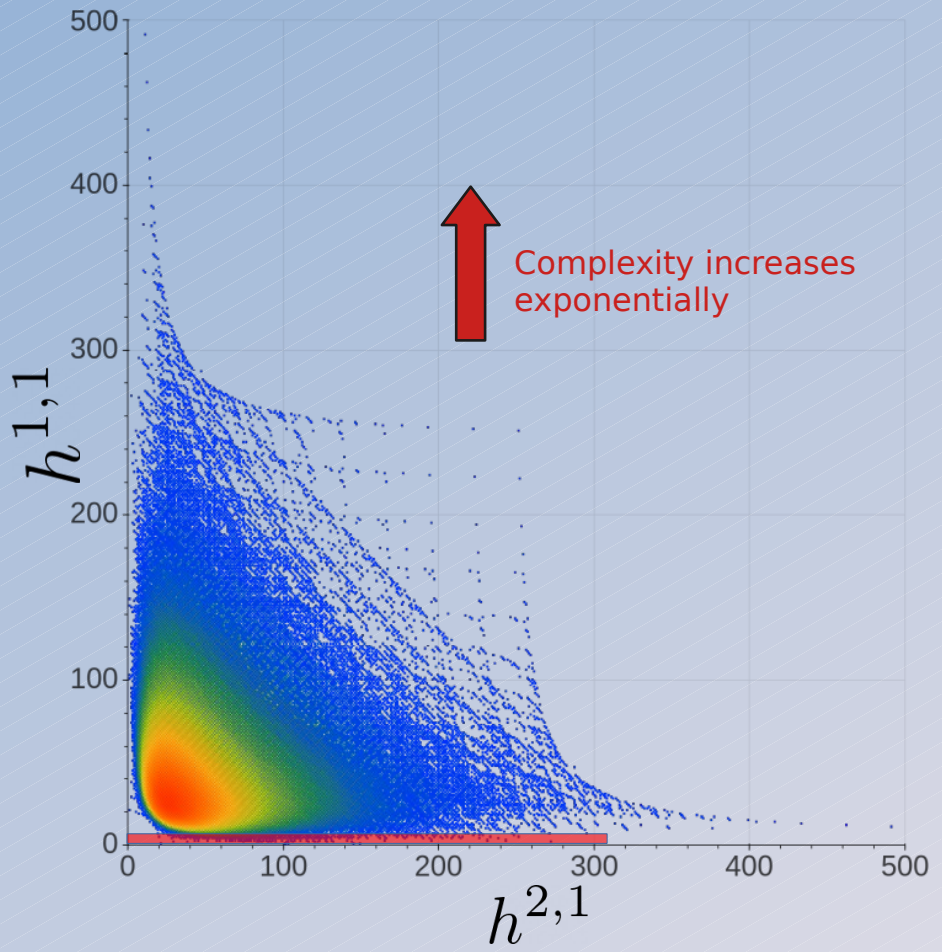


[Demirtas, McAllister, A.R.T., 2008.01730]



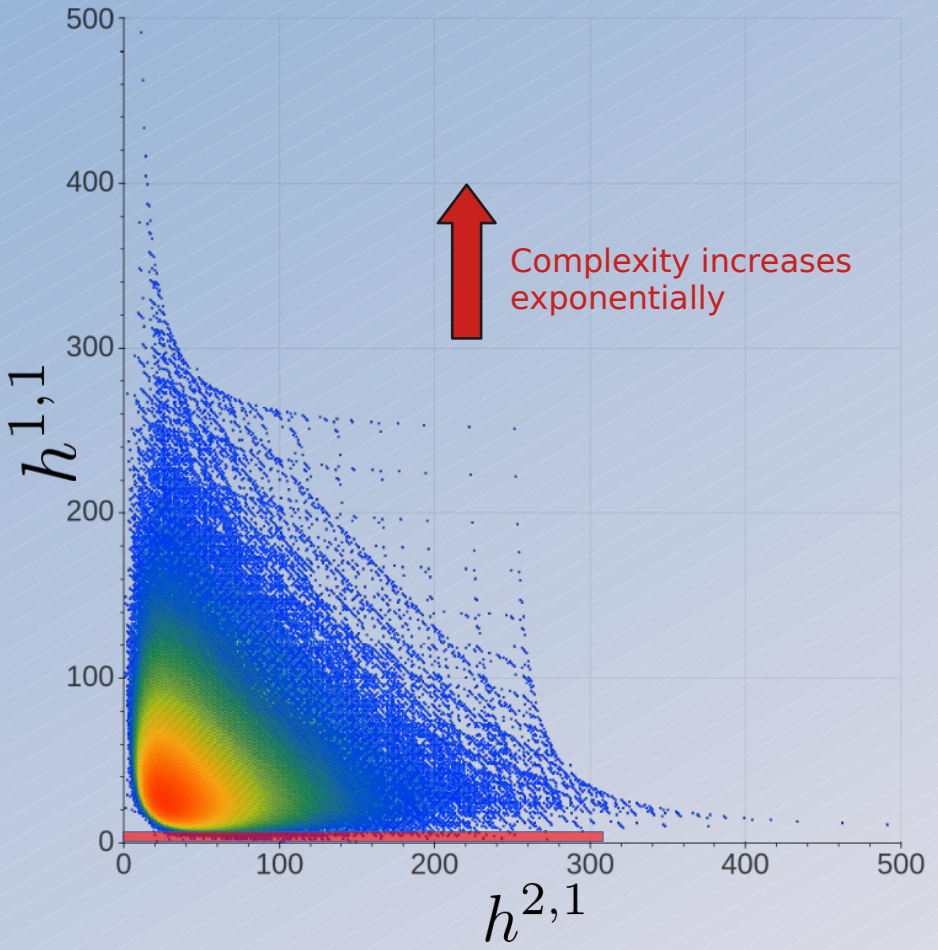
Exploring the KS database

- The complexity of the database grows exponentially with the number of points on the polytope ($\sim h^{1,1}$).
- The $h^{1,1} \gg 1$ region potentially contains most of the KS landscape.
- For context, in 2014 an immense effort and over 100,000 CPU hours were necessary to extract all CYs with $h^{1,1} <= 6$.
  [Altman, Gray, He, Jejjala, Nelson, 1411.1418]

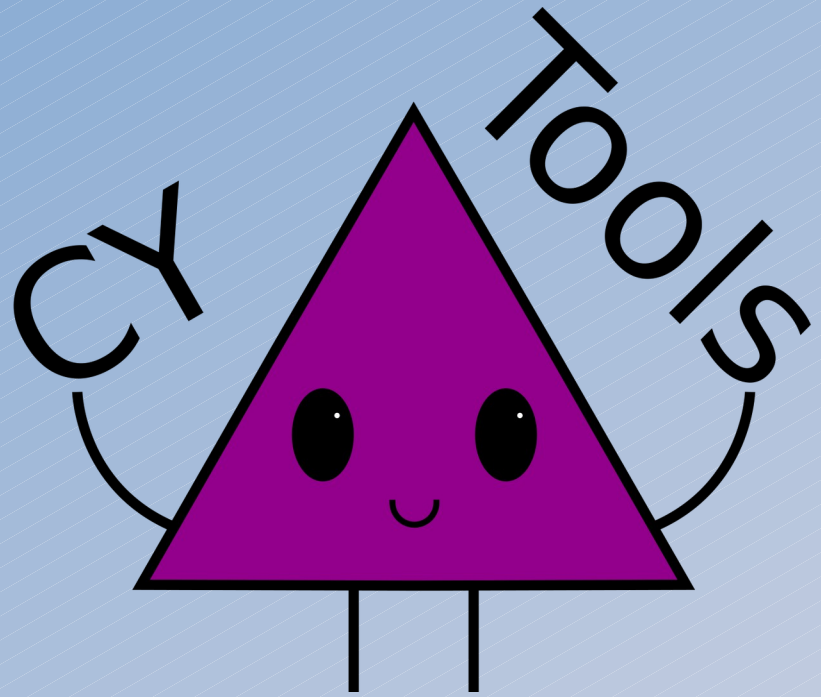


Exploring the KS database

- The complexity of the database grows exponentially with the number of points on the polytope ($\sim h^{1,1}$).
- The $h^{1,1} \gg 1$ region potentially contains most of the KS landscape.
- For context, in 2014 an immense effort and over 100,000 CPU hours were necessary to extract all CYs with $h^{1,1} <= 6$.
  [Altman, Gray, He, Jejjala, Nelson, 1411.1418]
- Even though PALP and Sage are fairly powerful packages, we abandoned them and developed almost everything from scratch.



Introducing CYTools



A software package for analyzing Calabi-Yau hypersurfaces in toric varieties.

Improvements for some key ingredients

1) 4D reflexive polytopes.

2) Fine, regular, star triangulations (FRSTs).

3) Intersection numbers.

4) Kähler cones.

5) Random sampling CYs.

Handling 4D reflexive polytopes

- PALP has been the go-to package for polytope computations, even in recent large-scale scans. [e.g. Huang, Taylor, 1907.09482]

Handling 4D reflexive polytopes

- PALP has been the go-to package for polytope computations, even in recent large-scale scans. [e.g. Huang, Taylor, 1907.09482]

- We use QHull/PPL, code by Volker Braun, and additional optimizations, to compute lattice points. Devised a super optimized face algorithm for 4D polytopes.

Handling 4D reflexive polytopes

- PALP has been the go-to package for polytope computations, even in recent large-scale scans. [e.g. Huang, Taylor, 1907.09482]

- We use QHull/PPL, code by Volker Braun, and additional optimizations, to compute lattice points. Devised a super optimized face algorithm for 4D polytopes.

- End result is our Python code being ~4x faster than PALP.

Handling 4D reflexive polytopes

- PALP has been the go-to package for polytope computations, even in recent large-scale scans. [e.g. Huang, Taylor, 1907.09482]
- We use QHull/PPL, code by Volker Braun, and additional optimizations, to compute lattice points. Devised a super optimized face algorithm for 4D polytopes.
- End result is our Python code being ~4x faster than PALP.
- When transcribed to Julia, our code is ~40x faster than PALP. We were able to obtain all points and faces of all reflexive polytopes overnight on a personal computer while working on [2008.01730].

Improvements for some key ingredients

1) 4D reflexive polytopes. ✓

2) Fine, regular, star triangulations (FRSTs).

3) Intersection numbers.

4) Kähler cones.

5) Random sampling CYs.

Finding appropriate triangulations

Batyrev's construction requires finding a finding a fine, regular, star triangulation (FRST) of the polytope.

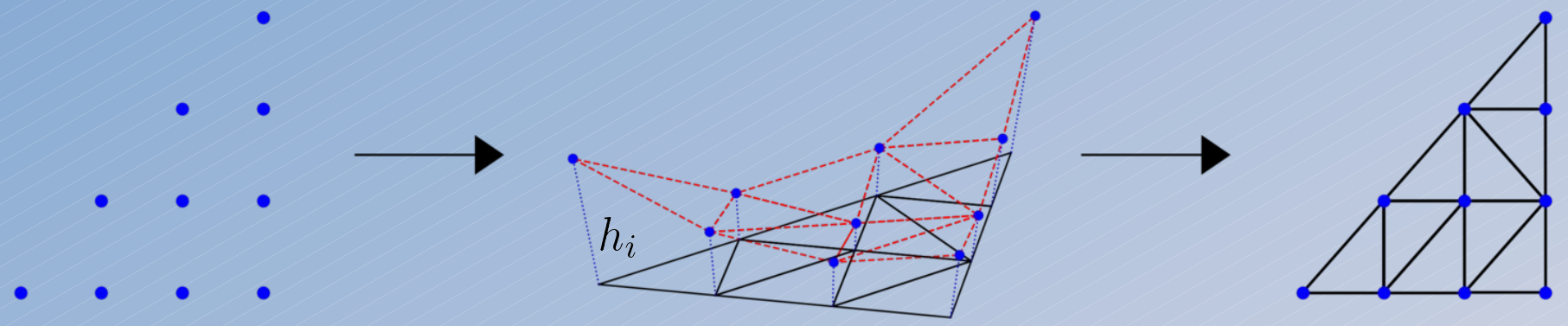
Star: Origin is vertex of all full-dim simplices → Simplices define cones of a fan

Fine: Uses all points (can exclude points interior to facets) → All singularities are resolved

Regular: Is needed so that the variety is Kähler, and will be key for our discussion. Let's see what it means more carefully.

Regular triangulations

A d-dimensional triangulation is regular if the points can be lifted by heights h_i to form a d+1-dimensional convex hull with the lower faces projecting down to the simplices of the triangulation.



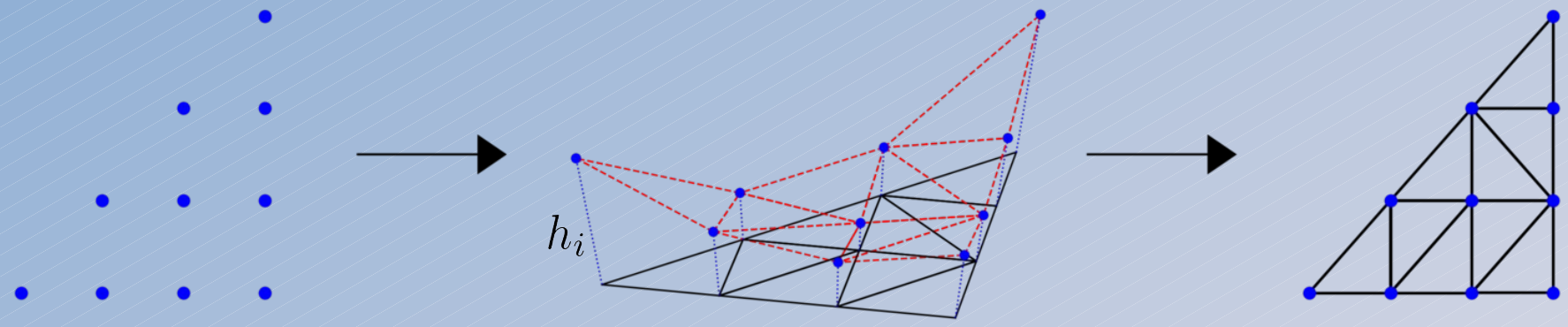
This property is the hardest to check, and is partly responsible for the difficulty of studying CYs with large $h^{1,1}$.

Finding triangulations efficiently

- Jörg Rambau's TOPCOM has been the main software used to find triangulations for a long time, but it struggles for large polytopes in the database, especially when requiring regularity.

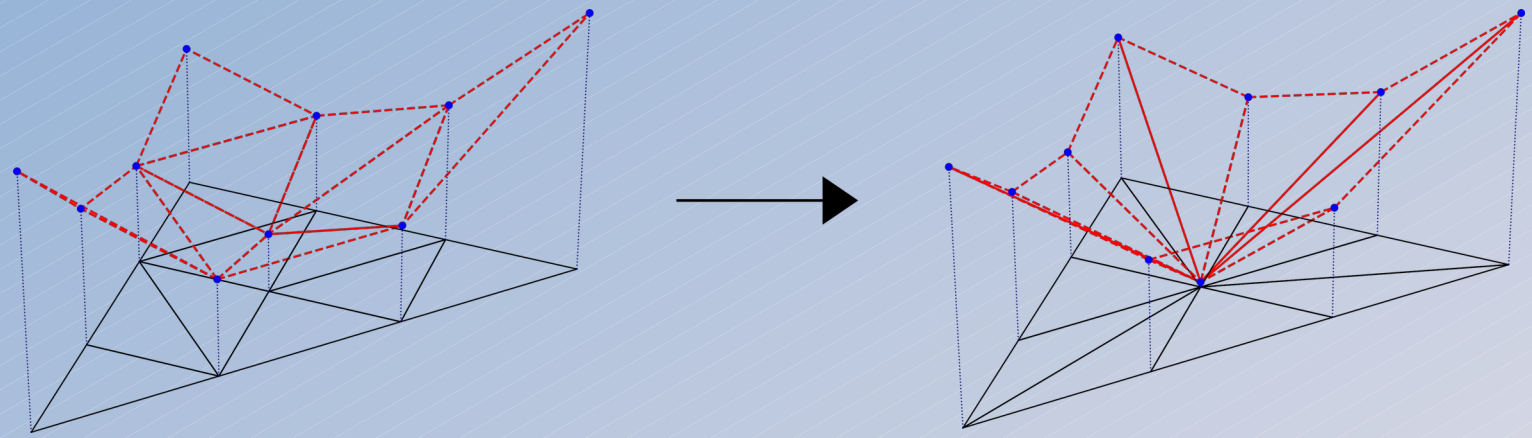
Finding triangulations efficiently

- Jörg Rambau's TOPCOM has been the main software used to find triangulations for a long time, but it struggles for large polytopes in the database, especially when requiring regularity.
- We use CGAL or QHull to construct regular triangulations by specifying heights.



Finding triangulations efficiently

- Jörg Rambau's TOPCOM has been the main software used to find triangulations for a long time, but it struggles for large polytopes in the database, especially when requiring regularity.
- We use CGAL or QHull to construct regular triangulations by specifying heights.
- The star condition can be enforced by using a much lower height for the origin.



Finding triangulations efficiently

- Jörg Rambau's TOPCOM has been the main software used to find triangulations for a long time, but it struggles for large polytopes in the database, especially when requiring regularity.
- We use CGAL or QHull to construct regular triangulations by specifying heights.
- The star condition can be enforced by using a much lower height for the origin.
- With appropriate choices of heights we can obtain FRSTs very efficiently.

Obtain one triangulation

Year	$h^{1,1}$	CPU time
2014	25	a few hours
2017	491	2s
2019	491	20ms

Finding triangulations efficiently

- Jörg Rambau's TOPCOM has been the main software used to find triangulations for a long time, but it struggles for large polytopes in the database, especially when requiring regularity.
- We use CGAL or QHull to construct regular triangulations by specifying heights.
- The star condition can be enforced by using a much lower height for the origin.
- With appropriate choices of heights we can obtain FRSTs very efficiently.
- This method does not allow a complete enumeration of triangulations, but this is impossible at large $h^{1,1}$.

Improvements for some key ingredients

1) 4D reflexive polytopes. ✓

2) Fine, regular, star triangulations (FRSTs). ✓

3) Intersection numbers.

4) Kähler cones.

5) Random sampling CYs.

Computing intersection numbers

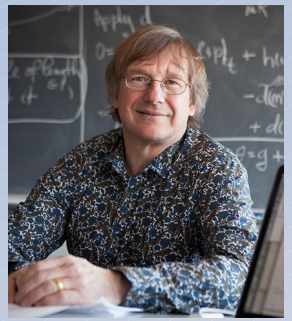
- Sage has been a common software for computing intersection numbers, but it struggles even at moderate $h^{1,1}$.

Computing intersection numbers

- Sage has been a common software for computing intersection numbers, but it struggles even at moderate $h^{1,1}$.
- There is a simple and computationally superior algorithm, as Mike Stillman explained to us: [c.f. e.g. 0803.1194]

  i. Compute intersection numbers of distinct divisors using full-dimensional cones in the fan.

  ii. Use linear relations of GLSM charge matrix to construct and solve equations for intersection numbers with self-intersections.

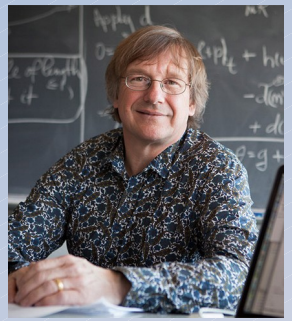


Computing intersection numbers

- Sage has been a common software for computing intersection numbers, but it struggles even at moderate $h^{1,1}$.
- There is a simple and computationally superior algorithm, as Mike Stillman explained to us: [c.f. e.g. 0803.1194]

  i. Compute intersection numbers of distinct divisors using full-dimensional cones in the fan.

  ii. Use linear relations of GLSM charge matrix to construct and solve equations for intersection numbers with self-intersections.



- We can find intersection numbers extremely quickly.

Compute intersection numbers

Year	$h^{1,1}$	CPU time
2017	100	30 mins
2018	491	30s
2019	491	3s

Improvements for some key ingredients

1) 4D reflexive polytopes. ✓

2) Fine, regular, star triangulations (FRSTs). ✓

3) Intersection numbers. ✓

4) Kähler cones.

5) Random sampling CYs.

Finding Kähler cones

- Finding the exact Kähler cone $\mathcal{K}_X$ of the CY is a well-know difficult problem.

Finding Kähler cones

- Finding the exact Kähler cone $\mathcal{K}_X$ of the CY is a well-know difficult problem.
- A common approximation is to use the Kähler cone $\mathcal{K}_V$ of the ambient toric variety, which can be obtained by an algorithm by Oda and Park, and further refined by Berglund, Katz and Klemm. [Oda, Park, '91] [Berglund, Katz, Klemm, hep-th/9506091]

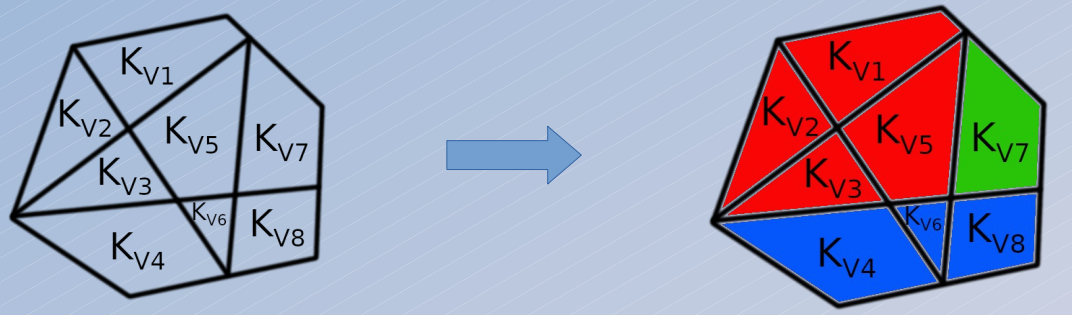
$$\mathcal{K}_V \subset \mathcal{K}_X$$

Finding Kähler cones

- Finding the exact Kähler cone $\mathcal{K}_X$ of the CY is a well-know difficult problem.
- A common approximation is to use the Kähler cone $\mathcal{K}_V$ of the ambient toric variety, which can be obtained by an algorithm by Oda and Park, and further refined by Berglund, Katz and Klemm. [Oda, Park, '91] [Berglund, Katz, Klemm, hep-th/9506091]

$$\mathcal{K}_V \subset \mathcal{K}_X$$

- There can be multiple FRSTs (i.e. ambient toric varieties) that give rise to the same CY. We can try to put all $\mathcal{K}_{V_i}$ together to better approximate $\mathcal{K}_X$.



Finding Kähler cones

- Finding the exact Kähler cone $\mathcal{K}_X$ of the CY is a well-know difficult problem.
- A common approximation is to use the Kähler cone $\mathcal{K}_V$ of the ambient toric variety, which can be obtained by an algorithm by Oda and Park, and further refined by Berglund, Katz and Klemm. [Oda, Park, '91] [Berglund, Katz, Klemm, hep-th/9506091]

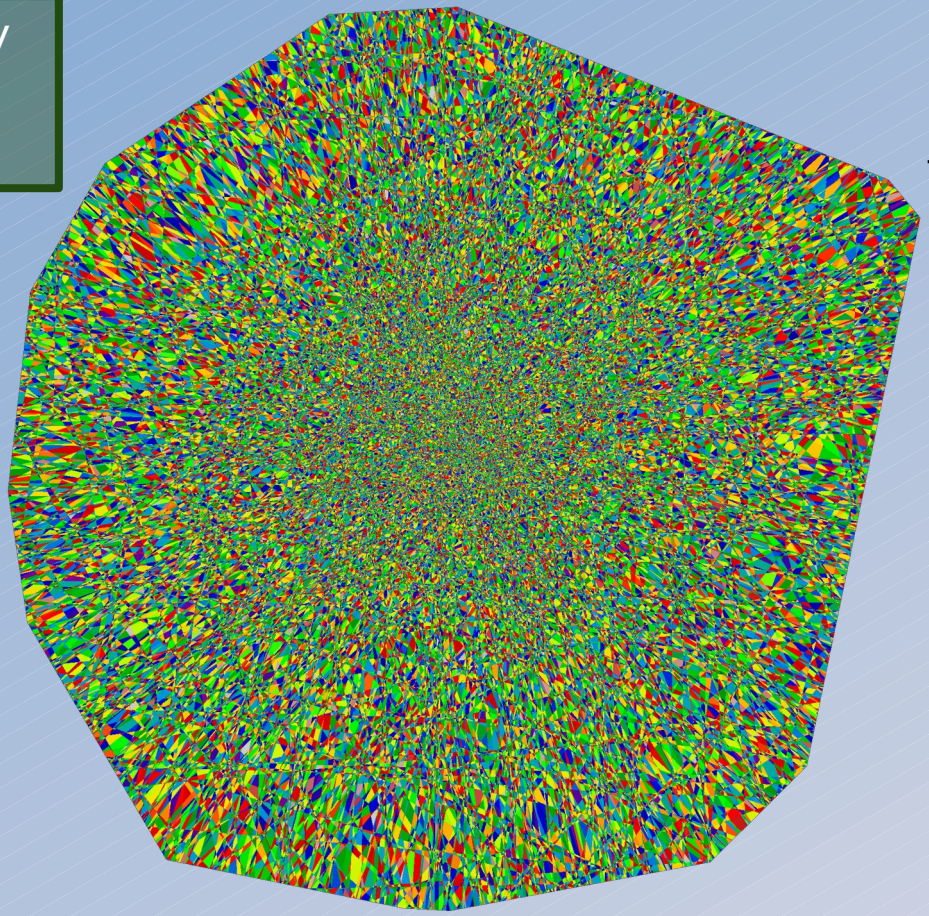
$$\mathcal{K}_V \subset \mathcal{K}_X$$

- There can be multiple FRSTs (i.e. ambient toric varieties) that give rise to the same CY. We can try to put all $\mathcal{K}_{V_i}$ together to better approximate $\mathcal{K}_X$.

$$\mathcal{K}_\cup := \bigcup_{\{V|X\subset V_i\}} \mathcal{K}_{V_i} \qquad \mathcal{K}_{V_i} \subset \mathcal{K}_\cup \subset \mathcal{K}_X$$

Visualizing the gluing of Kähler cones

2D slice of secondary fan of the largest 4D reflexive polytope

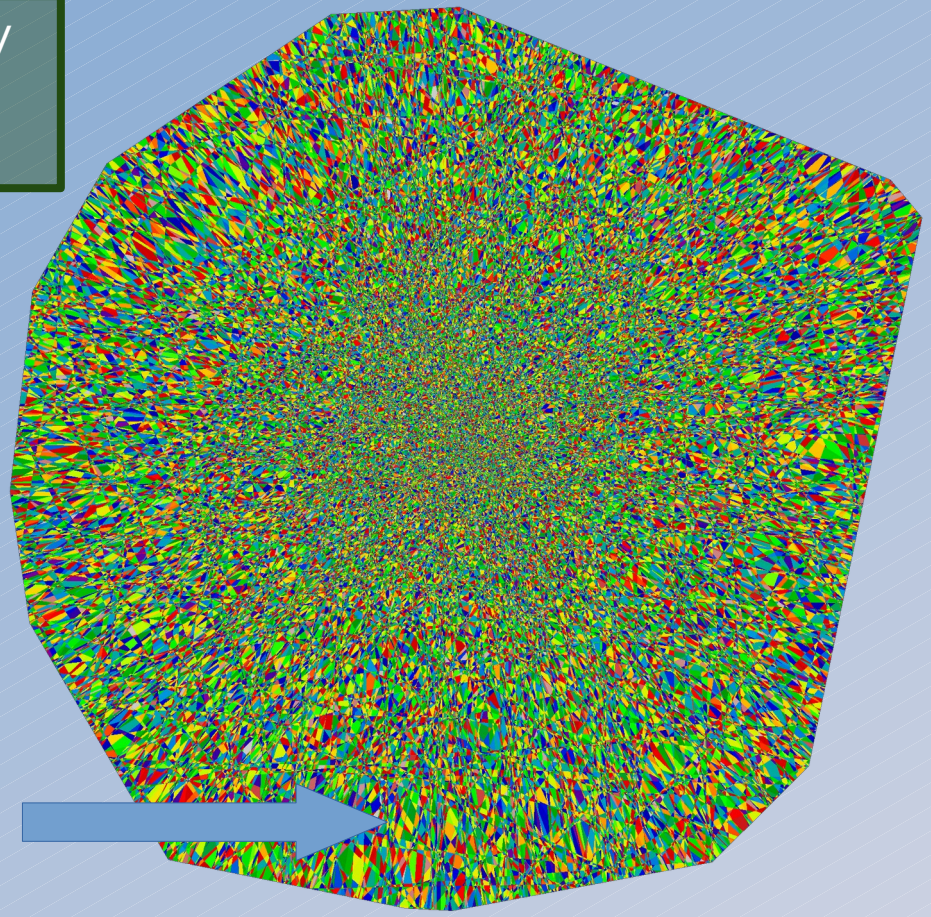


Note: To render this image we computed 16 million triangulations of the largest polytope!

Each colored region is the Kähler cone of a different toric variety

Visualizing the gluing of Kähler cones

2D slice of secondary fan of the largest 4D reflexive polytope

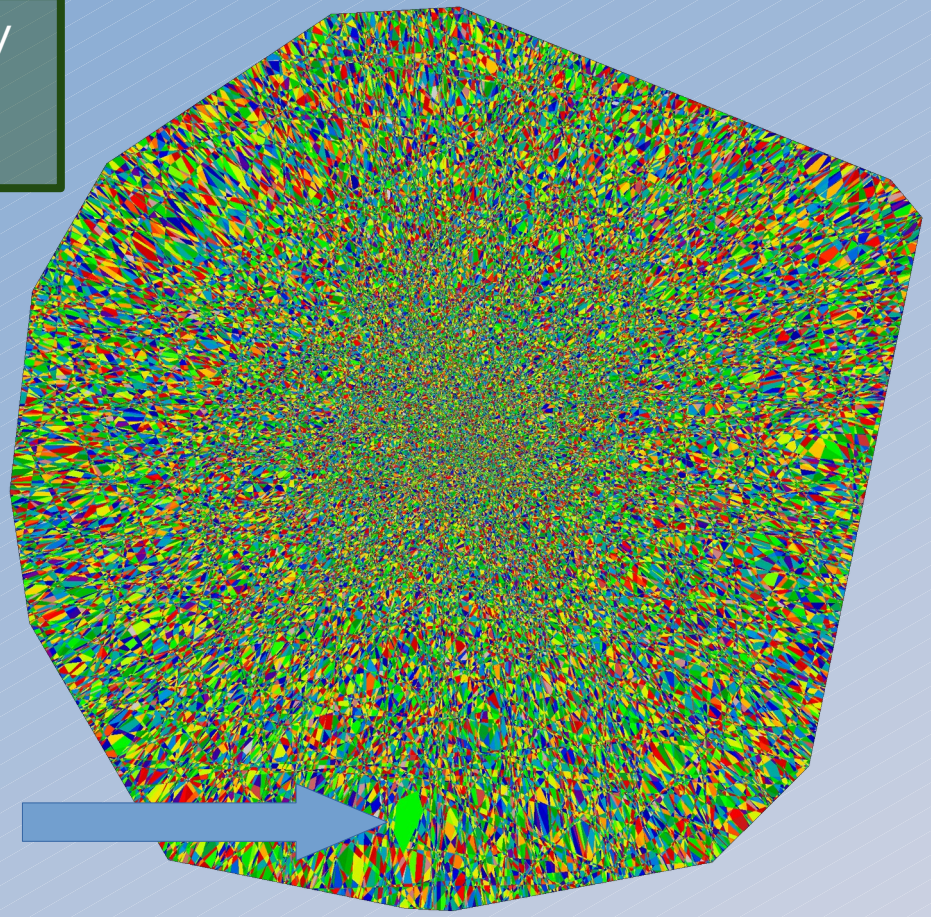


Each colored region is the Kähler cone of a different toric variety

Let us look at a CY sitting here and glue the Kähler cones of all ambient varieties with the same CY.

Visualizing the gluing of Kähler cones

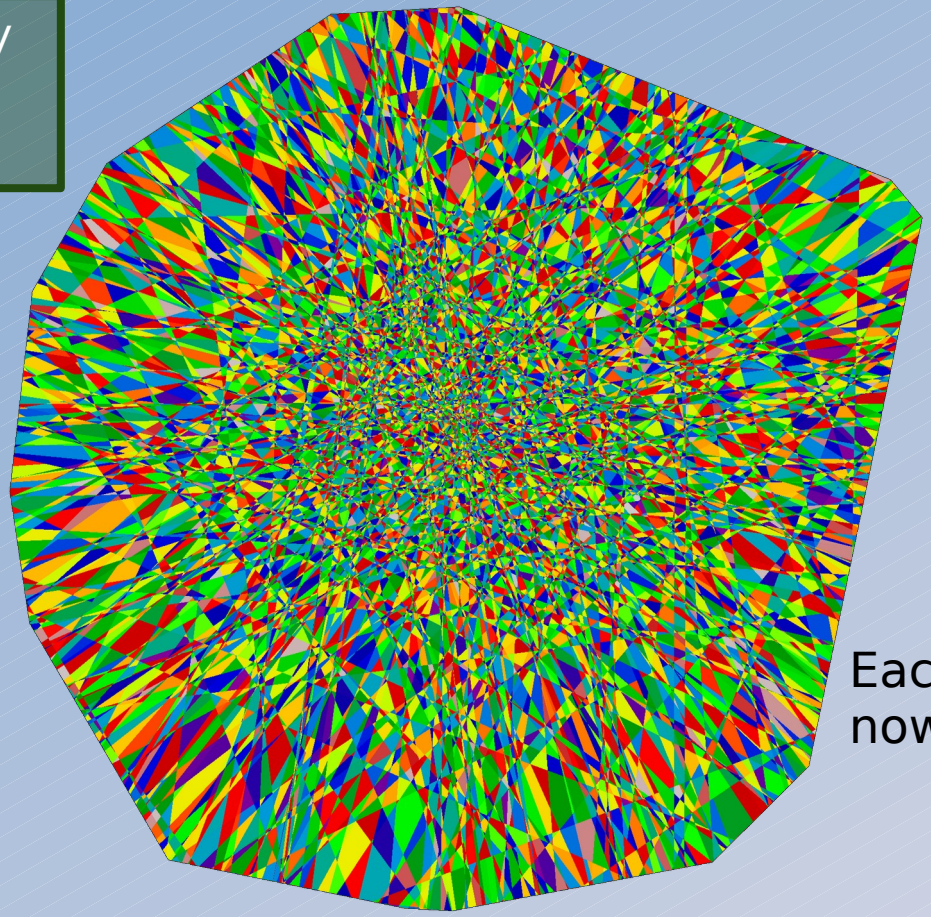
2D slice of secondary fan of the largest 4D reflexive polytope



Let us look at a CY sitting here and glue the Kähler cones of all ambient varieties with the same CY.

Visualizing the gluing of Kähler cones

2D slice of secondary fan of the largest 4D reflexive polytope



Each colored region is now $\mathcal{K}_{\cup}$ of a different CY.

Computing $\mathcal{K}_{\cup}$

- At small $h^{1,1}$ this can be done by finding all triangulations with the same CY and gluing the $\mathcal{K}_{V_i}$ of the ambient varieties.

  [e.g. Altman, Gray, He, Jejjala, Nelson, 1411.1418]

Computing $\mathcal{K}_{\cup}$

- At small $h^{1,1}$ this can be done by finding all triangulations with the same CY and gluing the $\mathcal{K}_{V_i}$ of the ambient varieties.

  [e.g. Altman, Gray, He, Jejjala, Nelson, 1411.1418]

- This becomes impossible very quickly since the number of triangulations grows exponentially.

Computing $\mathcal{K}_{\cup}$

- At small $h^{1,1}$ this can be done by finding all triangulations with the same CY and gluing the $\mathcal{K}_{V_i}$ of the ambient varieties.

  [e.g. Altman, Gray, He, Jejjala, Nelson, 1411.1418]

- This becomes impossible very quickly since the number of triangulations grows exponentially.

- With a good understanding of the secondary fan, we have managed to extend the Oda-Park/Berglund-Katz-Klemm algorithm so that $\mathcal{K}_{\cup}$ can be computed directly even at large $h^{1,1}$. (Work in progress)

Improvements for some key ingredients

1) 4D reflexive polytopes. ✓

2) Fine, regular, star triangulations (FRSTs). ✓

3) Intersection numbers. ✓

4) Kähler cones. ✓

5) Random sampling CYs.

Random sampling CYs

- Obtaining fair samples of CYs at large $h^{1,1}$ is difficult due to the astronomic number of possible triangulations.

Random sampling CYs

- Obtaining fair samples of CYs at large $h^{1,1}$ is difficult due to the astronomic number of possible triangulations.

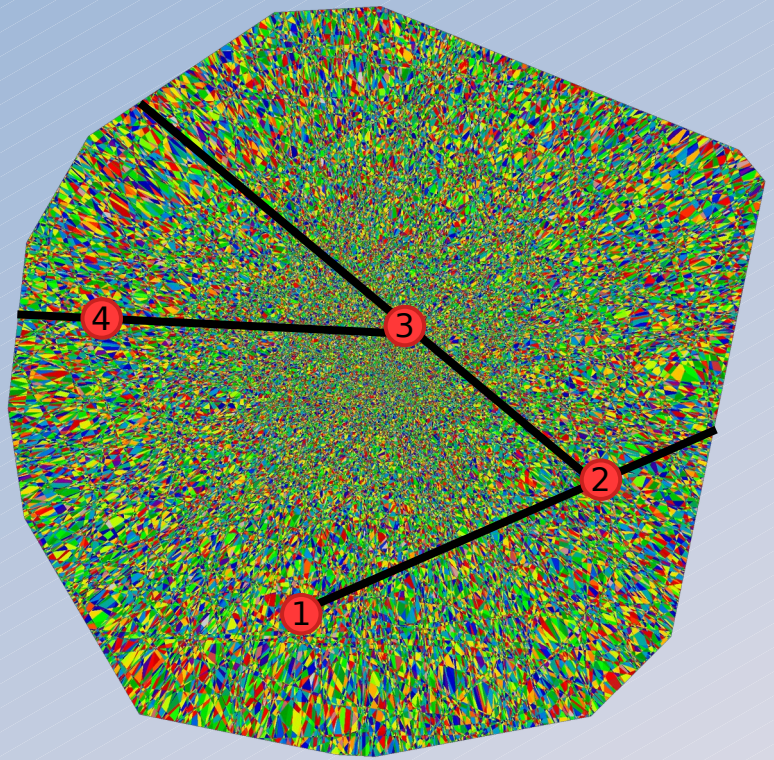
- The typical method of applying random bistellar flips is very slow and is biased towards the starting triangulation.

Random sampling CYs

- Obtaining fair samples of CYs at large $h^{1,1}$ is difficult due to the astronomic number of possible triangulations.

- The typical method of applying random bistellar flips is very slow and is biased towards the starting triangulation.

- The way we solve this is by combining random walks in the secondary fan with random bistellar flips. At small $h^{1,1}$ we verify that the sampled distribution is representative of the full set.
  [Demirtas, McAllister, A.R.T., 2008.01730]



Improvements for some key ingredients

1) 4D reflexive polytopes. ✓

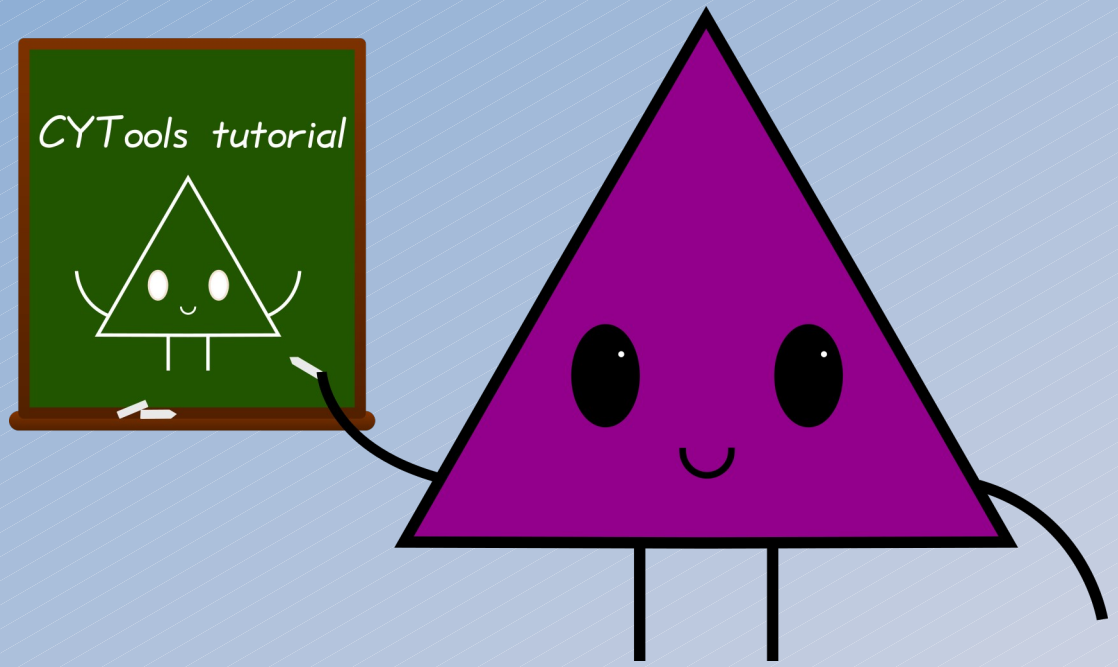
2) Fine, regular, star triangulations (FRSTs). ✓

3) Intersection numbers. ✓

4) Kähler cones. ✓

5) Random sampling CYs. ✓

Live demo/tutorial



Conclusions

- We have developed tools that not only significantly outperform previous packages, but allow previously impossible computations.

Conclusions

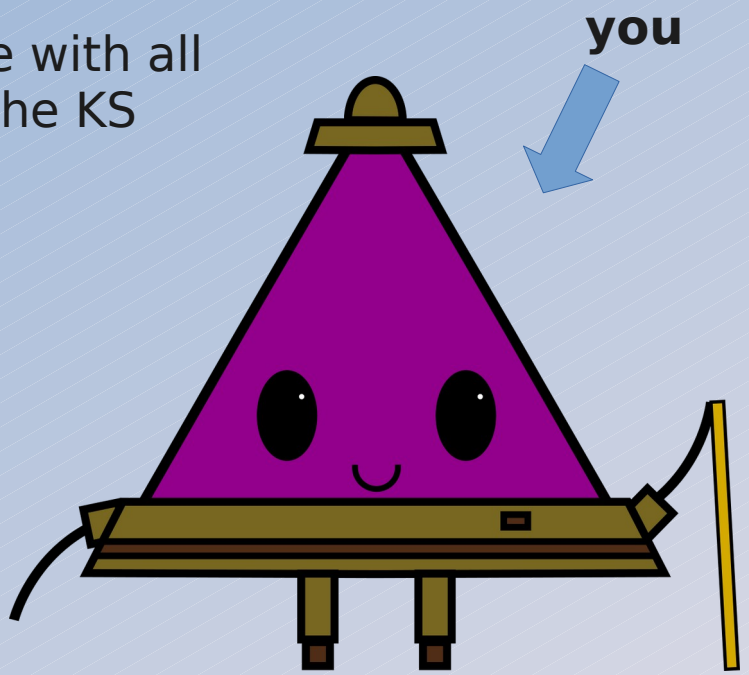
- We have developed tools that not only significantly outperform previous packages, but allow previously impossible computations.

- We will soon release an open-source package with all of these tools so that everyone can explore the KS landscape with ease.

Conclusions

- We have developed tools that not only significantly outperform previous packages, but allow previously impossible computations.

- We will soon release an open-source package with all of these tools so that everyone can explore the KS landscape with ease.

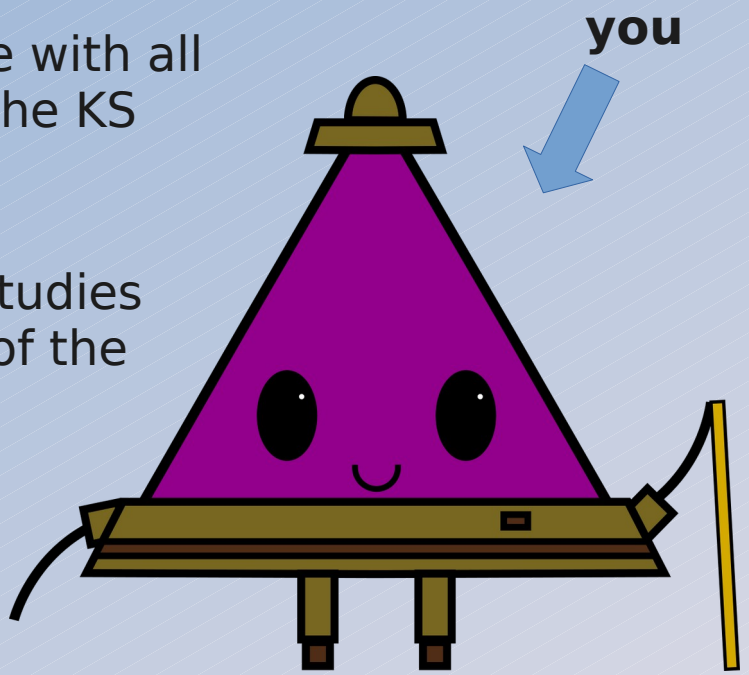


Conclusions

- We have developed tools that not only significantly outperform previous packages, but allow previously impossible computations.

- We will soon release an open-source package with all of these tools so that everyone can explore the KS landscape with ease.

- We hope our package will allow systematic studies of string vacua, and a better understanding of the landscape.



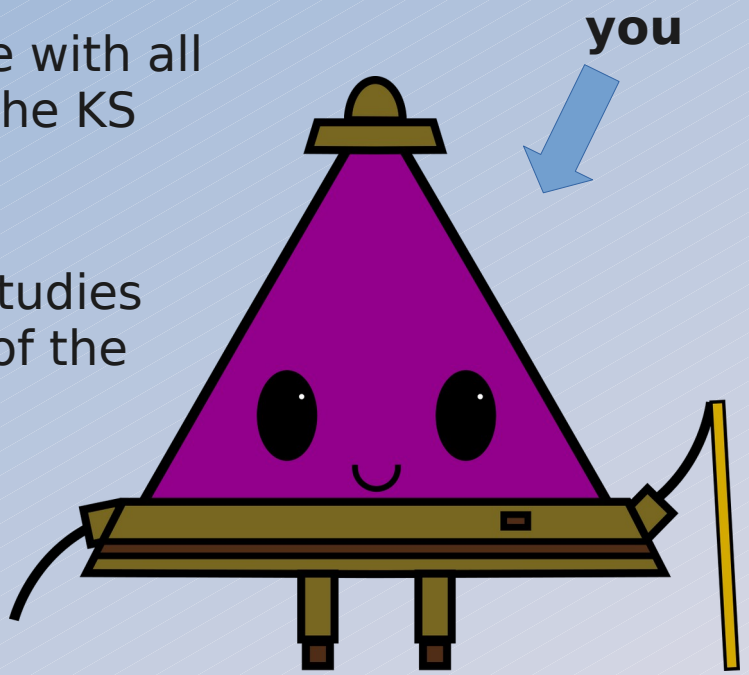
Conclusions

- We have developed tools that not only significantly outperform previous packages, but allow previously impossible computations.

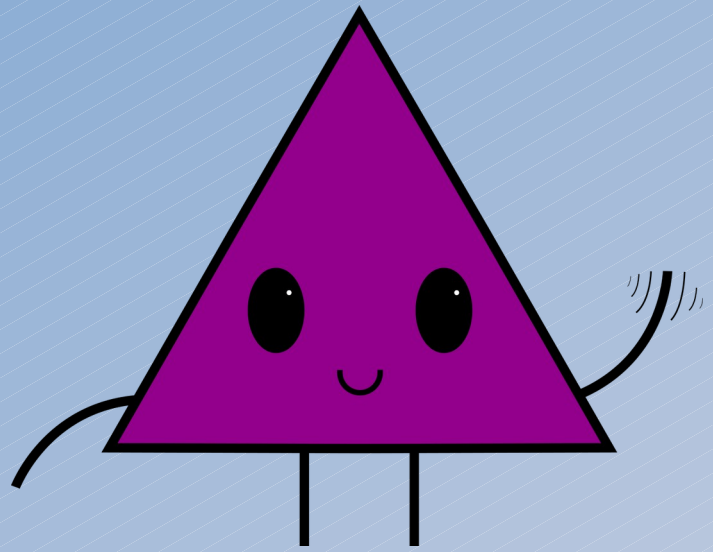
- We will soon release an open-source package with all of these tools so that everyone can explore the KS landscape with ease.

- We hope our package will allow systematic studies of string vacua, and a better understanding of the landscape.

If you want to be a beta tester you can contact either Mehmet Demirtas or me.



Thank you!



Questions?