Random Monomial Ideals: A Macaulay2 Software Package

Genevieve Hummel, Parker Joncus, Daniel Kosmas, Richard Osborn, Monica Yun, and Tanner Zielinski

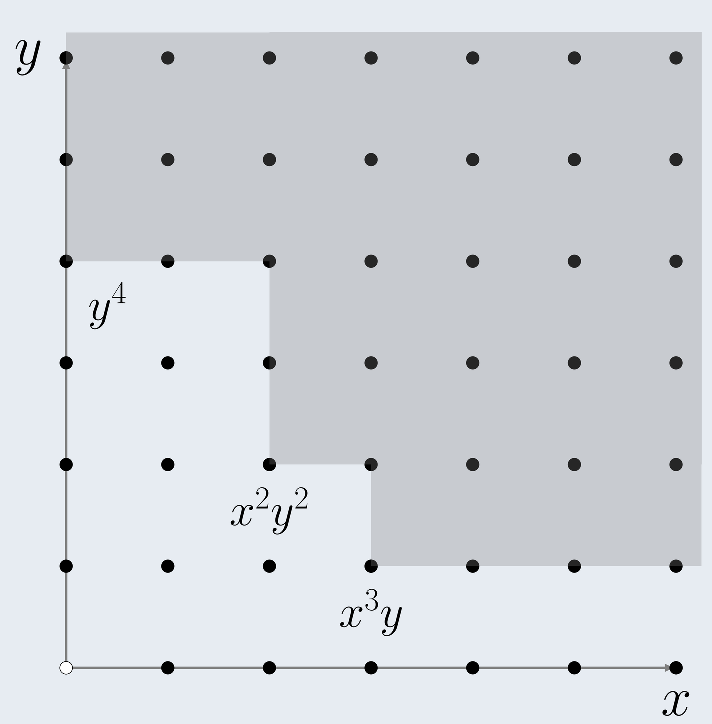
Advisors: Sonja Petrović, Despina Stasi, and Dane Wilburne, Illinois Institute of Technology

Macaulay2

Macaulay2 is an open source software system devoted to supporting research in algebraic geometry and commutative algebra. It includes core algorithms for computations in those fields, such as Gröbner bases algorithms for computing a fundamental generating set of a polynomial ideal. It's easy to contribute packages to via GitHub, and there is a huge development community.

Monomial Ideals

A *monomial ideal* in the polynomial ring $S = k[x_1, \dots, x_n]$ is an ideal I that can be generated by monomials.



Staircase representation of the monomial ideal $I = \langle y^4, x^2y^2, x^3y \rangle$.

Motivation

Monomial ideals:

- capture key information about general ideals
- play a fundamental role in commutative algebra

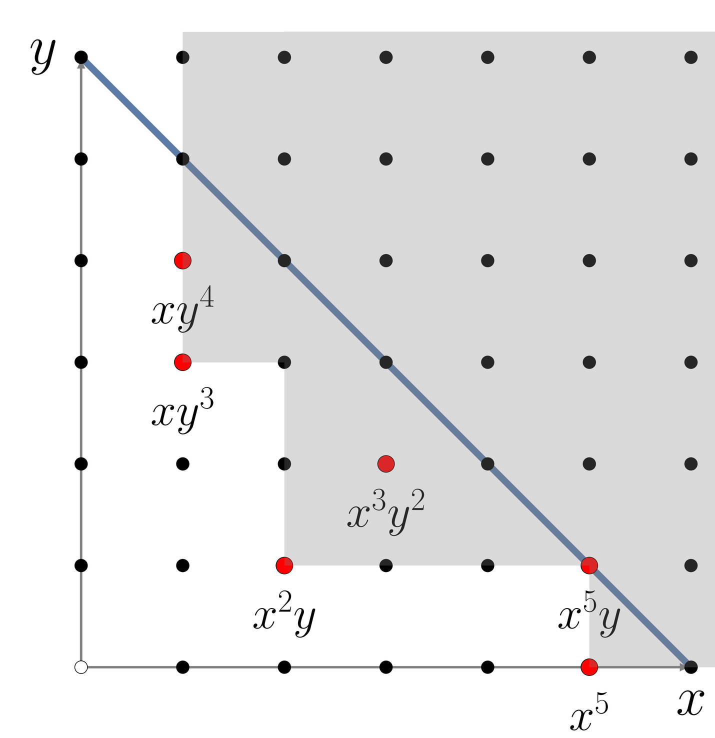
Random monomial ideals also generalize random graphs, hypergraphs, and simplicial structures. By studying random monomial ideals, we can understand the "average monomial ideal", and related combinatorial structures.

The Three Models

The ER Model [1]: Include each monomial x^α of degree at most D in $\mathfrak{B}$ independently with probability $p = p(n, D)$, then set $\mathfrak{I} = (\mathfrak{B})$.

The Fixed M Model: Choose M monomials from the set of all monomials x^α of degree at most D to include in $\mathfrak{B}$, then set $\mathfrak{I} = (\mathfrak{B})$.

The Graded Model: Either ER Model or Fixed M Model, where the probability of including a monomial x^α of degree i is $p_i = p(n, i)$ for all $1 \le i \le D$, or choose M_i monomials from the set of all monomials of degree i for all $1 \le i \le D$.



Probability Param.: $p = \frac{1}{7}$

Or Num. Monomials: $M = 6$

Maximum degree: $D = 6$

Random monomials: $\mathfrak{B} = \{xy^4, xy^3, x^2y, x^3y^2, x^5y, x^5\}$

Random monomial ideal: $\mathfrak{I} = \langle xy^3, x^2y, x^5 \rangle$

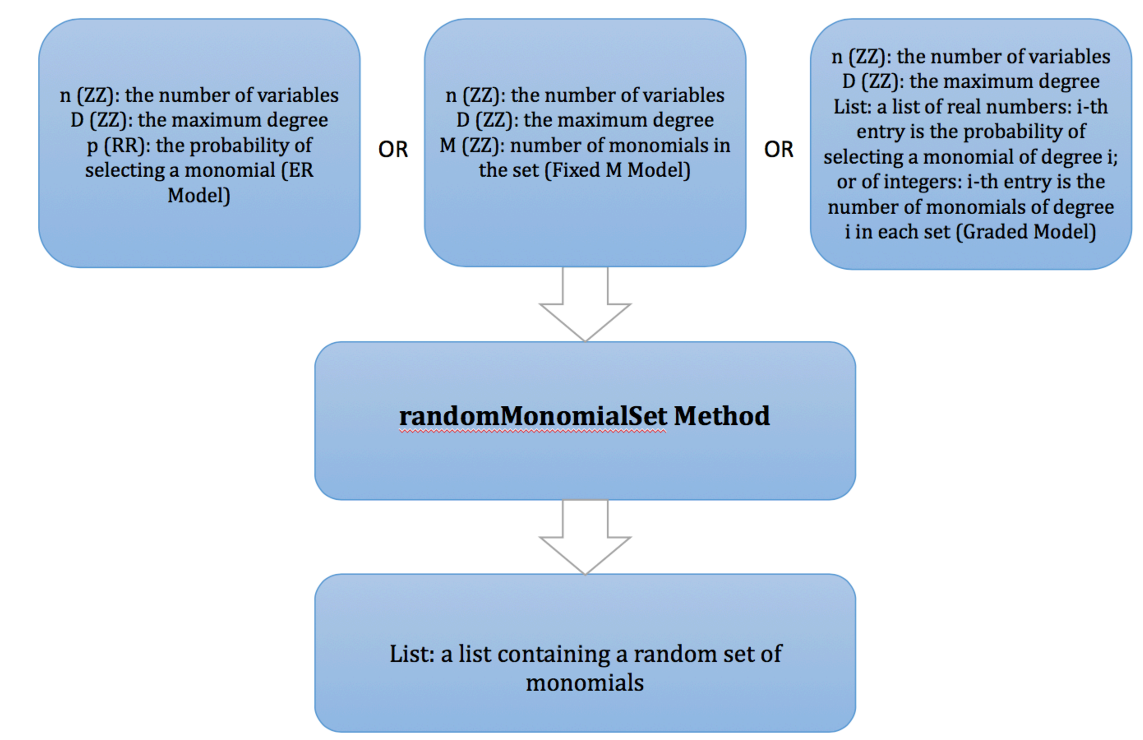
A Faster Process

The standard process selects monomials from the list of all possible monomials up to our degree bound, often causing us to randomly select redundant monomials. These monomials already belong to the monomial ideal, such as x^3y^2. This can bog down computation time for large n and D.

By selecting monomials of a fixed degree d, then taking the quotient of the ring by the monomial ideal generated by those monomials, we guarantee that no multiples of any selected monomial can be produced. This produced a different probability distribution on the bases, but not on the ideals.

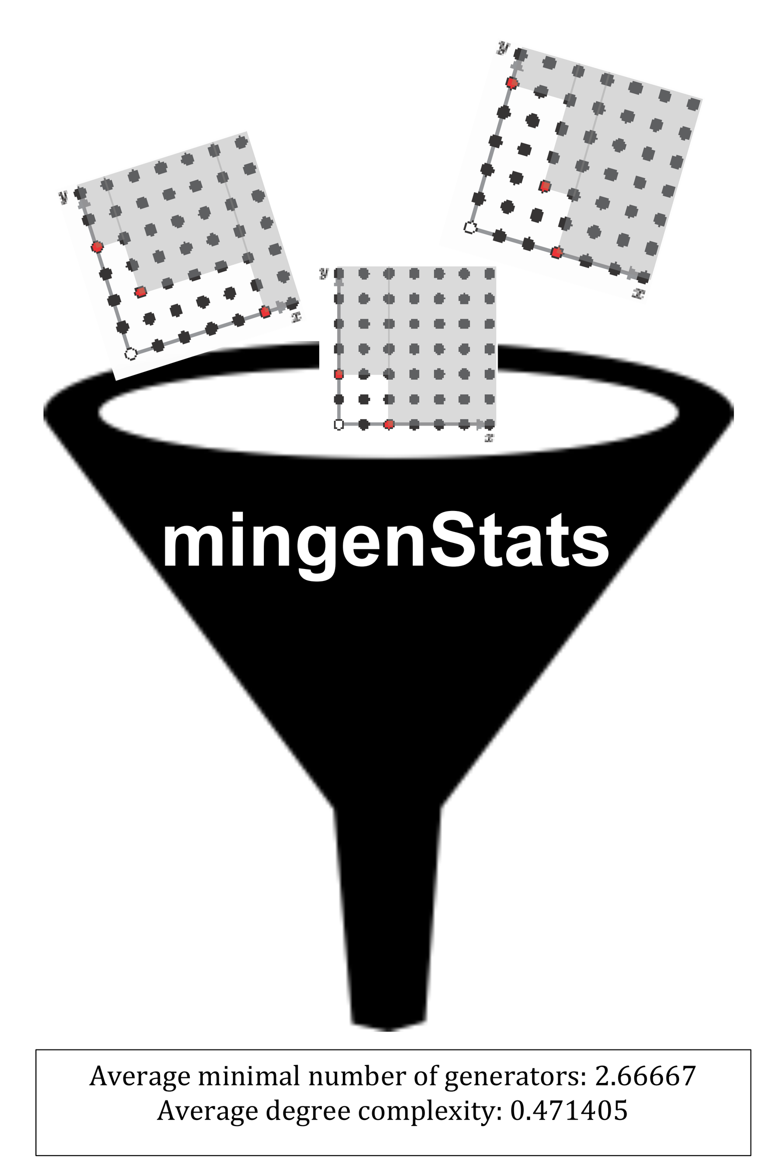
The Basic Method: randomMonomialSet

`randomMonomialSet` generates a random set of monomials based on one of the three models.



mingenStats

`mingenStats` computes statistics on the minimal generators of a list of monomialIdeals: number and degree complexity



Overview of Methods [2]

- `randomMonomialSet` randomly generates a list of monomials in fixed number of variables up to a given degree
- `randomMonomialIdeals` randomly generates monomial ideals in fixed number of variables, with each monomial up to a given degree
- `idealsFromGeneratingSets` creates ideals from sets of monomials
- `mingenStats` statistics on the minimal generators of a list of monomialIdeals: number and degree complexity

Note: There are many similar methods that produce statistics on different characteristics of the ideals:

- dimension,
- degree,
- regularity,
- Betti numbers

The corresponding method names are xxxxxStats

Future Work

- Generalize the ER model to Laurent monomials
- Develop models for randomly generating toric ideals
  - Model 1: Randomly subtract pairs of monomials.
  - Model 2: Create a design matrix from the selected monomials and determine the toric ideal from its kernel.
- Generalize the β-model for random graphs
  - Each variable in the model is assigned a weight
  - Monomials are selected based off the weight of their support
- Improve Macaulay2's handling of files
  - Improve handling of file pointers
  - Interface for streaming data from a file

References

[1] De Loera, J. A., Petrović, S., Silverstein, L., Stasi, D., and Wilburne, D. 'Random Monomial Ideals.' (2017) Submitted. arXiv: 1701.07130.

[2] Petrović, S., Stasi, D., Wilburne, D., Hummel, G., Joncas, P., Kosmas, D., Osborn, R., Yun, M., and Zielinski, T. RandomMonomialIdeals. (2017) [Online]. Available: https://github.com/RandCommAlg/RMI/