

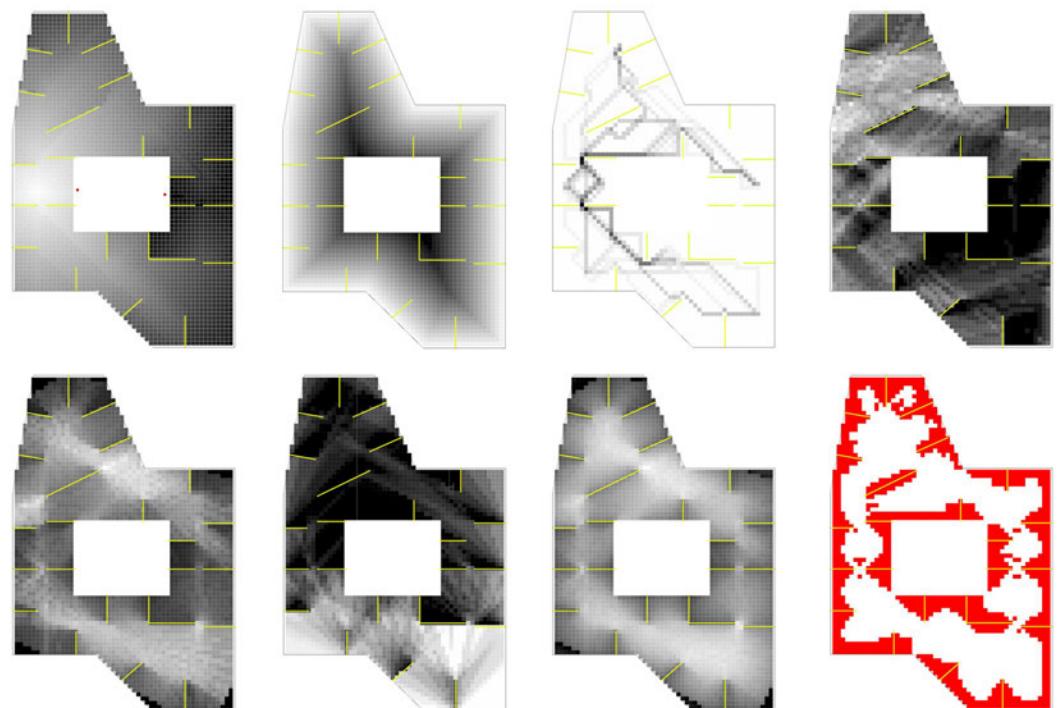
Intro to
PlanBee for Grasshopper

Quick Disclaimer

PlanBee is a collection of utilities that borrow concepts from theories such as Space Syntax. I have been exposed to some of these concepts but am not a Space Syntax expert! Welcome to any and all feedback on how to improve PlanBee.

What is PlanBee?

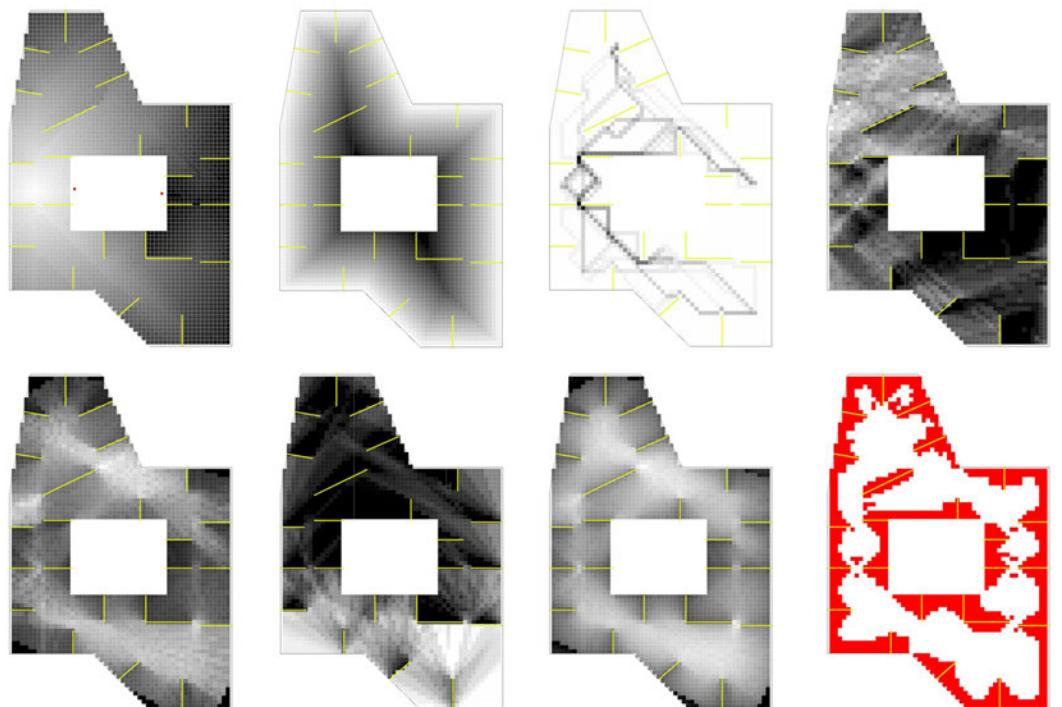
- Floor plan analysis Grasshopper plugin
 - Grid-based
 - Cell-by-cell score for each metric
 - Task-capable analysis components
 - Static analyses
 - 'Heatmap' outputs
 - Heatmap viz built into components, no need to create custom viz if not required.
 - Intended to reveal inherent qualities in a configuration
 - Borrows heavily from Space Syntax concepts
 - WIP
 - Food4Rhino link:
<https://www.food4rhino.com/app/planbee>
 - Github Repository: <https://github.com/M-JULIANI/planbeeGH>



Source: Author's work

What to use PlanBee for?

- Floor plan analysis visual outputs
 - Visual connectivity
 - Max exit distances
 - Daylight access
 - Covid
 - Etc.
- Analysis data production for downstream data processing/ unsupervised ML/ other
- Sketch-to-floor plan production (WIP!)

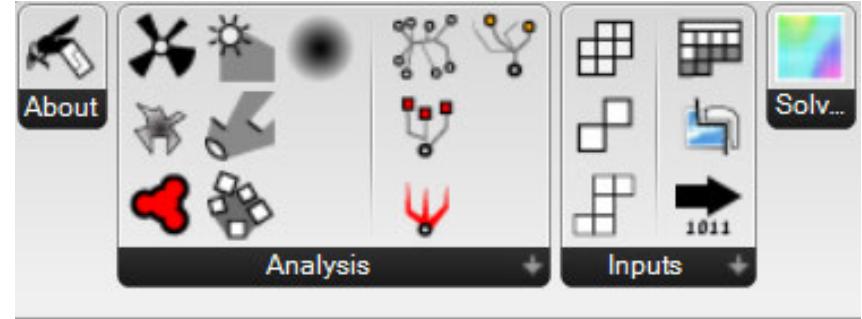


Source: Author's work

PlanBee v0.0.2

New features:

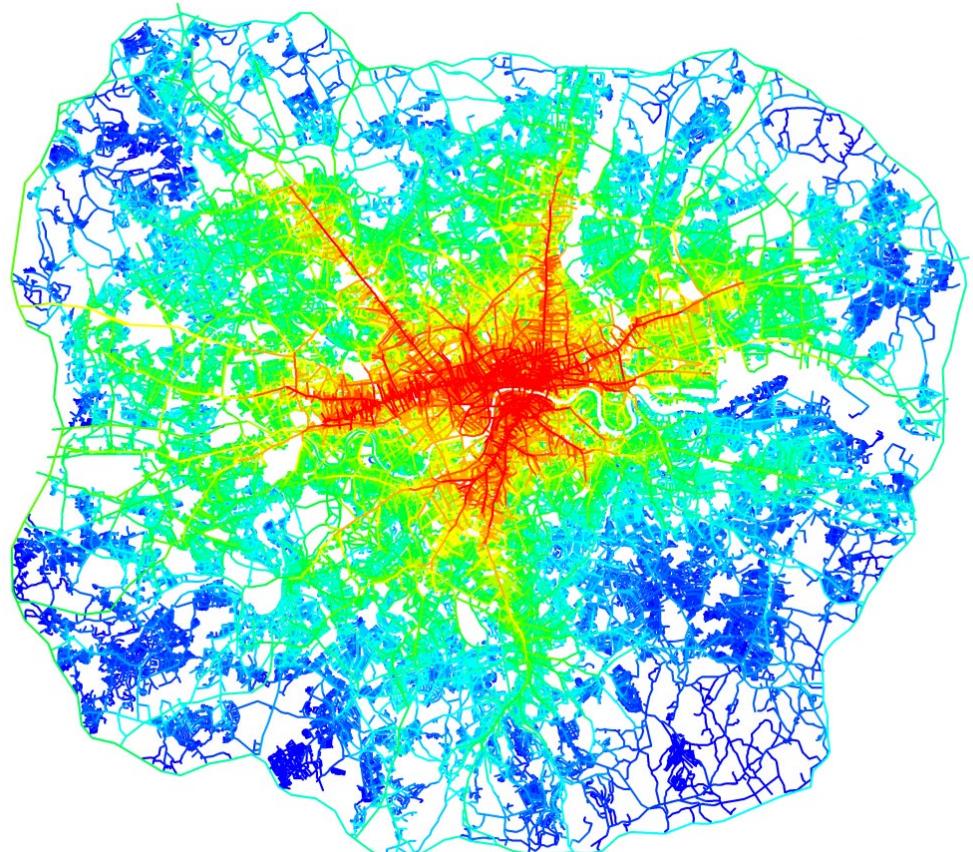
- Analysis grid:
 - Simplified grids, no need for core inputs
- Extended floor plan metrics:
 - Mean Shortest Path
 - Means (Sampled) Shortest Path
 - Neighborhood Size
 - Covid metric
 - 'Between path' metric
 - 'Exit frequency' metric
- Image-processing component:
 - For image-to-vector/plan



Source: Author's work

What is Space Syntax?

- A set of theories and techniques for the analysis of spatial configurations (Wikipedia).
- Pioneered at UCL Bartlett in the late 1970s.
- Remains an area of study at UCL Bartlett.
- A host of propriety and open source software has been developed for the purpose of carrying out Space Syntax-type analysis.
- There is also a design consultancy called Space Syntax Ltd.

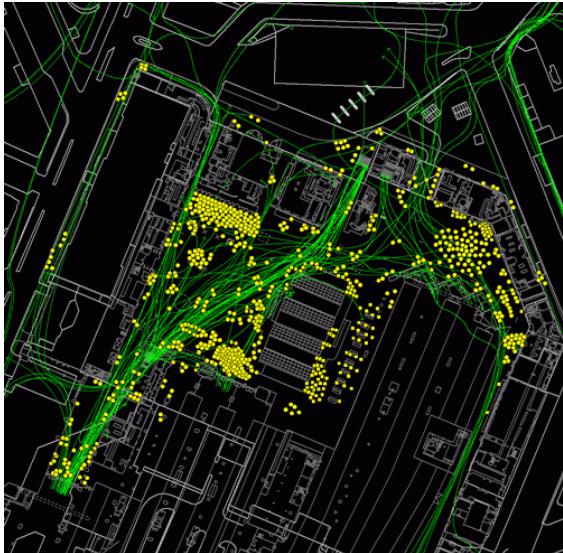


Integration Map of London
Credits: spacesyntax.net

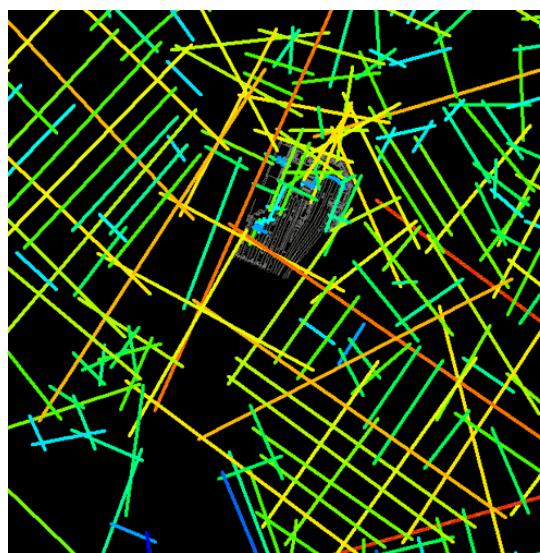
What is Space Syntax?

Scale:

Urban scale (street network)

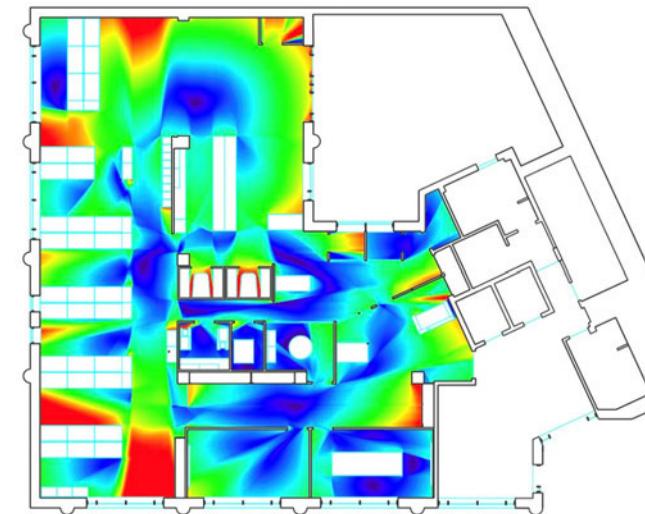


Passenger flows Victoria Station
Credits: spacesyntax.net



Axial map Victoria Station
Credits: spacesyntax.net

Building scale (interior space)

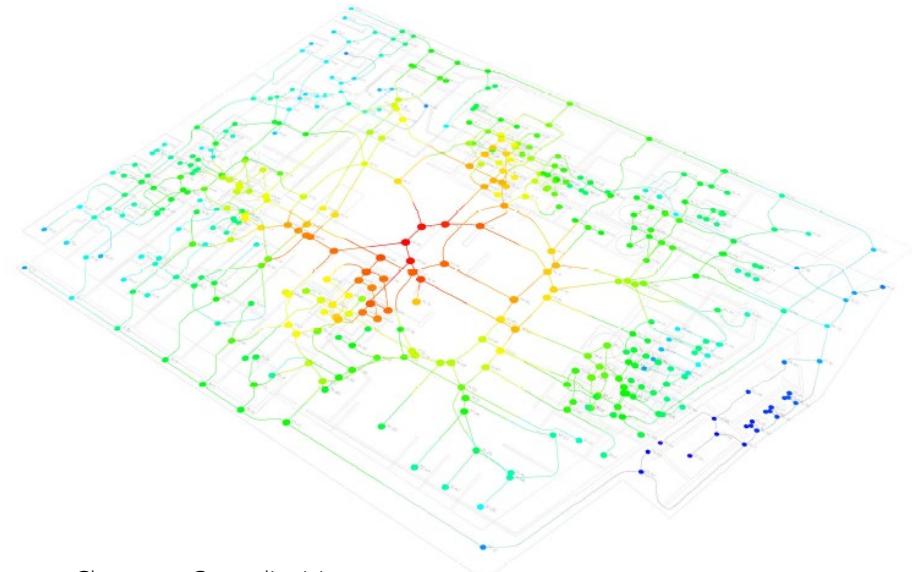


'Drift' Analysis
Credits: isovists.org

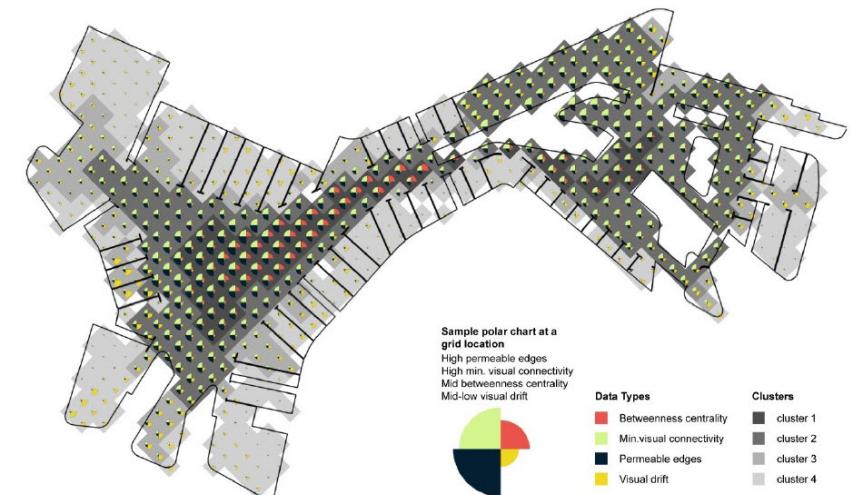
What is Space Syntax?

Different abstractions needed:

- Space as graph/ network
 - Enables us to study space through the lens of the mathematical field of Graph Theory
- A graph/network of discrete:
 - Lines
 - Convex Spaces
 - Pixels



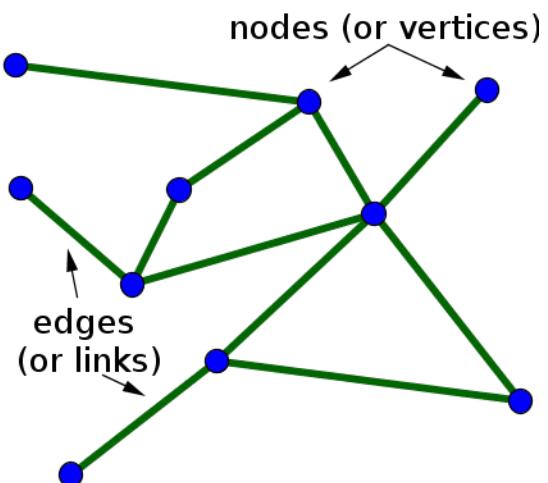
Closeness Centrality Map
Credits: Christian Derix, Woods Bagot



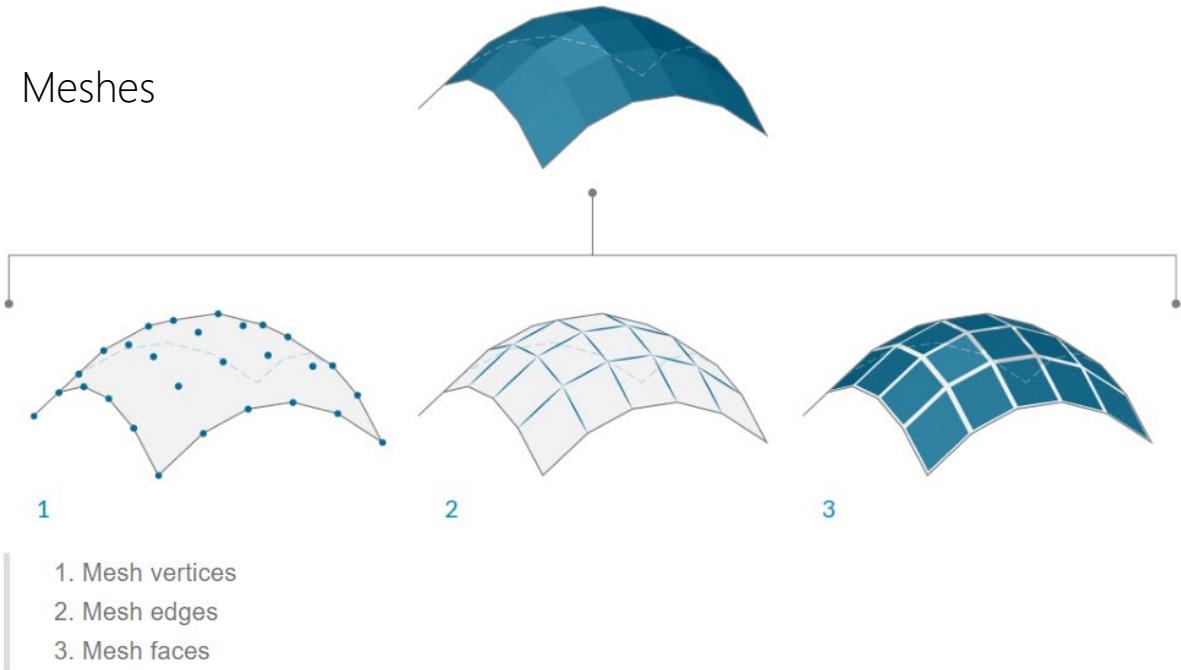
SOM Feature Map
Credits: Christian Derix, Woods Bagot

'Graph' you say?

Abstract



Meshes



Credits: Math Insight

Credits: ModeLab GH Primer

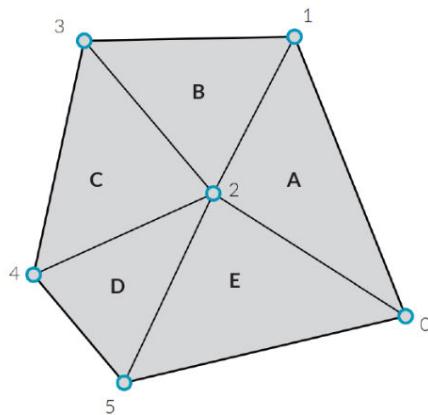
Meshes as graphs

- Meshes are graphs:
 - Collection of nodes & edges
 - Relationship between the two
- *Topology* = Connectivity information

1
2

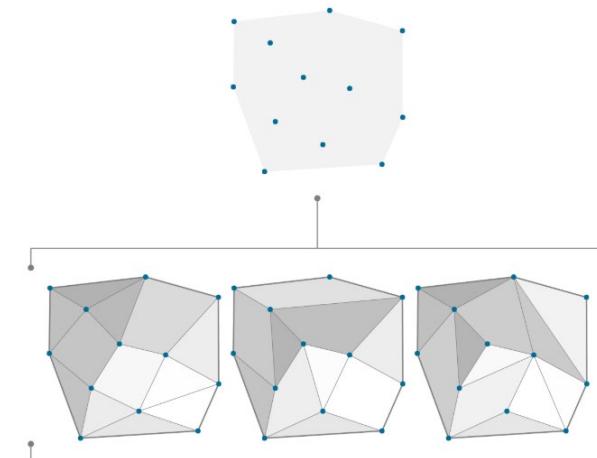
Vertex List
0 = [3.0, -2.0, 0.0]
1 = [1.0, 2.0, 0.0]
2 = [0.0, 0.0, 0.0]
3 = [-2.0, 2.0, 0.0]
4 = [-2.5, -1.0, 0.0]
5 = [-1.0, -2.5, 0.0]

Face List
A = {0;2;1}
B = {1;2;3}
C = {3;2;4}
D = {4;2;5}
E = {5;2;0}



1. A list of vertices.
2. Faces with groupings of vertices

Credits: ModeLab GH Primer



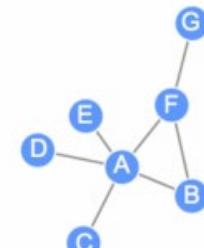
The same set of vertices can have different connectivity information, resulting in different topology.

Credits: ModeLab GH Primer

Graph Theory

- A network can be defined as a graph in which nodes and/or edges have attributes (Wikipedia).
- Any graph/network can be expressed mathematically in the form of an adjacency matrix.
- Using graphs opens the possibility of using the various tools already available in graph & network theory for the analysis of space.

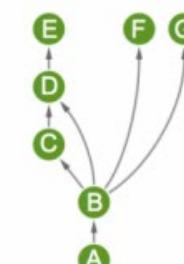
Undirected



| | A | B | C | D | E | F | G | Degree |
|---|---|---|---|---|---|---|---|--------|
| A | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 5 |
| B | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| C | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| D | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| E | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| F | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 3 |
| G | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

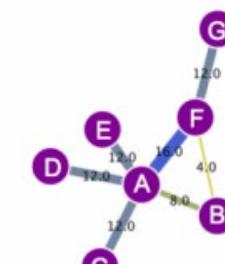
Adjacency matrices

Directed



| | A | B | C | D | E | F | G | Out-degree |
|---|---|---|---|---|---|---|---|------------|
| A | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| B | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 4 |
| C | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| D | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| E | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Weighted

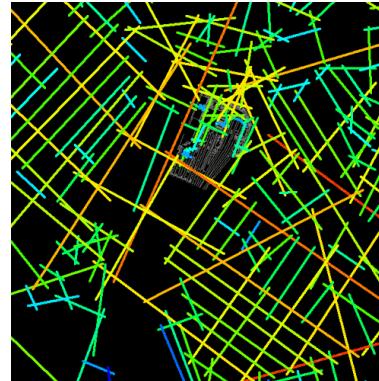


| | A | B | C | D | E | F | G | Degree |
|---|----|---|----|----|----|----|----|--------|
| A | 0 | 8 | 12 | 12 | 12 | 16 | 12 | 72 |
| B | 8 | 0 | 0 | 0 | 0 | 4 | 0 | 12 |
| C | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| D | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| E | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 12 |
| F | 16 | 4 | 0 | 0 | 0 | 0 | 12 | 32 |
| G | 12 | 0 | 0 | 0 | 0 | 12 | 0 | 24 |

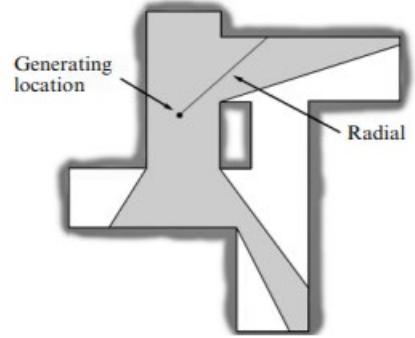
Credits: <https://www.ebi.ac.uk/training/online/course/network-analysis-protein-interaction-data-introduction/introduction-graph-theory/graph-0>

Conceptions of space/ (building blocks)

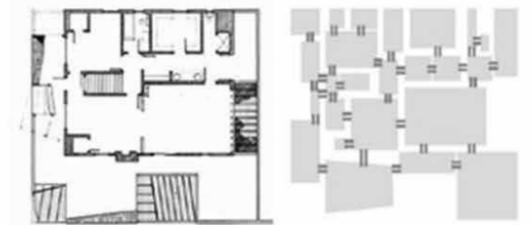
- **Isovist:** Or viewshed or visibility polygon, the field of view from any point
- **Axial space/map:** Axial space represents the maximum extension of a space as a point in one dimension.
- **Convex space/map:** Convex space is one in which no straight line drawn between any two points goes outside the space.



Credits:
<https://spacesyntax.com/project/victoria-station/>



Credits: Turner, A.
<https://core.ac.uk/download/pdf/1668886.pdf>

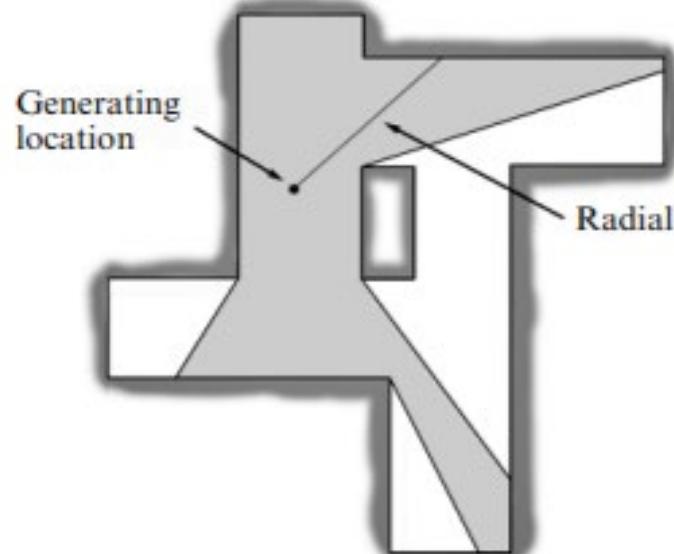


Credits: journals.open.tudelft.nl

Building Analysis

- Isovist

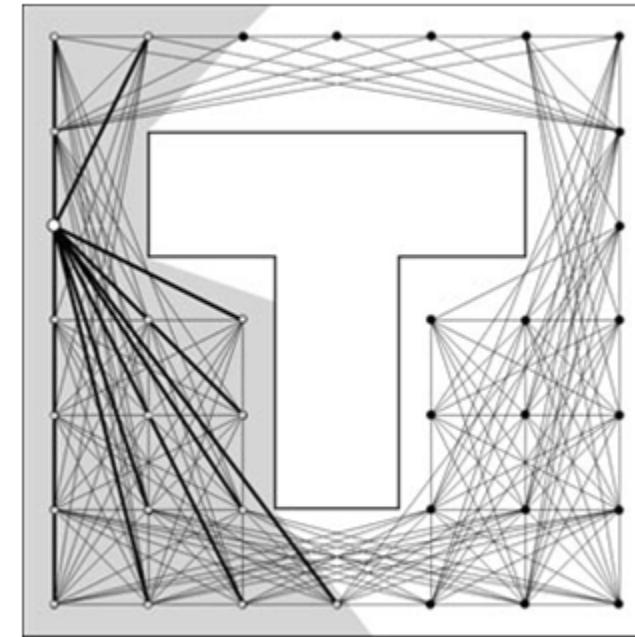
Or viewshed or visibility polygon, the field of view from any point



Credits: Turner, A. <https://core.ac.uk/download/pdf/1668886.pdf>

- Visibility Graph

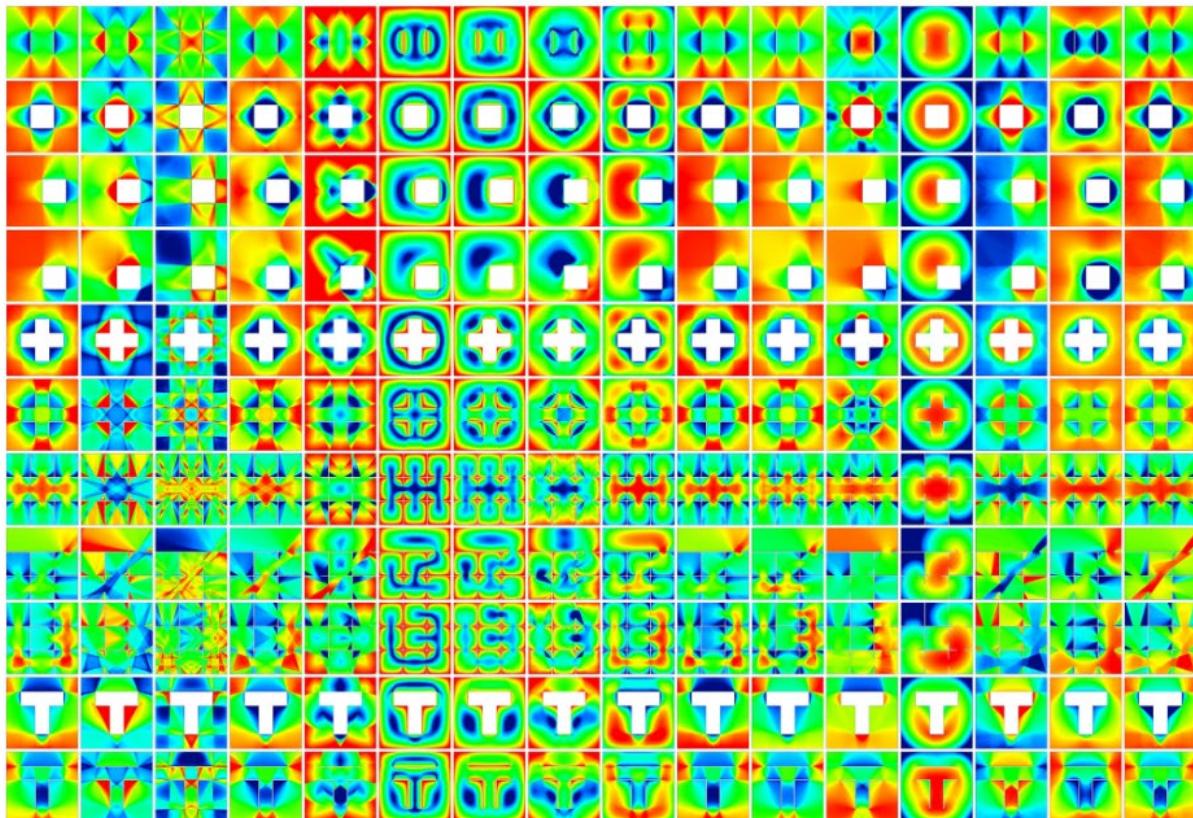
Visibility graph is the graph of mutually visible locations in a spatial layout.



Credits: spacesyntax.net

Space Syntax Software

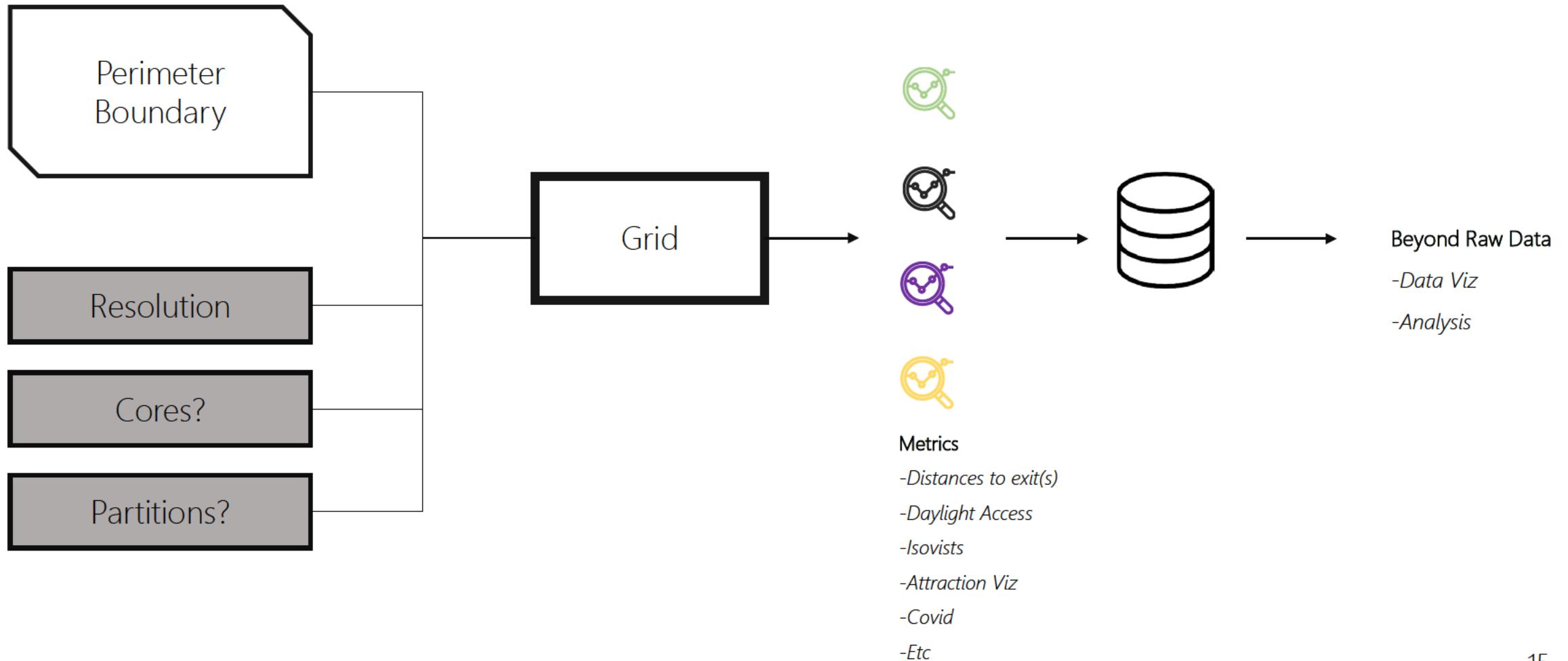
- DepthmapX:
<https://github.com/SpaceGroupUCL/depthmapX/releases>
- Isovists.org: <https://isovists.org/>
- Decoding Spaces:
<https://toolbox.decodingspaces.net/>



A grid of test geometries for all measures; left to right; visibility, compactness, occlusivity, perimeter, vista, drift, variance, skewness, average radial, connectivity, control, controllability, mean metric depth, mean visual depth, mean angular depth and integration (HH).

Credits: isovists.org

PlanBee sample workflow



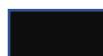
PlanBee metrics

Custom Metrics

- *Distance to perimeter metric*
- *Daylight metric*
- *Attractor viz metric*
- *Covid metric*
- *Exit access metric*
- *Exit path most travelled metric (WIP)*
- *'Between points' most travelled metric (experimental)*



Path-related



Other

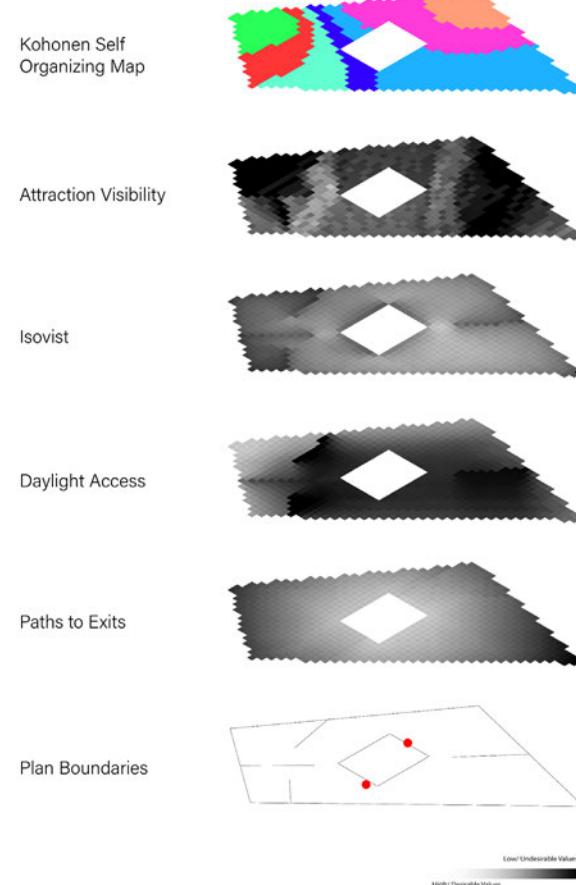
Space Syntax Derived Metrics

- *MSP (sampled) metric (WIP)*
- *Neighborhood Size/ Visual connectivity/VGA/ Isovist Area*
- *Clustering Coefficient metric (WIP)*

Multi-dimensional field

Taken together, when all metrics are calculated for each cell, each cell represents a multi-dimensional vector.

Taken as a whole a floor plan becomes a multi-dimensional field.

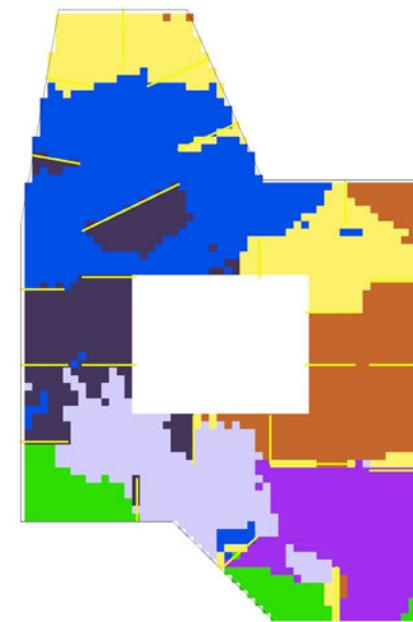


Stacked Metrics

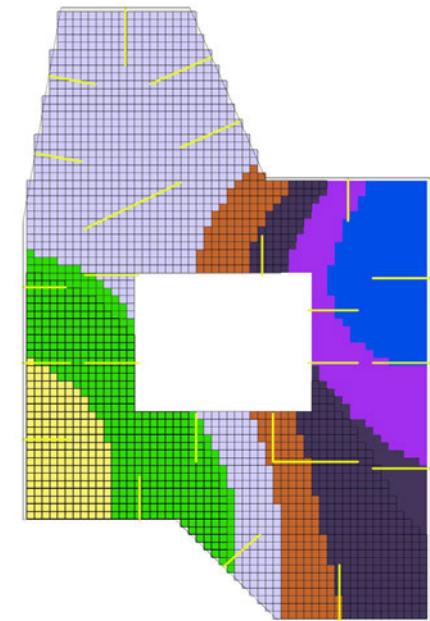
Beyond raw data

Unsupervised ML/Classification

- ProvingGround's LunchboxML
 - K-means clustering
 - Gaussian mixture model
- PlanBee Kohonen Self-Organizing Feature Map
- Other dimensionality reduction/clustering techniques...



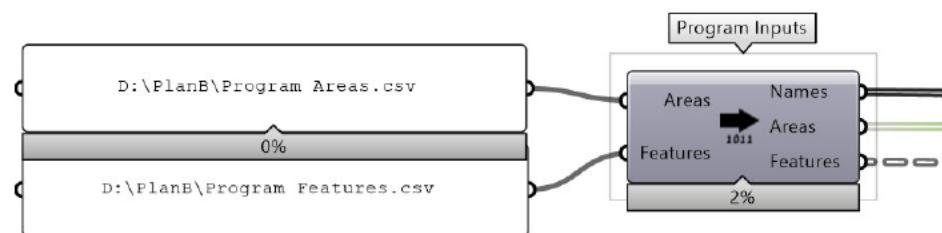
K-means clustering



Kohonen Self-Organizing Feature Map

PlanBee Components: Inputs

Program Inputs:



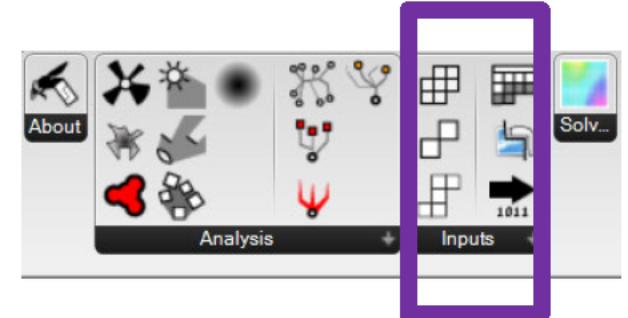
Program Inputs Component

| | A | B |
|---|-----------|------|
| 1 | Program | Area |
| 2 | Open | 800 |
| 3 | Storage | 200 |
| 4 | Kitchen | 1200 |
| 5 | Meeting | 1500 |
| 6 | Reception | 550 |
| 7 | Private | 700 |
| 8 | MultiUse | 1500 |
| 9 | | |

Program Areas

| | A | B | C | D | E |
|---|-------------|---------------|--------------------|-----------------------|---------------------|
| 1 | Program | Natural Light | Views to landmarks | Proximity to entrance | Visual connectivity |
| 2 | Open Office | 1 | 0.95 | 0.7 | 0.9 |
| 3 | Storage | 0.01 | 0.01 | 0.6 | 0.01 |
| 4 | Kitchen | 0.75 | 0.8 | 0.5 | 0.8 |
| 5 | Meeting | 0.4 | 0.4 | 0.2 | 0.3 |
| 6 | Reception | 0.4 | 0.25 | 1 | 0.85 |
| 7 | Private | 0.1 | 0.15 | 0.08 | 0.02 |
| 8 | MultiUse | 0.8 | 0.1 | 0.95 | 0.7 |
| 9 | | | | | |

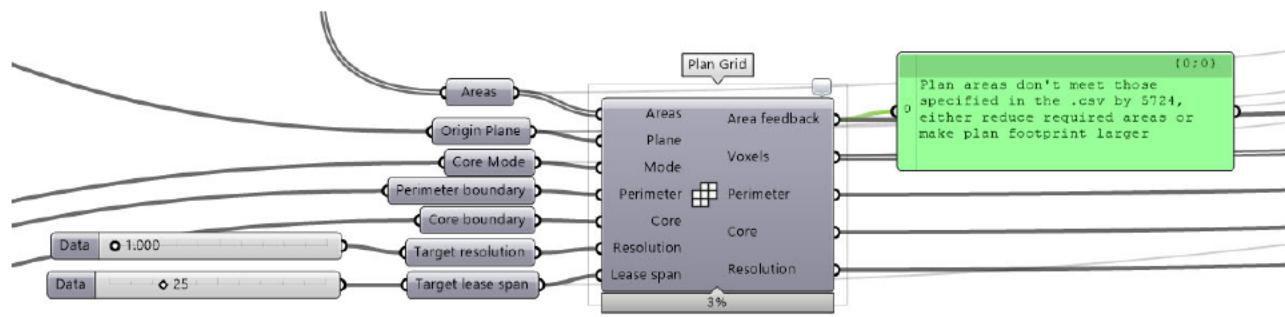
Program Features



Source: Author's work

PlanBee Components: Inputs

Analysis Grids



Max Inputs:

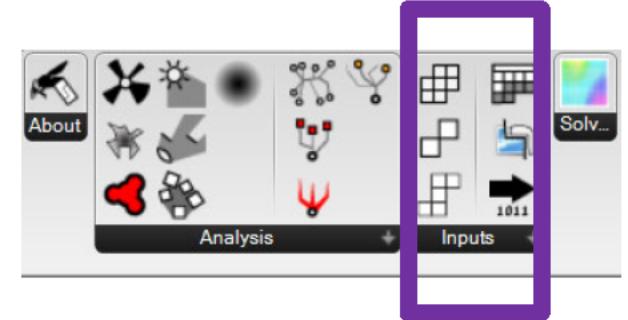
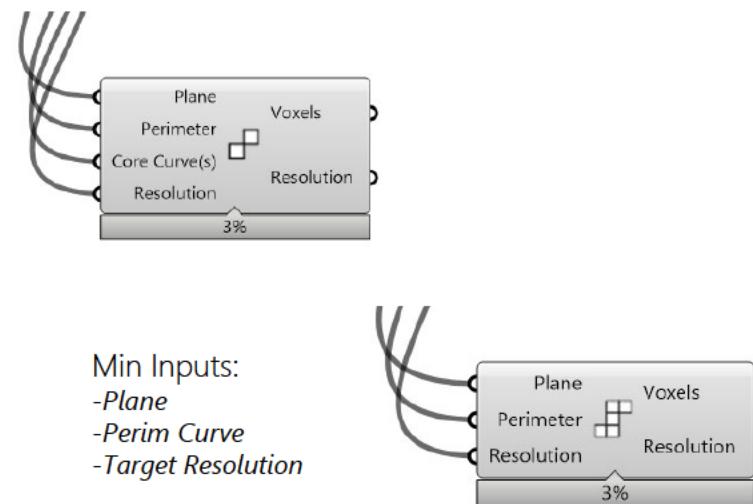
- Program Areas
- Plane
- Core Mode
- Core(s)
- Target Resolution
- Target lease span

Mid Inputs:

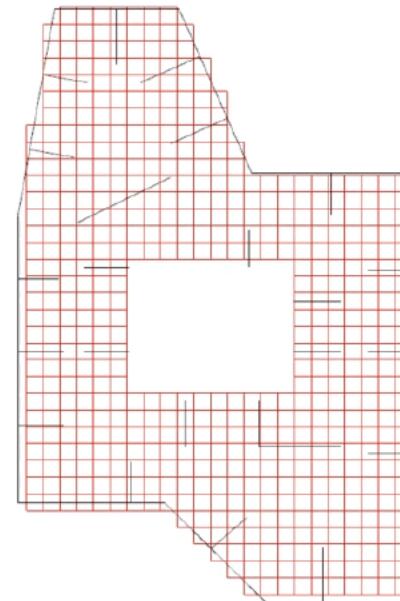
- Plane
- Perim Curve
- Core(s)
- Target Resolution

Min Inputs:

- Plane
- Perim Curve
- Target Resolution

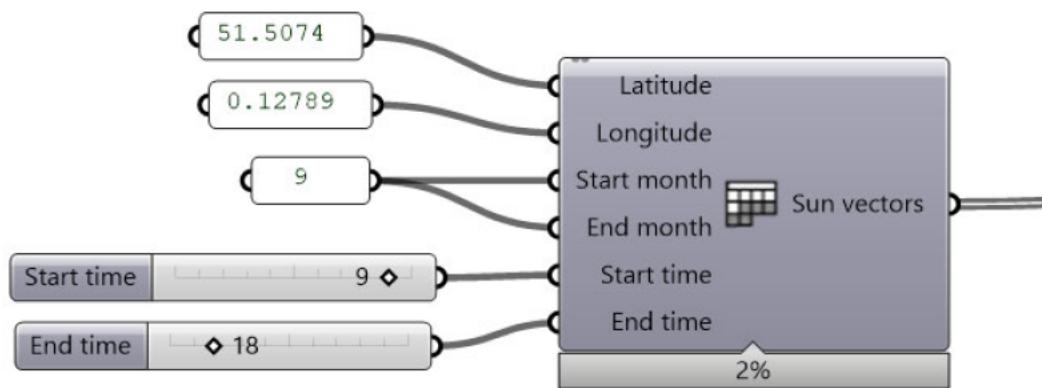


Source: Author's work

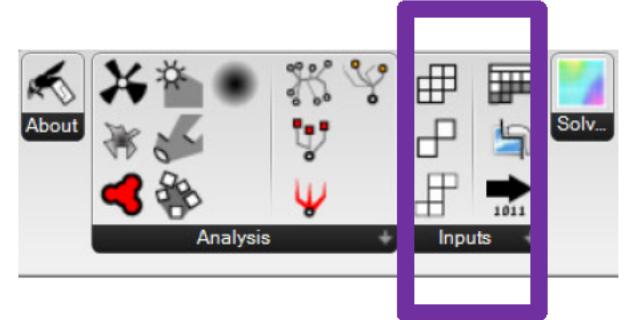


PlanBee Components: Inputs

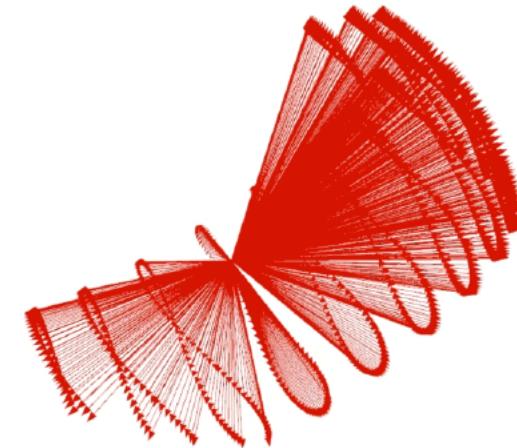
Daylight Period



*Creates a sampling of vector for an analysis period using the built-in Rhino sun

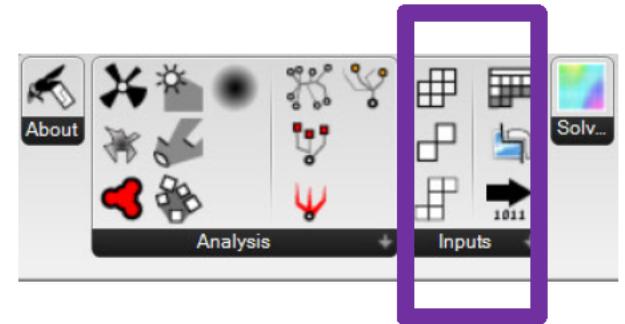


Source: Author's work

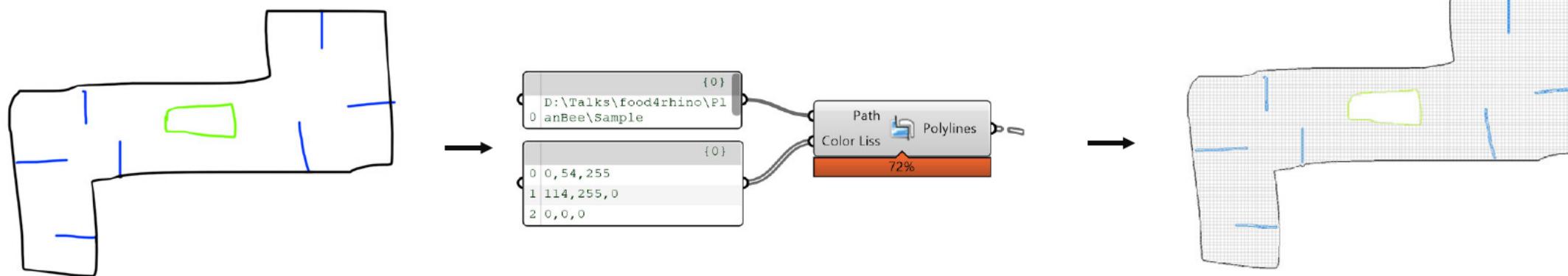


PlanBee Components: Inputs

Image-to-plan



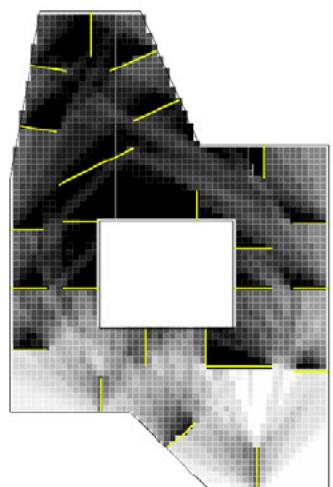
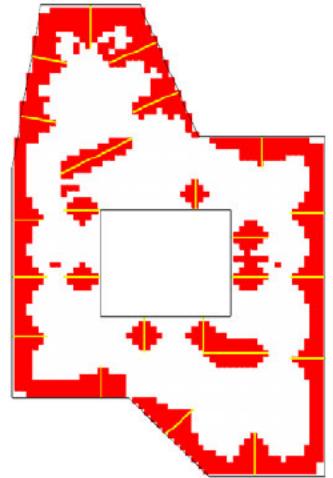
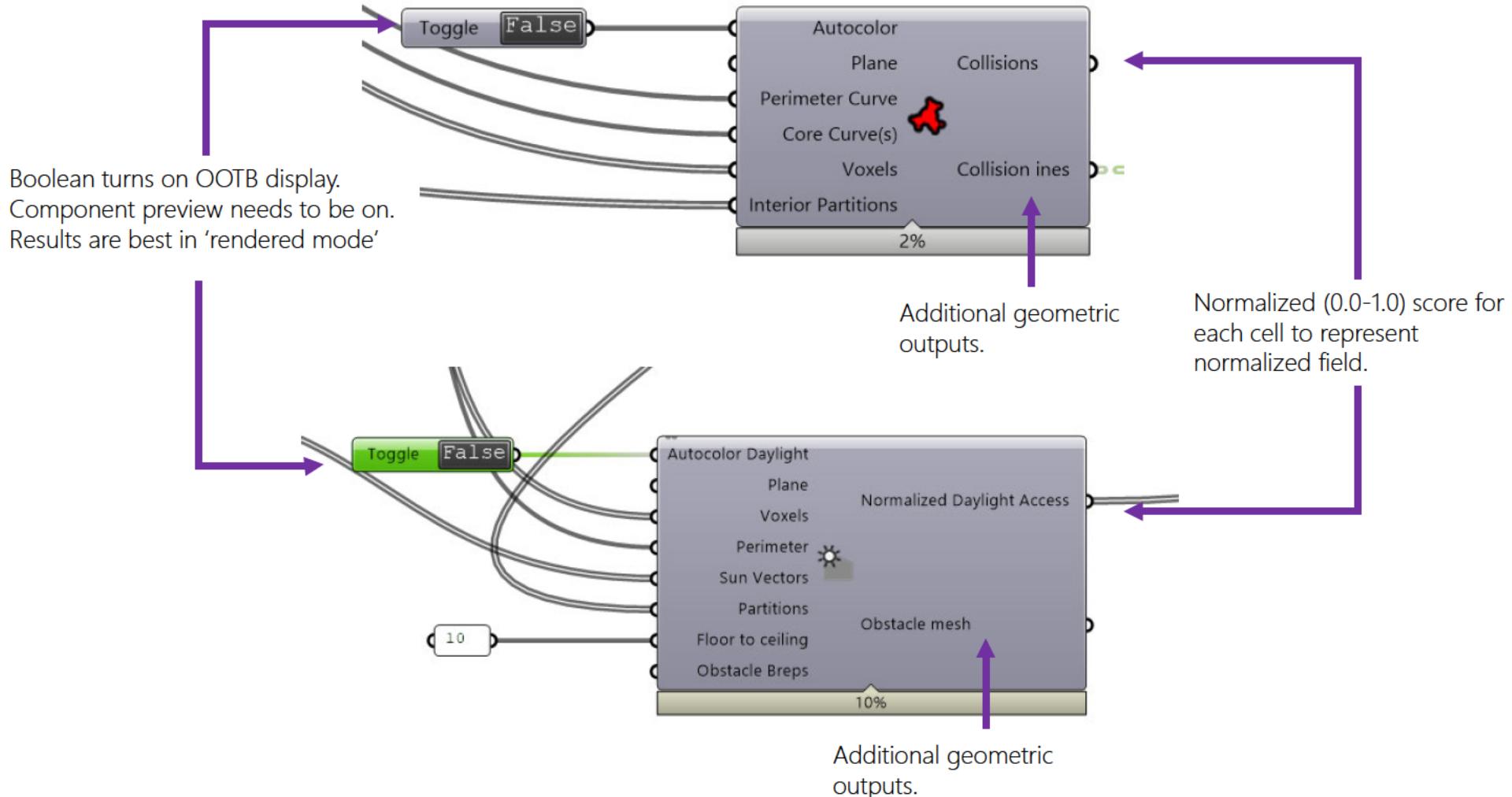
Source: Author's work



*Image processing component that converts pixels to usable polylines

Quick demo...

Anatomy of an Analysis Component

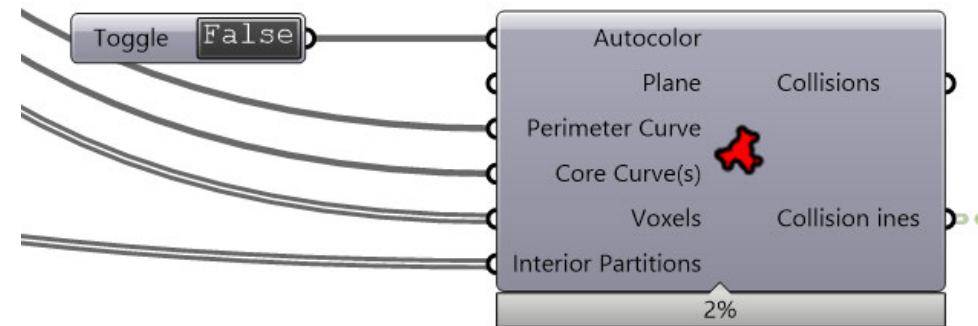


Anatomy of an Analysis Component

A word of caution:

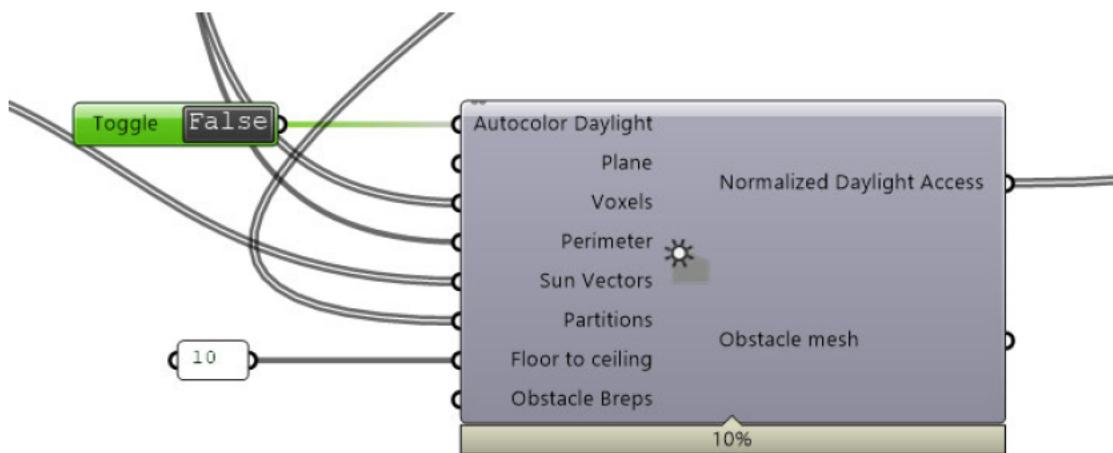
-Built-in display overrides are costly as they are currently built. Make sure you don't leave the 'autocolor' set to 'True' for any component you don't intent to view/capture and image- it will slow your GH definition way down.

-Pathfinding components are the most computationally intensive. If you have multiple analysis components in a definition it will slow it down. Disable components that are not being used, especially pathfinding ones!

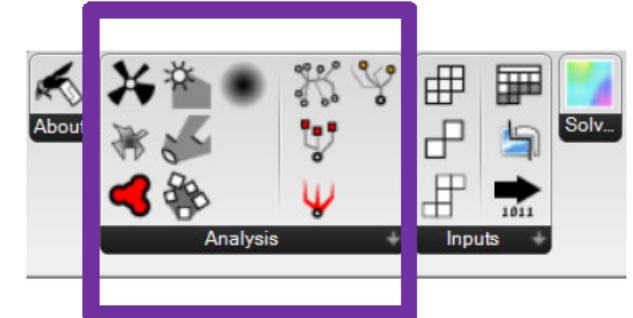


PlanBee Components: Analysis

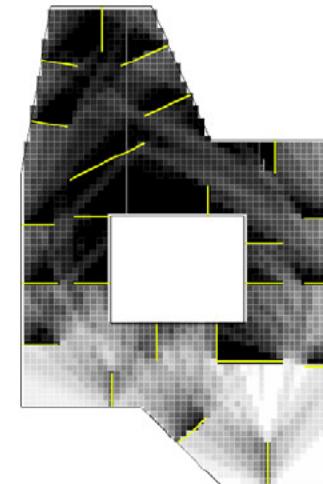
Daylight Metric



Using sampled sun rays, each cell is analyzed for the total number of collisions with obstacle objects (interior partitions, an automatically generated ceiling at a specified height, as well as optional additional obstacle objects).

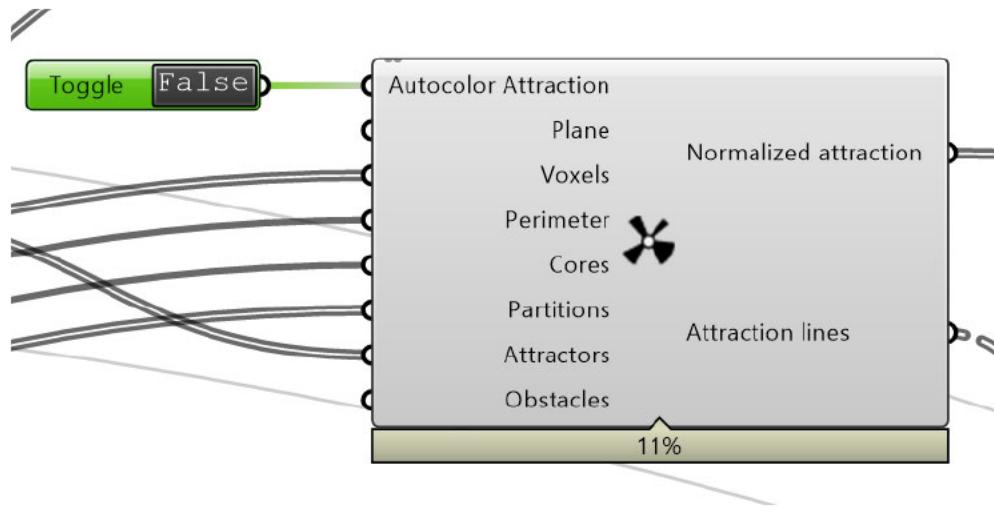


Source: Author's work

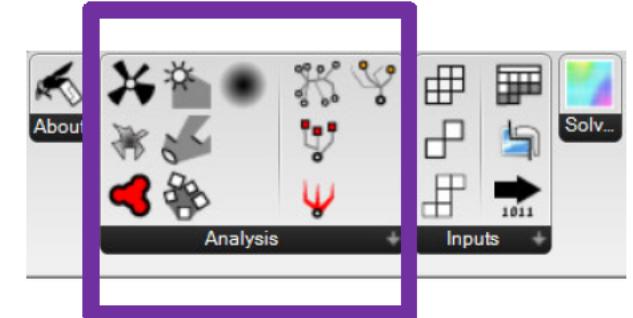


PlanBee Components: Analysis

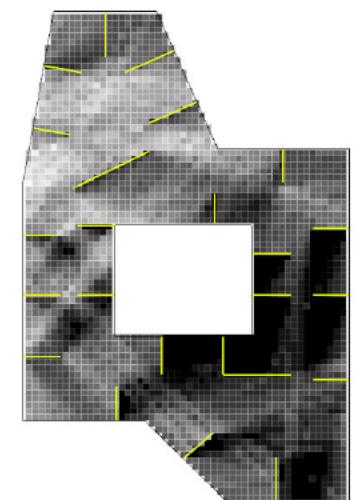
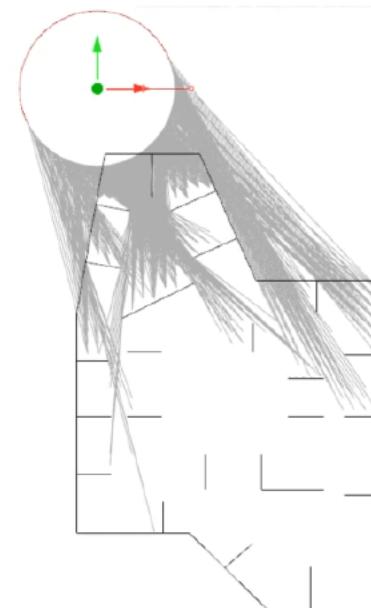
Attractor Visibility Metric



By raycasting from each cell to attractor objects we determine how much visibility to attractors each cell has. Interior partitions, cores, and additional obstacle geometry can be fed.

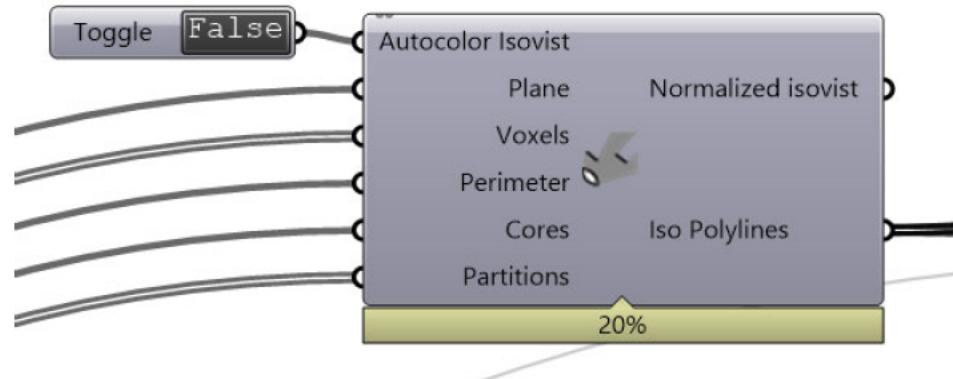


Source: Author's work

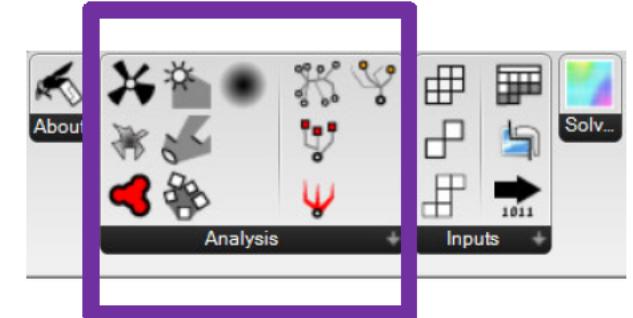


PlanBee Components: Analysis

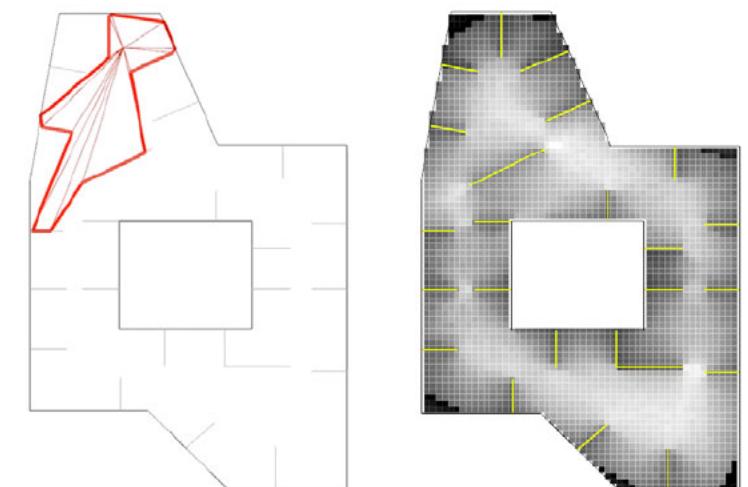
Isovist Metric



Standard (sampled) isovist. Similar to Grasshopper built-in isovist. The length of all obstacle-colliding-lines from a starting cell is added together.

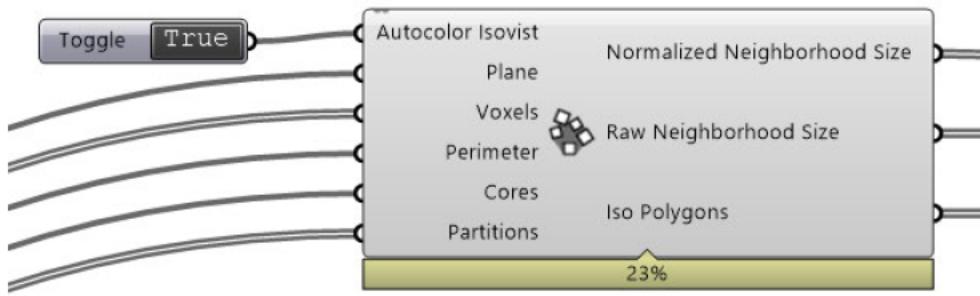


Source: Author's work



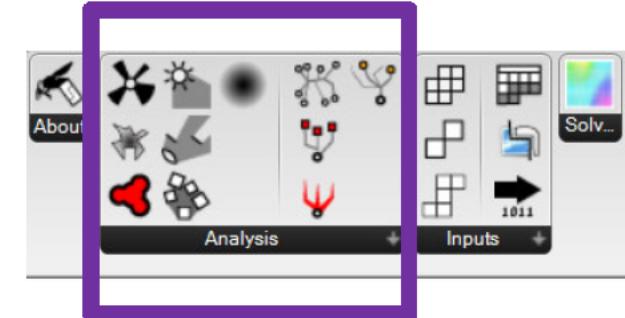
PlanBee Components: Analysis

Neighborhood Size/ Visual
Connectivity/VGA/Isovist Area/ Metric

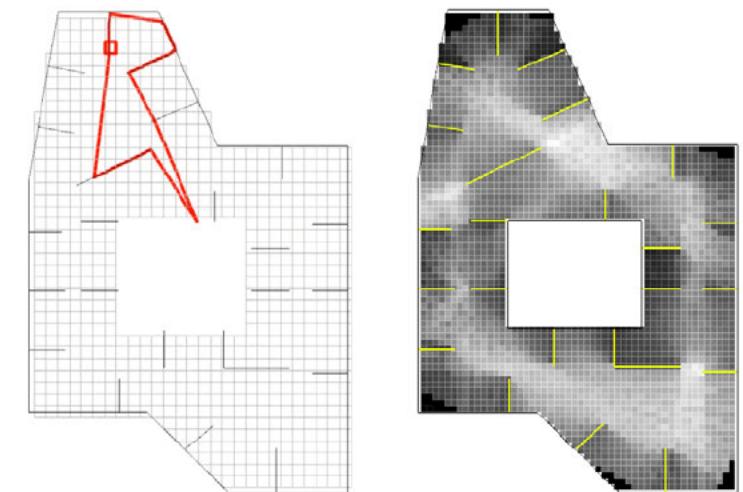


Like isovist, however the isovist polygon area is reported rather than the length of collision lines. This metric is very similar to Space Syntax's *Visual Graph Analysis* (VGA):*Neighborhood size* and *VGA:Connectivity* metrics.

*This metric is indicative of the relative connectivity of different points on the floor plan.

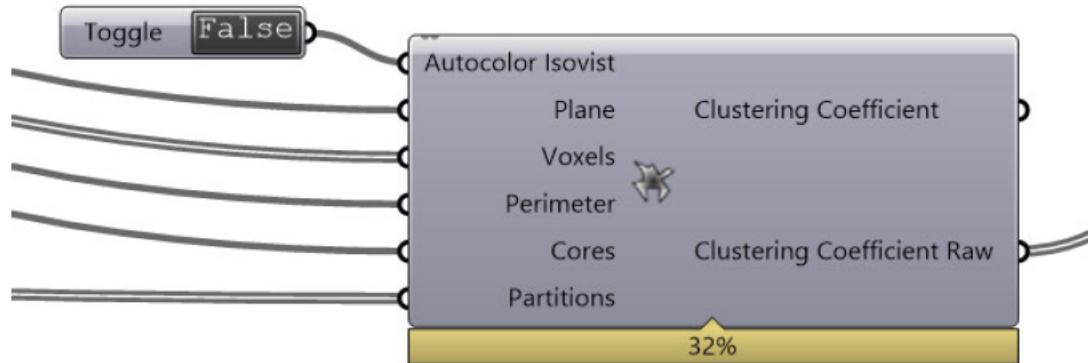


Source: Author's work



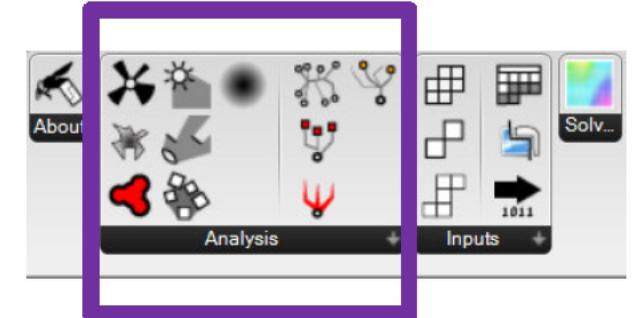
PlanBee Components: Analysis

Clustering Coefficient Metric

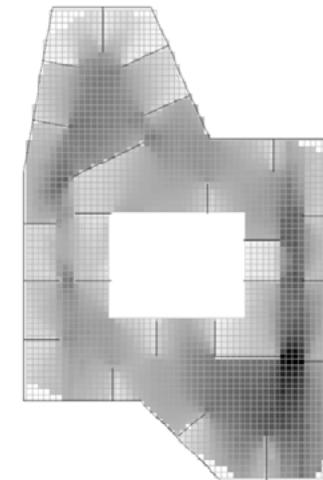


This metric is telling of how a person's field of vision changes as they move through space. There is a low coefficient if as you move through space your field of vision changes a lot. Similarly a high coefficient if your field of vision changes only slightly. For more on this please take a look at:

[From Isovists to Visibility Graphs A Methodology for the Analysis of Architectural Space](#)



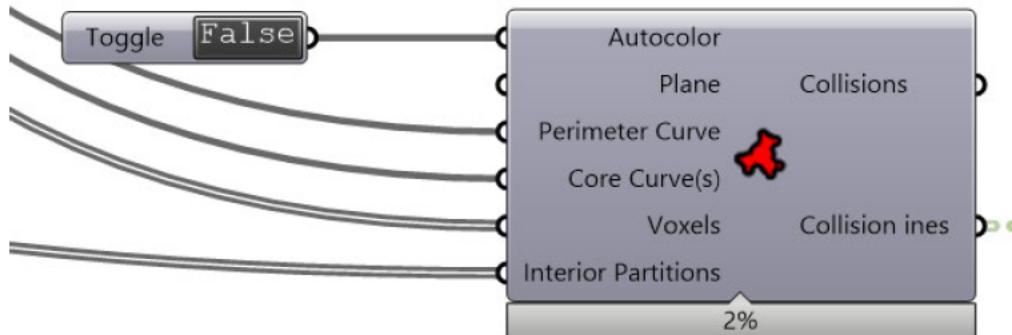
Source: Author's work



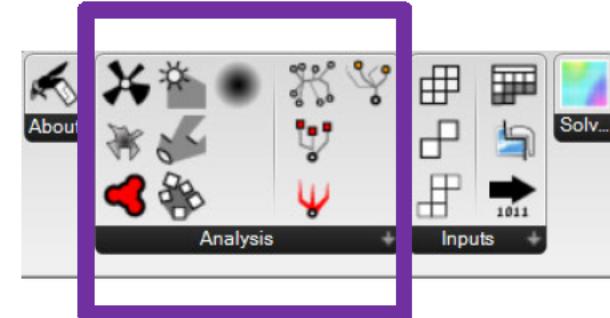
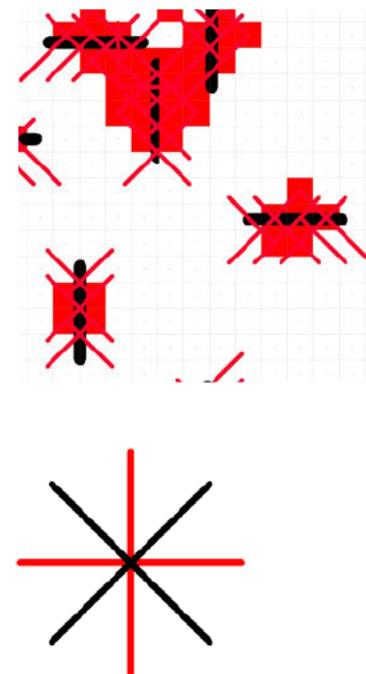
Legend:
-High coefficient factor = bright
-Low coefficient factor = dark

PlanBee Components: Analysis

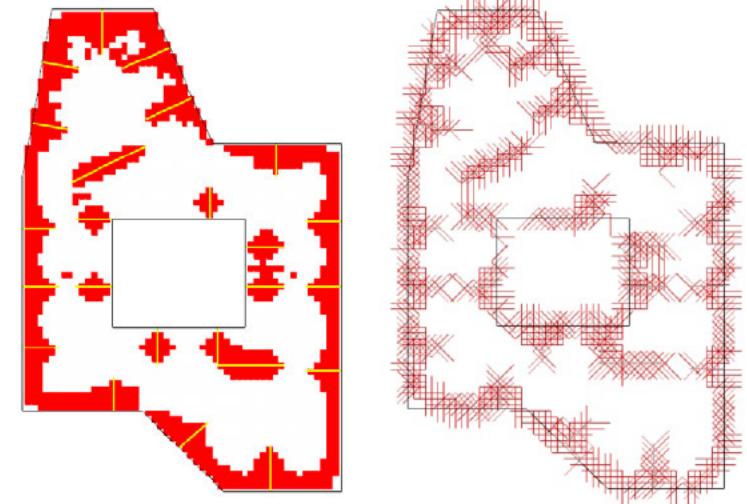
Covid Metric



A simple proxy metric to help determine whether a cell is 'compromised'. If two different rays which are 90-270 degrees apart are colliding with obstacles, the cell is deemed 'compromised'. On metric projects (meters) it uses 2m, on imperial (feet) it uses 6'.



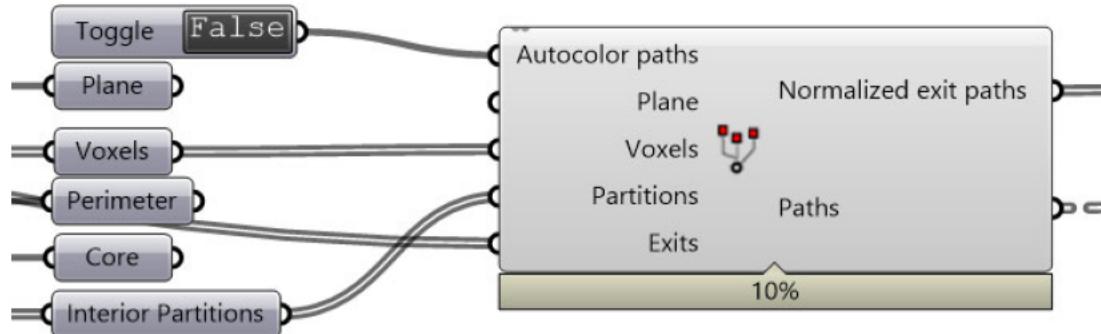
Source: Author's work



Legend:
-Red = compromised cell

PlanBee Components: Analysis

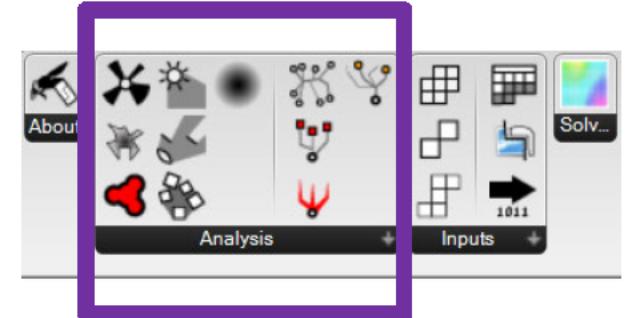
Exit Path/Distance Metric



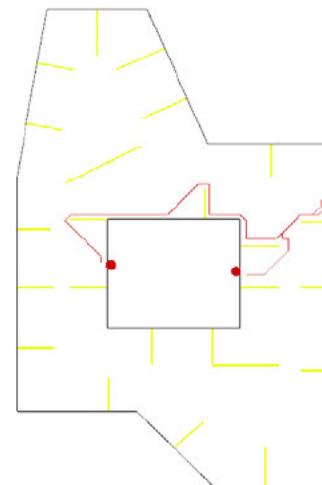
-A* Shortest Path algorithm implementation from each cell to specified target points.

-QuickGraph (Dijkstra shortest path algorithm) library implementation also available.

*This component and other path components only accept a max amount of target locations (keep to a minimum)

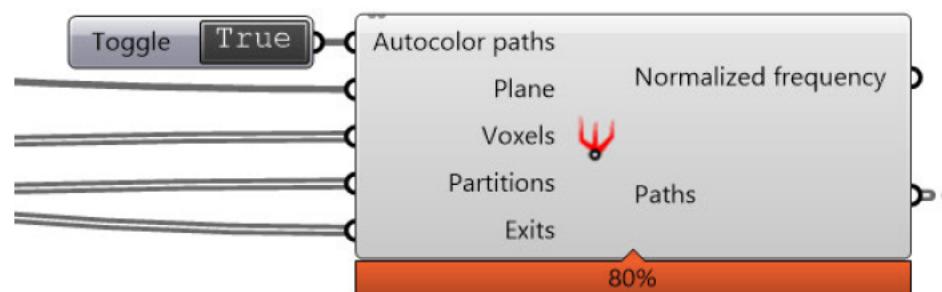


Source: Author's work



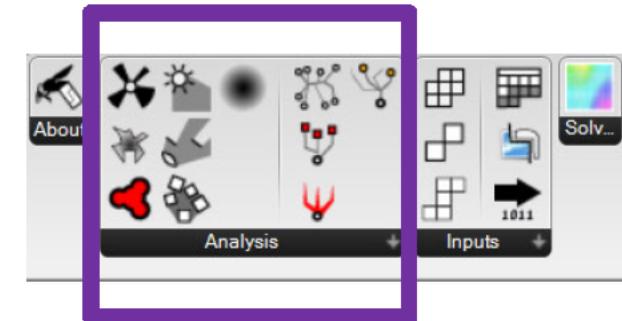
PlanBee Components: Analysis

Exit Paths Most Travelled Metric

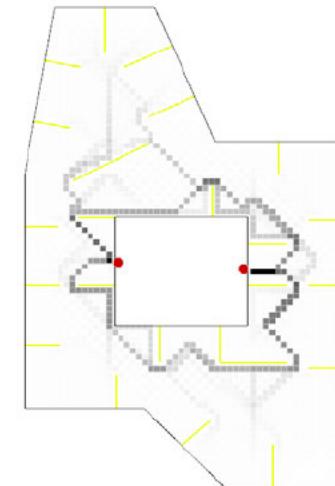


-Building on previous exit path travelled metric, this one looks at the frequency of each cell in all the paths of travelled.

*This component and other path components only accept a max amount of target locations (keep to a minimum)

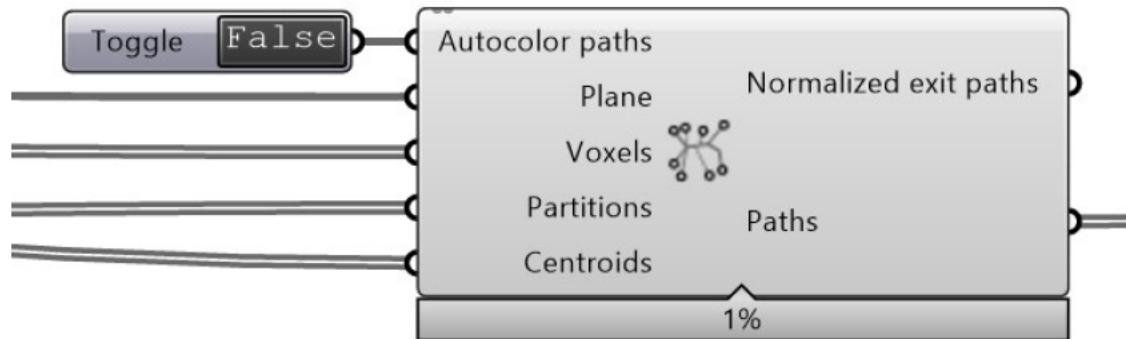


Source: Author's work



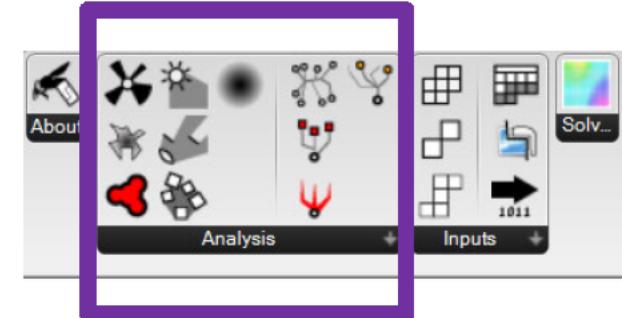
PlanBee Components: Analysis

'Between Points' Most Travelled

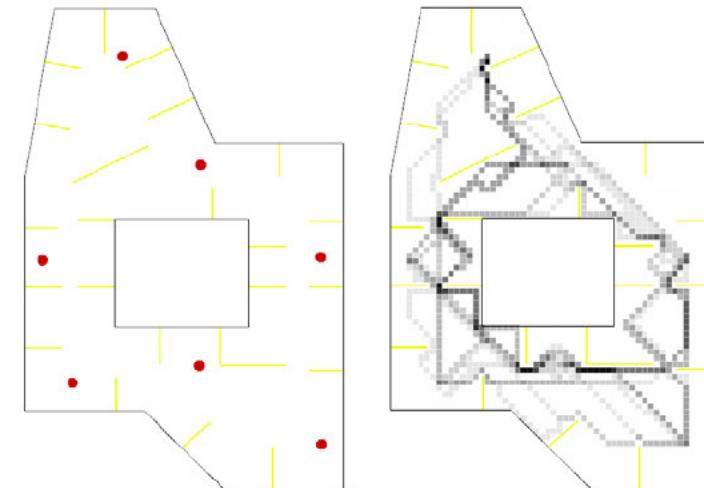


-Finds paths most traveled between a set of specified points. Each cell is given a metric based on how frequently travelled it is.

*This component and other path components only accept a max amount of target locations (keep to a minimum).

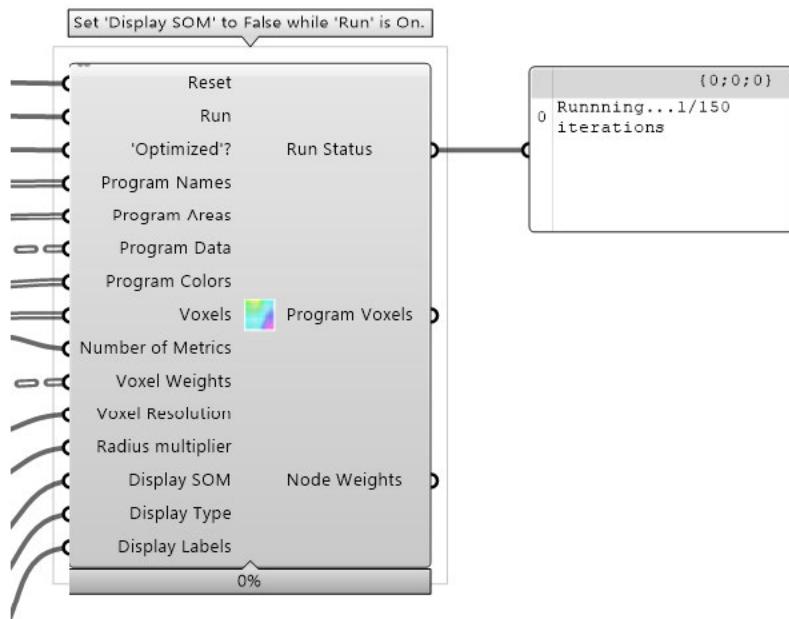


Source: Author's work



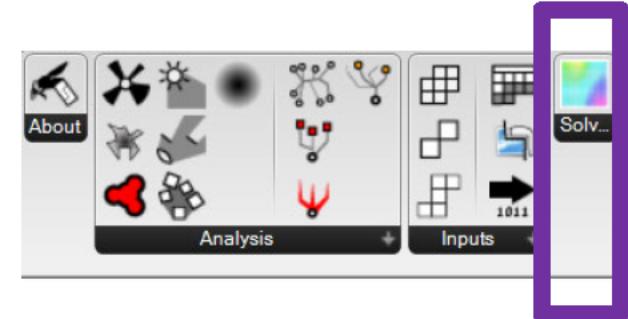
PlanBee Components: KSOM Solver

Kohonen SOM (WIP)

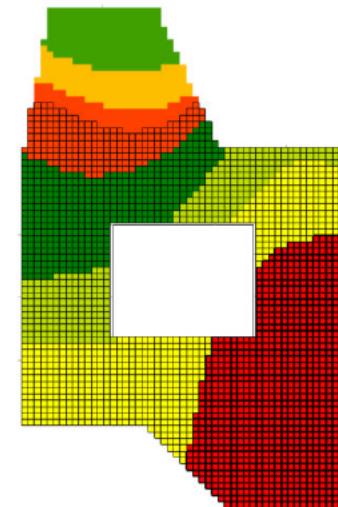


Produces a 2D 'feature map' describing the various metrics of the plan.

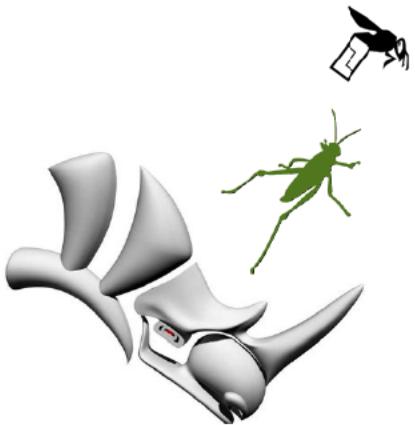
*The column count of the *feature* spreadsheet (not including the first 'program' column) must match the number of items in each branch of the 'Voxel Weights' as the example file suggests.



Source: Author's work



Quick demo...



Thanks!

Marco Juliani
marco.juliani@crtkl.com |
marcotjuliani@gmail.com
 @_m_juliani