EASAL (Efficient Atlasing, Analysis and Search of Molecular Assembly Landscapes)

Aysegul Ozkan* Meera Sitharam*

Abstract

We leverage a mix of classical concepts such as stratifications of semialgebraic sets, and recent theoretical results concerning configuration spaces with a convex parametrization. These lead to a key observation that most regions of assembly and packing configuration spaces indeed have a convex parametrization. In this they differ starkly from configuration spaces used for folding, or structure determination, for example. This observation leads to (1) a novel, efficient and intuitive representation of configuration spaces which we call the Atlas; (2) an efficient algorithm for generating the Atlas and sampling the configuration space. The latter uses recent algorithms for efficiently realizing geometric constraint systems.

1 Introduction

It is a longstanding problem to efficiently and intuitively describe and predict the geometric structure and properties of high dimensional molecular assembly or packing configuration spaces. This leads to long open problems. A satisfactory answer to these and other related question requires an efficient and intuitive description and prediction of how to (i) determine the configurational entropy, a type of weighted volume, which determines free energy during the assembly of packing process (ii) isolate those intermolecular interactions that are crucial for successful assembly pathways, which are heavily influenced by entropy considerations.

Different modeling goals related to assembly require different levels of refinement during analysis, searching, sampling and visualization of configuration spaces. A satisfactory method should possess the following features: (a) provide intuitive and explicit relationships between the input molecular data and the geometric properties of the configuration space; (b) provide quantitative accuracy guarantees derivable from the input data, including running time estimates; (c) flexibly scale down effort at a lower refinement, but preserve key features of the configuration space structure such as lower dimensional boundaries – these often include highly probable regions of the configuration space; (d) be computationally efficient; (e) the visualization, GUI and other functionalities should be intuitive for the biophysics, biochemistry or structural biology user.

Molecular assembly or packing configuration spaces are specified by known intermolecular interactions between a collection of constituent molecular units. These include weak forces, hydrogen bonds, steric constraints, tethering constraints as well as global energy and symmetry constraints. These interactions, can, with some work, be represented as static, geometric constraints such as distance and angle intervals between geometric primitives that are used to represent molecular units.

Configurational entropy of a collection of molecular units can be viewed as the weighted volume of the configuration space. Here each configuration is weighted by the probability distribution that specifies its likelihood of occurrence during the relevant process. Together with energy values, this determines the free energy.[5]

Molecular dynamics methods mixed with specially designed energy functions are most commonly used for problems (i) and (ii), but they are computationally very intensive and one size fits all. For instance, they do not exploit the special properties of assembly or packing configuration spaces as opposed to folding configuration spaces. If run for long enough, starting from sufficiently many initial configurations, they sample and explore all likely regions of the configuration space, giving a reasonable estimate of configurational entropy. However, the requirements (a), (b) and (d) are not met by such algorithms.

Other common methods for sampling or exploring configuration spaces, such as Monte Carlo mixed with constraint resolution and/or energy minimization by gradient descent are more efficient than molecular dynamics, but often go outside the feasible region and discard many samples, which hurts their efficiency. Furthermore, by their nature, these methods cannot guarantee uniform sampling of the configuration space, and since they are not informed by true dynamics, repeat sampling is not consistent with more probable configurations. Overall, the requirements (a), (b), (c) are not met by such algorithms.

EASAL algorithms and software (to be opensource, available upon request) have been specifically designed to satisfy (a), (b), (c), (d), (e) and answer (i) and (ii). EASAL is currently being validated on AAV virus assembly data from the lab of Mavis Agbandje-Mckenna at the University of Florida.


## 2 Contribution and Organization

The new contributions are based on a classical concept of *stratification* of semialgebraic sets and recent theoretical and algorithmic results on: (1) configuration spaces (of geometric constraint systems) that have convex so called Cayley parametrization and how to obtain good description and bounds for them and (2) decomposition of geometric constraint systems and optimizing the algebraic complexity of solving or realizing them; i.e., to convert a parametrized Cayley configuration into the standard cartesian configurations.

We develop the notion of Atlas of a stratified configuration space for an assembly system. The Atlas consists of carefully parametrized, convex regions that correspond to the regions of the stratification.

Our new method to find and sample the Atlas at a desired level of refinement shows promise for the size and type of configuration spaces that arise in packing or assembly settings. Specifically, we have shown that these recent theoretical and algorithmic ingredients make it significantly simpler and intuitive to approach the assembly problem than to approach the conformational or structure determination and folding problem.

Section 3 gives the required definitions and theory including recent results that are being leveraged, as well as the new observation that most regions of assembly configuration spaces have convex Cayley parametrizations because they are specified by active constraint graphs that belong to a special class called 3-realizable graphs which include a well-known class called partial 3-trees, or graphs with tree-width 3; this section also describes two new concepts - Charts and Atlas of configuration spaces. Section 4 gives the new algorithm for sampling the assembly configuration space for the case of 2 molecular units only. The concluding Section 5 lists straightforward extensions of our algorithm as well as theoretical guarantees, including complexity. Screenshots of a running example of packing 2 toy molecules obtained from the current EASAL implementation are used to illustrate the concepts throughout the paper.

## 3 Theory: Stratifications and Atlases of Assembly configuration spaces

An *assembly or packing constraint* system consists of the following.

A collection of globally rigid *molecular units*, each represented as the internal cartesian coordinates of a collection of *atomic units*, which are in turn represented as points/spheres or lines/cylinders.

A set of *intermolecular assembly or packing constraints*, of 3 general types. a) A *local atomic assembly constraint* is specified as a distance and/or angle bound or interval between a pair of atomic units in different molecules. These

![Figure 1: Easal screenshot: Strata of different dimensions consisting of active constraint regions in stratification of assembly constraint system shown in inset. Nodes are the active constraint regions and edges show containment in a parent region one dimension up.](Image)

represent steric constraints, vanderwaals and weak force interactions. b) A *pairwise molecular tether constraint* is specified between a pair of molecules by giving a set of pairwise distance upper bounds between pairs of atomic units, one in each of the molecules and stipulating that at least one of these distance upper bounds is met and a *composite molecular tether constraint* is specified between a composite of several molecules by stipulating that a tree of pairwise molecular tether constraints must be satisfied. c) A *global assembly constraint* is specified as a bound on some (e.g. energy) function of (the cartesian coordinates) of a configuration of the given collection of rigid molecular units. Next we introduce the *stratification* of an assembly configuration space used in this paper. Note that our assembly configuration spaces are semi-algebraic sets: the variables are the coordinates of the atomic units internal to molecular composite. A configuration is in fact a solution to a system of quadratic polynomial inequalities. This is because each local assembly constraint asserts a distance/angle value (equality) or a distance/angle interval (two inequalities) between the positions of the participating two atomic units.

**Definition 3.1.** Consider an assembly configuration space $\mathcal{A}$ of $k$ rigid molecular units $r_i$, defined by a system $A$ of assembly constraints. The configuration space of the composite has dimension $m \leq (k - 1) \ast 6$, the number of internal degrees of freedom of the composite. For $k = 2$, $m$ is at most 6 and in the presence of a composite bi-tether constraint, it is at most 4. Here 6 is the number of rotational and translational degrees of freedom of a rigid object in 3 dimensional Euclidean space.

A *stratification* of the configuration space $\mathcal{A}$ is a partition of the space into regions grouped into strata. Starting with a filtration of nested, closed strata $X_i$ of $\mathcal{A}$,
any fixed value of distances for assembly of 2

Figure 2: All possible well-constrained active constraint graphs for assembly of 2 molecular units, hence active constraint graphs are all possible subgraphs of these.

∅ ⊂ X₀ ⊂ X₁ ⊂ . . . ⊂ Xₘ = A where m = (k − 1)/6. Each Xᵢ is a union of nonempty closed regions R_Q where a set Q ⊆ A, |Q| = m − i inequality constraints are active, i.e., equality is attained for these constraints, and they are independent. These nonempty regions are called active constraint regions. See Figure 1. Each such active constraint set Q is itself part of at least one, possibly many, nested chains of the form ∅ ⊂ Q₀ ⊂ Q₁ ⊂ . . . ⊂ Qₘ = Q ⊂ . . . ⊂ Qₘ. Here the chains in which Q participates are indexed by l. See Figure 4.

This gives rise to corresponding reverse nested chains of active constraint regions R_Qₗ of the form: ∅ ⊂ R_Qₐ₁ ⊂ R_Qₐ₂ ⊂ . . . ⊂ R_Qₘ = R_Q ⊂ . . . ⊂ R_Qₘ. Note that here for all l, j, R_Qₗ ⊊ X_j and are closed and j dimensional. We use the word active constraint region associated with the active constraint set Q to refer to the closed regions R_Q.

We represent the active constraint system as a graph with vertices representing the participating atomic units (at least 3 in each molecular unit) and edges representing the active constraints between them. Between a pair of molecular units, there are only a small number of possible active constraint graph isomorphism types (all have at most 12 vertices) as shown in Figure 2.

Definition 3.2. A graph is d-realizable if for every possible distance value assignment to its edges, if it is Euclidean realizable, i.e., there is a positioning of the vertices in any Euclidean dimension satisfying the given distance values, then it is realizable in d dimensional Euclidean space.

Next we define the notion of inherently convex Cayley configuration space for a distance constraint graph.

Definition 3.3. A distance constraint graph G = (V, E) has an inherently convex 3d Cayley configuration space if the following holds. Take any partition of edges E = H ⊔ F, and any fixed value of distances d_F associated with F, and any distance inequalities d_H associated with H. Define the set Φ_H(G, F, d_F, d_H) as the set of all possible values of squared-distances for H attained by Euclidean realizations or configurations of the vertices of G, satisfying the constraints d_F and d_H. Now Φ_H(G, F, d_F, d_H) is convex. Note that each point in Φ_H(G, F, d_F, d_H) corresponds to at least one, but potentially many Euclidean realizations or configurations of the distance constraint system given by G and d_F, d_H. However, if G is generically well-constrained (sometimes called minimally rigid [4]), then for every point in Φ_H(G, F, d_F, d_H), the corresponding set of realizations is generically finite. See Figure 3.

Definition 3.4. Extend an active constraint graph, or any distance constraint graph G_F = (V, F) system by adding edge set H to give an extended graph G = (V, E = H ⊔ F). If this extended graph G has an inherently convex 3d Cayley configuration space, then the corresponding active constraint region R_G_F, when parametrized by the squared-distance or Cayley parameters associated with the edges H, is guaranteed to be convex. This parametrized region is just Φ_H(G, F, d_F, d_H), from Definition 3.3 and, if G is additionally well-constrained, it is called an exact convex Chart of the active constraint region R_G_F, using parameters H. See Figure 3 and 5.

Next we state the crucial convex parametrization theorem of [9], which tells us a necessary condition for active
Theorem 3.6. A nonconvex or badly disconnected one. See Figure 6.

Theorem 3.5. [9] A 3-d distance constraint graph $G = (V, E)$ has an inherently convex Cayley configuration space if and only if it is 3-realizable.

A natural class of 3-realizable graphs called partial 3-trees in fact occur most often as active constraint graphs. These graphs can be defined in a recursive way using so-called 3-sums. A complete 3-tree is defined as follows. Take two complete 3-trees and paste them along a common triangle. Or start with a triangle and at each step, add a new vertex that is adjacent by edges to 3 old vertices that form a triangle. A partial 3-tree is obtained from a complete 3-tree by removing some edges. The next theorem also from [9] indicates how to choose the parameters to obtain an exact convex Chart for an active constraint region corresponding to a partial 3-tree and how to compute its description and bounds. The choice of parameters is extremely crucial: the paper [9] gives elementary examples that illustrate how one choice of parameters gives a convex Chart and another could give a nonconvex or badly disconnected one. See Figure 6.

Theorem 3.6. If an active constraint graph $G_F = (V, F)$ is a partial 3-tree, then by adding edge set $H$ to give a complete 3-tree $G = (V, E = H \cup F)$, we obtain an exact convex Chart $\Phi_H(G, F, d_F, d_H)$, of the active constraint region $R_{G_F}$, using the parameters $H$. The exact convex Chart $\Phi_H(G, F, d_F, d_H)$ has a linear number of boundaries in $|G|$ that can be output as implicit quadratic polynomial equalities in linear time. In fact, the explicit bounds of each parameter in $H$, in sequence, given the values of the preceding parameters, can be computed in quadratic time in $|G|$. Since $G$ is of constant size (at most 12 vertices) when the assembly problem is restricted to 2 molecular units.

Better bounds are given in [2]. A majority of active constraint graphs are 3-realizable even partial 3-trees, see Figure 2. For 3-realizable graphs that are not partial 3-trees (where

Definition 3.7. An Atlas of an assembly configuration space is a representation of its stratification into active constraint regions: each active constraint region is represented by its active constraint graph, its exact convex or optimally tight Chart together with the parameters used for obtaining the Chart. See Figure 1 and Figure 6.

The Cayley configurations in the Atlas need to be converted to cartesian realization. For active constraint graphs
that are partial 3-trees, or Henneberg 1 graphs that are a
generalization of partial 3-trees, realization is straightforward.
For others that are merely 3-realizable or the rare cases that
are none of the above, we use a decomposition algorithm
[7] and an algorithm to optimize algebraic complexity of the
recombination systems to be solved [10], followed by a sub-
division algorithm [3] for solving the algebraic system.

**Theorem 3.8.** [10] Let $G$ be a well-constrained 3d distance
constraint graph that is decomposed into well-constrained
or minimally rigid subgraphs that are maximal in the sense
that no well-constrained graph contains them except possibly
$G$ itself. Let $P_G$ be the polynomial system for obtaining
the cartesian realizations of $G$ from the realizations of these
subgraphs. There is an algorithm that runs in time linear in
$|G|$ that optimizes the algebraic complexity of $P_G$ within a
class of natural parametrizations.

4 EASAL Algorithm and Formal Guarantees

On input consisting of the packing or assembly sys-
tem, our deterministic algorithm outputs a visual, query-
searchable stratification of the cartesian configuration space, and
if required, samples it at a desired level of refinement; this is done by efficiently computing an Atlas of the config-
uration space consisting of parametrized Charts.

4.1 Algorithm

The algorithm captures, stores and labels the regions
(represented by Charts) of the stratification (represented by
the Atlas) of the configuration space. The regions of At-
alas are stored as nodes of a directed acyclic graph, with the
edges of the graph representing immediate containment or
reachability. Each region of the Atlas is represented as an ac-
tive contact graph. By using the combinatorial structure of the (small) active constraint graph graph as a lookup label,
newly computed regions can be tested to ensure that they are
not already present in the current stratification. Only if the
contact graph is new, is the region further explored. Explo-
ration is, by default, depth first and new active constraints
and regions are added one by one.

**Note.** The simplified version we explain here explores the as-
sembly configuration space for 2 molecular units, by travers-
ing depth first. Extended features are listed in Section 5.

4.1.1 Pseudocode

The method to generate the stratification and the Atlas is
GenerateExploreAtlas. It calls the main method GenerateEx-
plorSubatlas which generates the stratification of a parent
region formed by some set of active constraints activeCon-
straintGraph.

This method calls itself recursively on new child re-
gions of the stratification that are discovered via SearchEx-
ploration (more interesting for approximate volume and entropy
computations) and SampleExplore (when uniform, step-wise
sampling of parametrized regions has to be performed).

4.2 Theorems guaranteeing correctness and
complexity

Here we prove the correctness and worst-case complex-
ity of the above algorithm i.e, the worst-case is when Sam-
pleExplore is used, i.e, when the algorithm steps through the
regions of the Atlas. SearchExplore works significantly
faster, but formal guarantees of correctness are outside this
paper’s scope.

The complexity of the methods whose computation
time depends on the output (number of Atlas nodes) as well
as the input (number of molecular units, their size, number
of constraints, required level of refinement) are explained
below.

**Proposition 4.1.** (Correctness) There is no omission of con-
figurations.

**Proof.** By partitioning the configuration space according to
strata and higher refinement at the lower dimensions, the
sampling is exhaustive and complete Atlas generation is
guaranteed. Also binary search at increasing the level of
refinement guarantees the recognition of each new region
when nonactive constraints become active and no difficult-
to-access regions are missed.

**Proposition 4.2.** The algorithm does not generate any re-
gions that are not present in the Atlas, nor sample any can-
didate configurations in them that are later rejected as in-
feasible.

**Proof.** If a set of constraints cannot be simultaneously active
then no superset can even be active. Since our approach
builds only on feasible active constraint sets, it prevents the
generation of regions or (parametrized) configurations that
are infeasible i.e., not present in the Atlas.

**Proposition 4.3.** Within a Chart of the Atlas, the algorithm
generates the minimum number of configurations that are
discarded as infeasible, i.e., that are not present in the cor-
responding region. This is because the Charts are a formally
optimal cover of the active constraint regions by Proposition
3.6. Let $ρ$ be a fixed ratio of feasible sample points to all
sample points.

**Theorem 4.4.** 1. Let $k$ and $N$ denote the number of
molecular units and the atomic units per molecule re-
spectively. There are $O(N^{k-1})$ bitethers which is the
maximum number of nonempty initial active constraint
regions that could be in the Atlas.

2. Each edge of Atlas requires $O(N^2)$ time for traversal.

3. Let $ρ$ be as in proposition 4.3. The average time it takes
per feasible sample point is $O(1/ρ)$. 
Proof. 1. We have to check all regions satisfying only the initial constraints, so if the \( k \) molecules with \( N \) atoms each are just constrained by a tree of bitethers, we have to check all possibilities of bitethers, which is \( N^{k-1} \), times the number of possible nonisomorphic trees of size \( k \), and the latter is constant if we assume \( k \) has some fixed upper bound.

2. Let \( s, t \) denote stepsize, tolerance respectively. The binary search part of SampleExplore and SearchExplore takes 
\[O(\log(s/t)N^2)\]
time. At each iteration of the loop, step size is halved. We quit from the loop when step size is less than the tolerance. Hence the loop will be repeated \( \log(s/t) \) times. Each iteration takes \( O(N^2) \) time to Realize and NonactiveConstraintCheck. The method Realize has constant time complexity, since 3-trees can be realized in time exponential in the size of the active constraint graph which is constant in the case of any fixed bound on \( k \) the number of molecular units. For non-partial 3-trees, use Proposition 3.8.

3. The bounds of convex Charts of active constraint regions, are computed in time quadratic in the size of the active constraint graphs by Theorem 3.6, which in the case of 2 helices, is constant time. Tightness of the convex Chart is proportional to \( \rho \), which gets its optimum value by 4.3 Also there is no double sampling and repetition of configurations, since every potential new activeConstraintGraph is checked for presence in current Atlas. Constant time for realization was discussed in 2.

\( \square \)

5 Conclusion

There are straightforward extensions to the algorithm section: (a) permit an already partially generated Atlas to be input; in this case algorithm proceeds from one of the unfinished regions of the current stratification; (b) start from a specified bi-tether or a specified region of the current stratification; (c) change the traversal of the stratification from depth to breadth first, for any specified region of the current stratification; (d) choose to only traverse specified regions of the current stratification; (d) allows increased sampling refinement for specified regions (e) limits stratification to regions satisfying global assembly constraints (f) extends stratification to include regions defined by active global assembly constraints.

Furthermore, there is a clear strategy for a more challenging extension of the algorithm to a small constant number of molecular units more than 2. This has been shown to be sufficient for dealing with arbitrarily large assemblies, using a multi-scale approach that employs decomposition into subassemblies and analyzing assembly pathways [8, 6, 1]. The Atlas facilitates computation of entropy. An efficient algorithm for computing the entropy, given the Atlas, would be very valuable.

References


