```
begin # dev-hack
    import Pkg
    Pkg.activate(".")
end
```

```
using Graphs, JLD2, MetaGraphs, MLMolGraph, PorousMaterials, StatsBase
```

Load Graph Data

Locations of cached Voro-graphs and Crystals with bonding graphs, and the number of graphs to load:

```
begin
    voro_dir = "data/cache/vspn"
    bond_dir = "data/cache/bonded_xtals"
    nb_graphs = 5000
end;
```

Array of 5000 structures for which to load graphs:

```
xtal_names = String.(reduce(vcat,
    split.(sample(readdir(voro_dir), nb_graphs, replace=false), ".jld2",
        keepempty=false)
));
```

Voro-Graphs

Function to load the Voro-graph for a specific Crystal:

```
function load_voro_graph(xtal_name::String; dir::String="")
    @load joinpath(dir, "$xtal_name.jld2") obj
    voro_graph, _ = obj
    return voro_graph
end;
```

Array of Voro-graphs for the selected structures:

```
voro_graphs = load_voro_graph.(xtal_names, dir=voro_dir);
```

Crystals

Function to load the Crystal corresponding to a specific Voro-graph:

```julia
function load_xtal(xtal_name::String; dir::String="")
    @load joinpath(dir, "$xtal_name.cif") obj
    xtal = obj[1]
    return xtal
end;
```

Array of Crystals for the selected structures:

```julia
crystals = load_xtal.(xtal_names, dir=bond_dir);
```

Calculate Properties

Node Degree

```julia
degree_population = reduce(vcat, degree.(voro_graphs));
```

Node Count

```julia
node_counts = nv.(voro_graphs);
```

Connected Components

```julia
conn_comps = length.(connected_components.(voro_graphs));
```

Graph Diameter

```julia
diameters = filter(d -> d ≠ Inf, [diameter(g) for g in voro_graphs if nv(g) > 0]);
```

Accessible Volume Coverage

The probe molecule and forcefield to use for calculating pore volume:

```julia
begin
    probe = Molecule("He")
    ljforcefield = LJForceField("UFF")
end;
```

A function to calculate the pore volume and accessibility grid of a crystal:

```julia
function pore_grid(crystal::Crystal)
    xtal = deepcopy(crystal)
    remove_bonds!(xtal)
    grid, _, _ = compute_accessibility_grid(xtal, probe, ljforcefield)
    return grid
```

```
end;
```

Arrays of pore volumes and accessibility grids for the selected structures:

```
grids = pore_grid.(crystals);
```

A function to determine if point (i, j, k) of the grid is within the sphere of radius r centered at x:

```
function covered(xtal, accessibility_grid, i, j, k, x, r)
    p = id_to_xf(accessibility_grid.n_pts, (i, j, k))
    d_xp = MLMolGraph.pbc_distance(x, p, xtal.box)
    return d_xp < r
end;
```

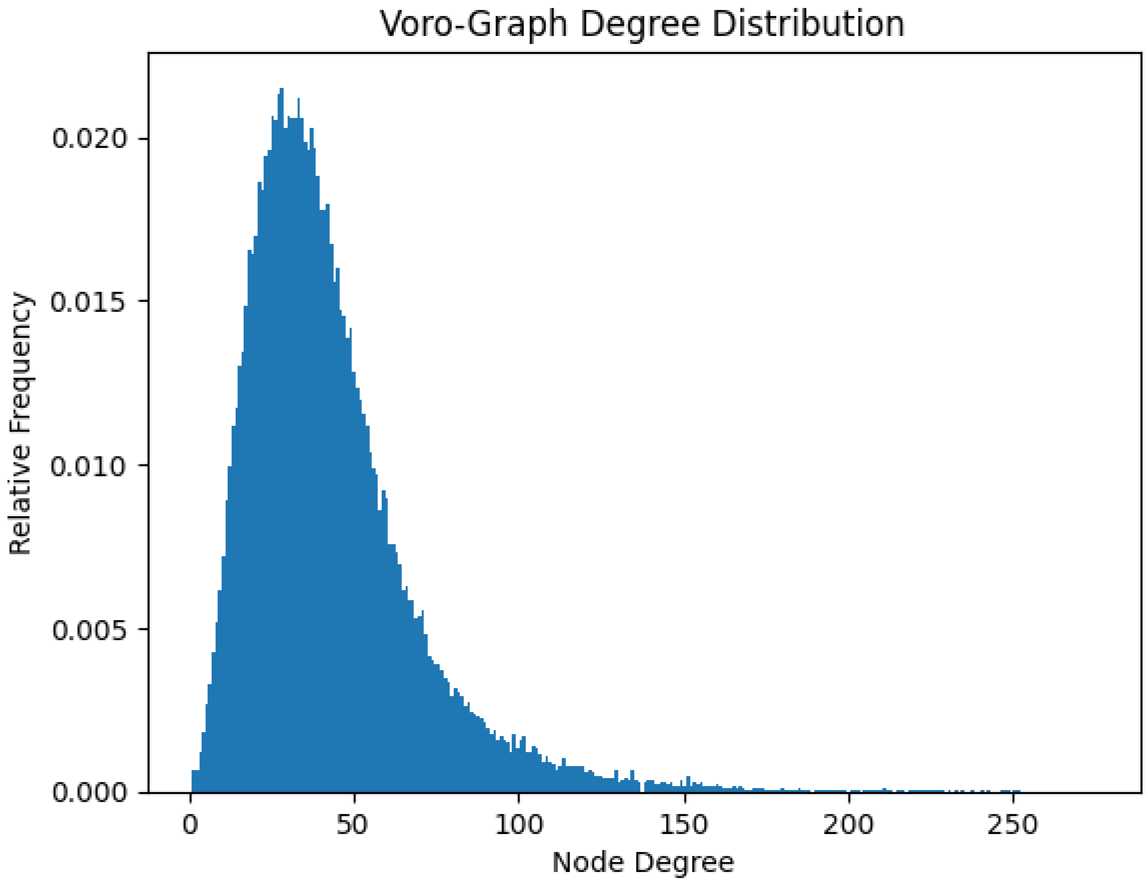
A function to calculate the proportion of a Crystal's accessible pore volume covered by the union of Voro-graph spheres:

```
function sphere_coverage(accessibility_grid, xtal, voro_graph)
    voro_grid = falses(size(accessibility_grid.data))
    n_pts_i, n_pts_j, n_pts_k = accessibility_grid.n_pts
    for v in vertices(voro_graph)
        x = MLMolGraph.shift_back(get_prop(voro_graph, v, :point).coords, xtal.box)
        r = get_prop(voro_graph, v, :radius)
        for k in 1:n_pts_k
            for j in 1:n_pts_j
                for i in 1:n_pts_i
                    voro_grid[i, j, k] =
                        voro_grid[i, j, k] ||
                        covered(xtal, accessibility_grid, i, j, k, x, r)
                end
            end
        end
    end
    bitvec1 = reshape(accessibility_grid.data .== true, :)
    bitvec2 = reshape(voro_grid, :)
    bitvec3 = bitvec1 .& bitvec2
    return sum(bitvec3) / sum(bitvec1)
end;
```

Array of sphere coverages as proportion of each structure's accessible pore volume:

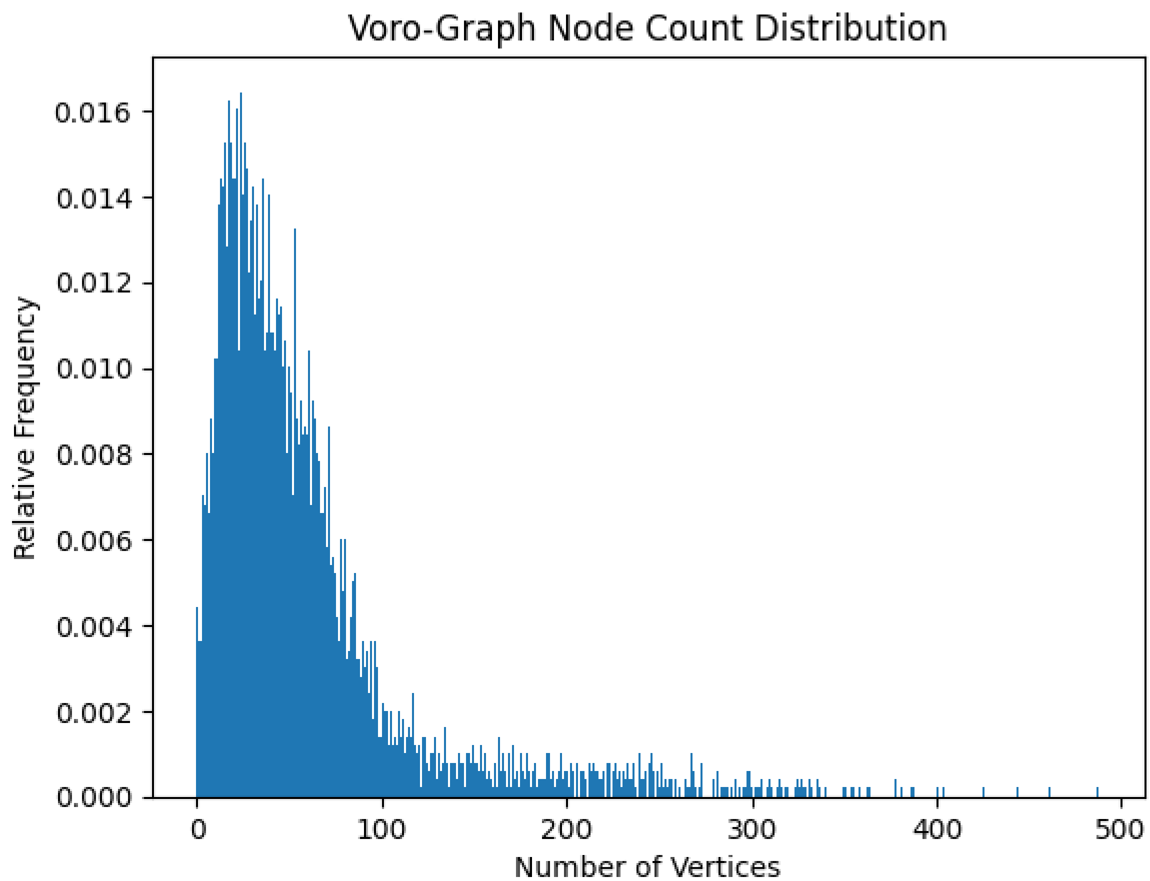
```
sphere_coverages = sphere_coverage.(grids, crystals, voro_graphs);
```

Plot Results



Graphs w/ nodes of degree zero: 0

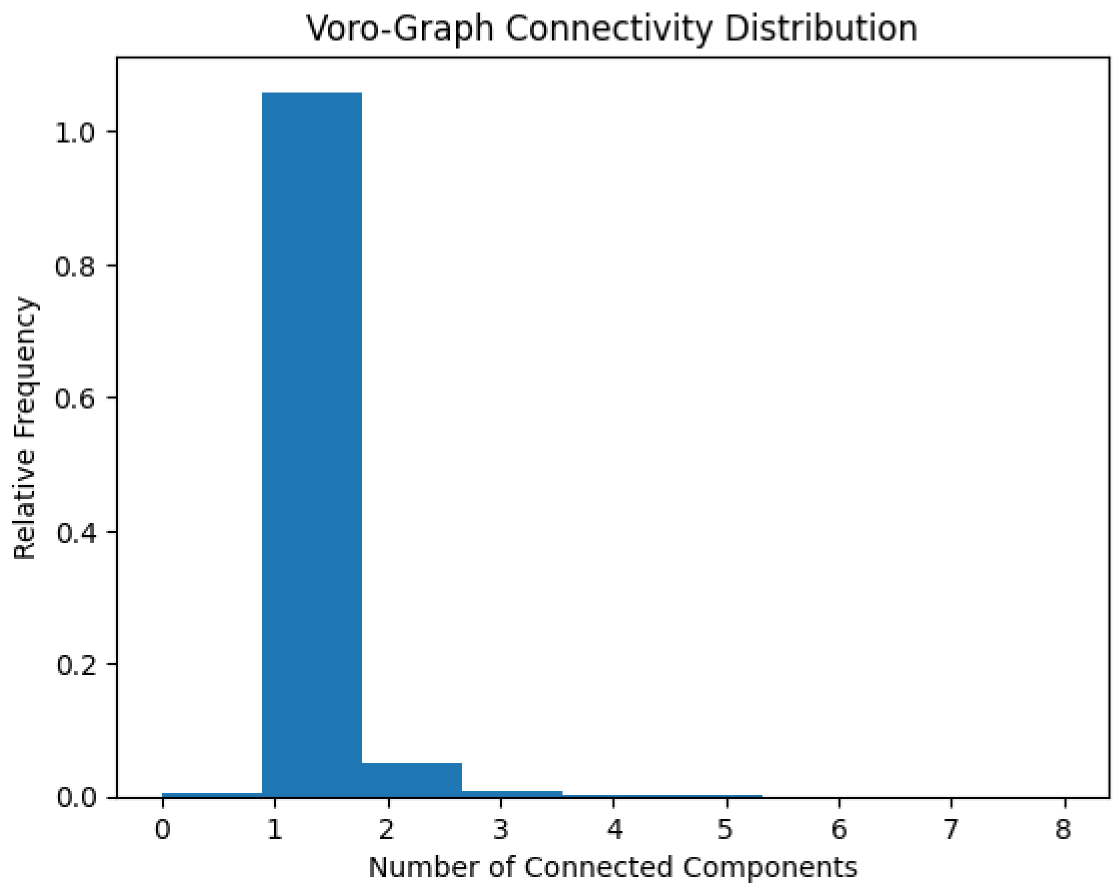
> **Note**
>
> No Voro-graphs end up w/ nodes having degree zero. Some graphs have extremely highly-connected nodes (probably not an issue...)



0.44 % of Voro-graphs have zero nodes.

Note

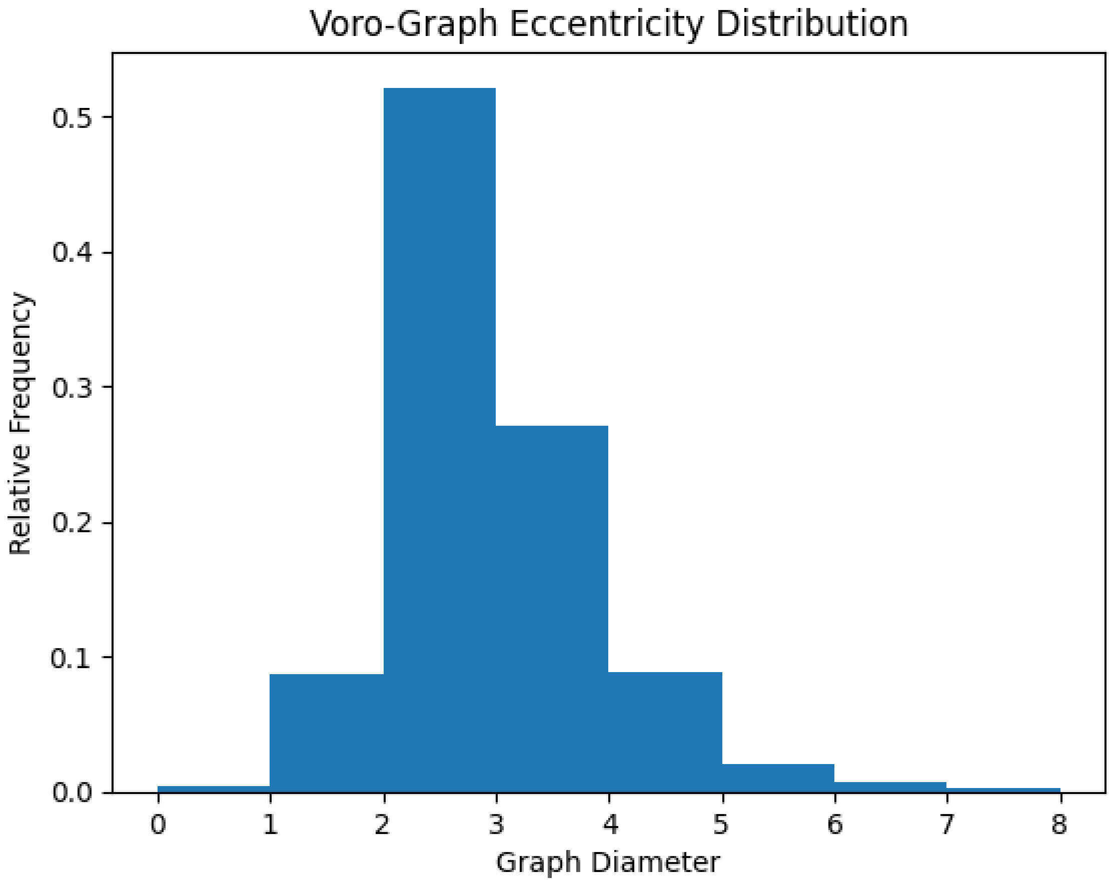
Some (but not many) Voro-graphs contain zero nodes. Why? Are these graphs w/o He-accessible pore space? Also, some graphs contain very large numbers of nodes (hopefully not a problem...)



94.0 % of graphs are a single connected component.

Note

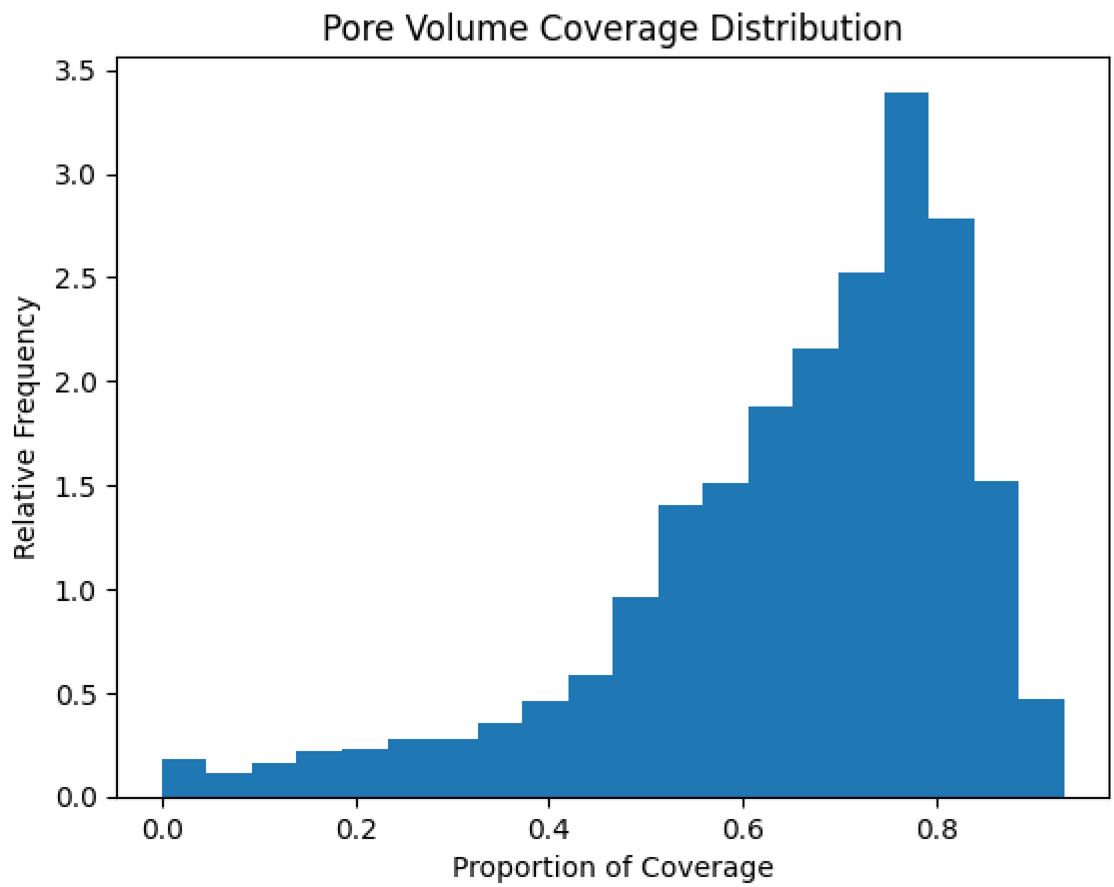
Very few Voro-graphs contain disjoint subgraphs. Are these structures with highly distinct 1D channels?



Largest graph diameter = 8

3.0 % of Voro-graphs have diameter > 4

> **Note**
>
> The diameters of the Voro-graphs are rarely larger than 4. Will 5 message-passing steps be good?



41.0 % of Voro-graphs provide less than 66 % coverage of the accessible pore volume.

Note

We should probably have a 3rd greedy selection stage that adds additional vertices until some threshold is reached.