Check for updates

EPB: Urban Analytics and City Science
2022, Vol. 0(0) 1–17
© The Author(s) 2022
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/23998083221133827
journals.sagepub.com/home/epb
SAGE

The cityseer Python package for pedestrian-scale network-based urban analysis

Gareth Simons
Building Stock Lab, UCL Energy Institute, BSEER, UCL, London, UK

Abstract

cityseer-api is a Python package consisting of computational tools for fine-grained street-network and land-use analysis, helpful in assessing the morphological precursors to vibrant neighbourhoods. It is underpinned by network-based methods developed specifically for urban analysis at the pedestrian scale. cityseer-api computes a variety of node and segment-based network centrality methods, land-use accessibility and mixed-use measures, and statistical aggregations. Accessibilities and aggregations are computed dynamically over the street-network while taking walking distance thresholds and the direction of approach into account, and can optionally incorporate spatial impedances and network decomposition to increase spatial precision. The use of Python facilitates compatibility with popular computational tools for network manipulation (NetworkX), geospatial topology (shapely), geospatial data state management (GeoPandas), and the NumPy stack of scientific packages. The provision of robust network cleaning tools aids the use of OpenStreetMap data for network analysis. Underlying loop-intensive algorithms are implemented in Numba JIT compiled code so that the methods scale efficiently to larger cities and regions. Online documentation is available from cityseer.benchmarkurbanism.com, and the Github repository is available at github.com/benchmark-urbanism/cityseer. Example notebooks are available at cityseer.benchmarkurbanism.com/examples/

Keywords

Computation, data science, geographical information systems, land-use analysis, morphometrics, network analysis, spatial analysis, urban analytics, urban planning, urban morphology, urbanism

Overview

cityseer-api is a street-network based spatial analysis package for pedestrian-scale urban morphological analysis. It combines street-network decomposition, distance-weighted implementations of network centrality and mixed-use measures, and contextually sensitive distance and aggregational methods to generate observations with a high degree of spatial precision. The synthesis of

Corresponding author:
Gareth Simons, UCL Energy Institute, BSEER, UCL, Building Stock Lab, Gower Street, London WC1E 6BT, UK.
Email: gareth.simons@ucl.ac.uk

these techniques enables cityseer-api to describe variances in morphological metrics at regular intervals along streetfronts in a manner facilitating comparative analysis of walkability, mixed-uses, and land-use accessibilities for urban planning scenarios.

Centrality and land-use analysis methods rely extensively on shortest-path algorithms, presenting substantial computational complexity due to nested computational loops. Pure Python network-based measures, such as those implemented in NetworkX (Hagberg et al., 2008) are consequently prohibitively slow if applied to analysis for larger towns and cities. Performance improvements can be attained through use of packages such as Graph-Tool (Peixoto, 2014), igraph (Csardi and Nepuszm, 2006), depthmapX (depthmapX development team, 2017), or pandana (Foti et al., 2012), which wrap underlying optimised C or C++ code. However, reliance on packages underpinned by lower-level programming languages presents a challenge for explorative research because it becomes difficult to manipulate underlying algorithms without incurring complexity or a loss of computational efficiency. This conundrum prompted the development of the codebase formalised as cityseer-api, which has adopted an approach leveraging pure Python and NumPy (Harris et al., 2020), but with computationally intensive loops optimised through use of Numba JIT (just-in-time) compilation (Lam et al., 2015). This approach has allowed for wide-ranging experimentation while permitting a set of pertinent issues to be addressed:

1. cityseer-api employs a ‘moving-window’ form of spatial analysis: each node in the network is visited in turn, with the network then isolated at a range of specified walking distance thresholds from the currently selected node. Centrality, land-use, and aggregational methods can then be computed for the locally windowed context. This is similar to radial forms of analysis used in Space Syntax (in the context of street-network centralities) and the notion of ‘overlapping buffer queries’ used in pandana (within the context of land-use accessibilities and data aggregations). Such forms of windowed distance thresholds can be based on either crow-flies euclidean distances or true network distances (Cooper, 2015); cityseer-api takes the position that true network distances are most representative when working at smaller pedestrian distance thresholds, particularly when applied to land-use accessibilities and mixed-use calculations. Moving window analysis is advantageous because it clearly and consistently defines the network boundary in relation to the current point of analysis, therefor sidestepping issues such as the robust definition of town or city boundaries, prevention of edge rolloff effects, and difficulties regarding normalisation of measures for comparisons between locations on differently sized networks.
2. It is common to use either shortest-distance or simplest-path (least angular ‘distance’) impedance heuristics when computing network centralities. When using simplest-path heuristics, it is necessary to modify the underlying shortest-path algorithms to prevent side-stepping of sharp angular turns; otherwise, two smaller side-steps can be combined to ‘short-cut’ sharp corners (Turner, 2007). This safeguard is not available in off-the-shelf network analysis packages. It is common for centrality methods to be applied to either primal network representations, generally used with shortest-path methods such as those applied by *multiple centrality assessment* analysis (Porta et al., 2006), or dual network representations, typically used with simplest-path methods in the tradition of *space syntax* (Hillier and Hanson, 1984). cityseer-api incorporates both forms of analysis while also allowing for angular centralities to be calculated on primal networks so that topological divergences between primal and dual networks do not skew observations comparing shortest and simplest path heuristics.
3. A range of centrality and mixed-use methods is available for urban analysis; cityseer-api incorporates specialised forms of these methods including distance-weighted versions greatly accentuating spatial precision. Some conventional methods, even if widely used, can

be problematic for urban analysis workflows: specifically, conventional formulations of closeness centrality do not behave as anticipated for windowed networks, and mixed-use analysis methods derived from larger-scale zoned or gridded aggregations become problematic if interpreted within the context of streets. cityseer-api incorporates implementations of these methods that are not susceptible to these issues. These methods and their implications are developed and explored at length in the accompanying papers on network centrality methods [Simons, n.d.a] and mixed-use methods [Simons, n.d.b];

4. Centrality methods are susceptible to topological distortions arising from ‘messy’ network representations as well as due to the conflation of topological and geometrical properties of street-networks, which has a detrimental impact on the calculation of network centralities. cityseer-api addresses these through the inclusion of robust network cleaning functions with substantial effort directed towards procedures for splitting geometrical properties from topological representations; the removal of parallel roadways; and the inclusion of segmentised forms of centrality measures, which are less susceptible to distortions introduced by varying intensities of nodes;
5. Pedestrian-scale analysis requires approaches facilitating the evaluation of respective measures at finely-spaced intervals along street fronts. Further, granular evaluation of land-use accessibilities and mixed-uses requires that land-uses be assigned to the street-network in a contextually precise manner. These are addressed in cityseer-api through the application of a network decomposition technique. Instead of assigning data-points to the nearest node, cityseer-api searches for the closest adjacent street edge and then uses a bidirectional assignment method. This allows for distances in relation to aggregations or accessibilities to be computed dynamically, while taking into account the direction of approach from the currently windowed node.

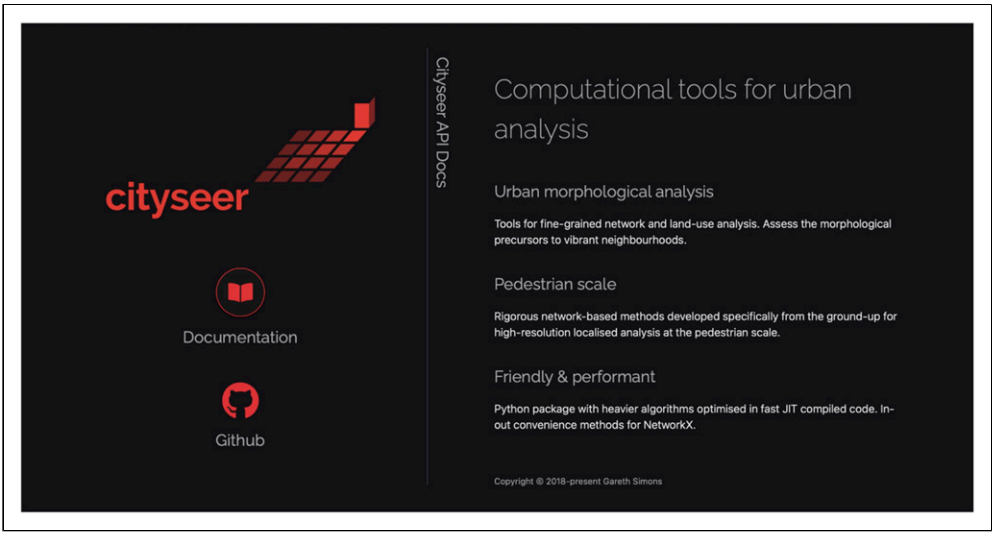
cityseer-api is intended to be data source agnostic and includes convenience methods for the general preparation of networks and their conversion into (and out of) the lower-level data structures used by the underlying algorithms. These network utility methods are designed to work with NetworkX to facilitate ease of use, and to enable workflows incorporating data from sources such as PostGIS or OpenStreetMap data, whether from API queries or from OSMnx (Boeing, 2017). Data state is managed with GeoPandas (Jordahl et al., 2020) to facilitate downstream analysis and modelling and for bridging to GeoPandas based workflows, such as used by momepy (Fleischmann, 2019).

Detailed package documentation is available at cityseer.benchmarkurbanism.com (Figures 1 and 2), including a guide and a growing collection of examples. Discussion and examples relating to use with other packages is provided in the documentation guide

A complement of code formatters, linters, type-checkers, and unit tests maintains the integrity of the code-base through general package maintenance and upgrade cycles. Where feasible, centrality methods are checked against NetworkX or against manually checked testing scenarios. Extensive mock data and test plots have been used to visually confirm the intended behaviour for divergent simplest and shortest-path heuristics and for confirming the assignment and aggregation of data-points.

Nuances of spatial aggregation

Computational tools have dramatically increased the range and depth of scientific analysis. Likewise, methods applied to spatial analysis have been revolutionised and hold tremendous potential for rigorous and scalable forms of urban analysis, which may prove helpful as benchmarking tools for principles espoused in urban theory and policy. Nevertheless, computational



Computational tools for urban analysis

Urban morphological analysis

Tools for fine-grained network and land-use analysis. Assess the morphological precursors to vibrant neighbourhoods.

Pedestrian scale

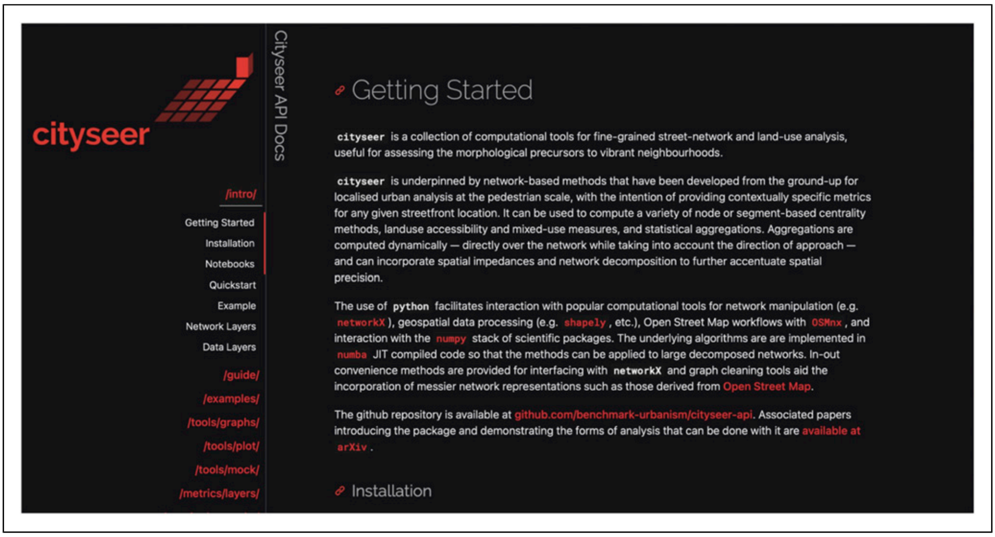
Rigorous network-based methods developed specifically from the ground-up for high-resolution localised analysis at the pedestrian scale.

Friendly & performant

Python package with heavier algorithms optimised in fast JIT compiled code. In-out convenience methods for NetworkX.

Copyright © 2018-present Gareth Simons

Figure 1. cityseer-api documentation homepage.



Getting Started

cityseer is a collection of computational tools for fine-grained street-network and land-use analysis, useful for assessing the morphological precursors to vibrant neighbourhoods.

cityseer is underpinned by network-based methods that have been developed from the ground-up for localised urban analysis at the pedestrian scale, with the intention of providing contextually specific metrics for any given streetfront location. It can be used to compute a variety of node or segment-based centrality methods, landuse accessibility and mixed-use measures, and statistical aggregations. Aggregations are computed dynamically — directly over the network while taking into account the direction of approach — and can incorporate spatial impedances and network decomposition to further accentuate spatial precision.

The use of python facilitates interaction with popular computational tools for network manipulation (e.g. networkX), geospatial data processing (e.g. shapely , etc.), Open Street Map workflows with OSMnx , and interaction with the numpy stack of scientific packages. The underlying algorithms are are implemented in numba JIT compiled code so that the methods can be applied to large decomposed networks. In-out convenience methods are provided for interfacing with networkX and graph cleaning tools aid the incorporation of messier network representations such as those derived from Open Street Map.

The github repository is available at github.com/benchmark-urbanism/cityseer-api. Associated papers introducing the package and demonstrating the forms of analysis that can be done with it are available at arXiv .

Installation

Figure 2. Getting started guide from the cityseer-api documentation website.

constraints and coarse data sources have historically favoured the use of larger areal units of spatial aggregation, such as spatial grids or areal zones, and these have typically been combined with simplified proximity methods such as crow-flies instead of network-based distance measures (Araldi and Fusco, 2016; Logan et al., 2017).

The use of crude spatial aggregations presents a dilemma from the perspective of urban design. The averaging of variables to overarching spatial units causes the loss of higher resolution information along streetfronts while obscuring relationships between observed variables for individual data-points, buildings, or plots. Statistical forms of analysis consequently encounter the *Ecological Fallacy*: correlations which may have been valid for a larger unit of analysis can become misleading if interpreted within the disaggregated context (Robinson, 2009); expressed differently, the context of a local street corner may be significantly removed from larger-scale statistical averages for a given neighbourhood, city, or region. More generally, aggregations can mask confounding variables (*Simpson's Paradox*) and the loss of information across 'geography' (space) and 'history' (time) may similarly confound spatially aggregated data (Cressie, 1998). This segues into the *Modifiable Areal Unit Problem* (MAUP) where statistical observations derived from spatially aggregated data is sensitive to the scale of aggregation; the arrangement of the data in relation to zonal extents; and spatial autocorrelation in the variables. As a rule-of-thumb, larger aggregations increase sensitivity to MAUP because of a smoothing effect in the distribution of the data due to decreasing levels of variance, with the implication that correlation coefficients will strengthen as the unit of aggregation increases (Fotheringham and Wong, 1991). Variance is likewise affected by spatial autocorrelation of variables or by the movement of boundaries relative to the geographic locations of data-points. Different spatial aggregations therefor trigger fluctuating and, sometimes, questionable statistical inferences if applied or interpreted across different scales of analysis or between varied zonal configurations (Robinson, 1956; Thomas and Anderson, 1965). These forms of problem are inherent to the use of spatially aggregated data and no simple solutions exist, with the issue proving particularly intractable for multivariate analysis. Nevertheless, attempts persist at better defining and managing the issue (Duque et al., 2018; Reynolds, 1998).

The expanding availability of spatially granular data sources combined with growing access to computational resources has begun to tip the scales in favour of higher resolution workflows capable of more contextually precise forms of spatial analysis (Yamada and Thill, 2010) that are less susceptible to aggregational artefacts. Further, rich data sources synthesised with street-network-based strategies heralds a paradigm shift from the aerial vantage point of the plan — traditionally the frame of reference for morphological analysis — to that of localised pedestrian-centric methods applied directly over the street-network (Araldi and Fusco, 2019): the pedestrian's vantage point can, in a literal sense, become the anchor and point of departure for spatial analysis.

Localised methods

'Moving-window' forms of a localised spatial analysis (also called 'radial' or 'buffered' methods) differ from gridded or zonal aggregations frequently used in Geographical Information Systems more widely. Calculations and aggregations are unfurled directly over the street-network at a set of selected distance thresholds: an algorithm visits each node in the network in turn; isolates the surrounding nodes at the specified distance thresholds; then centrality, land-use, or other aggregational or statistical measures can be computed for the currently selected location. The process subsequently repeats for every other node in the network, making it a 'localised' method because the calculations are repeated on an individual basis relative to each node (Figure 3). The full resolution of the data thus remains available to each sampled point: data remains spatially anchored and the distance from each point of analysis to each surrounding data-point is knowable (Figure 4). Contextually specific relationships between variables are therefor not sacrificed and it becomes possible to use spatial impedances to further accentuate locality relative to pedestrian walking tolerances. The dynamic nature of localised moving-window methods ameliorates the zonal aspect of the Modifiable Areal Unit Problem (MAUP) because the extents are defined consistently and methodically relative to the origin of each point of analysis and the scale of aggregation. Nevertheless, as with MAUP effects more generally, statistical variances tend to decrease for aggregations at increasingly large network distances, with the implication that correlations cannot be directly compared between smaller and larger distance thresholds.

Whereas aggregations at increasingly small distance thresholds become more locally focussed, there is a point at which distance cutoffs can become too small in relation to the topological

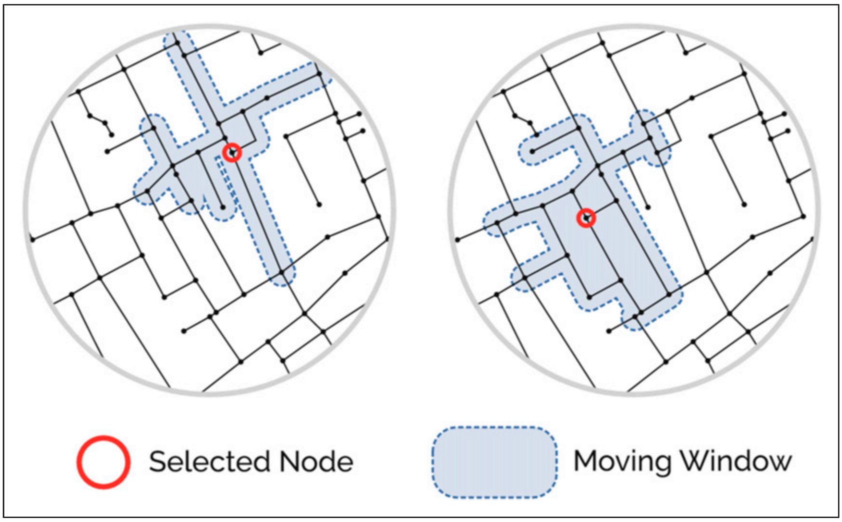


Figure 3. Moving-window' localised network analysis: metrics are calculated for each node in the network using a 'moving-window', taking into account all other nodes or data-points within a specified distance threshold.

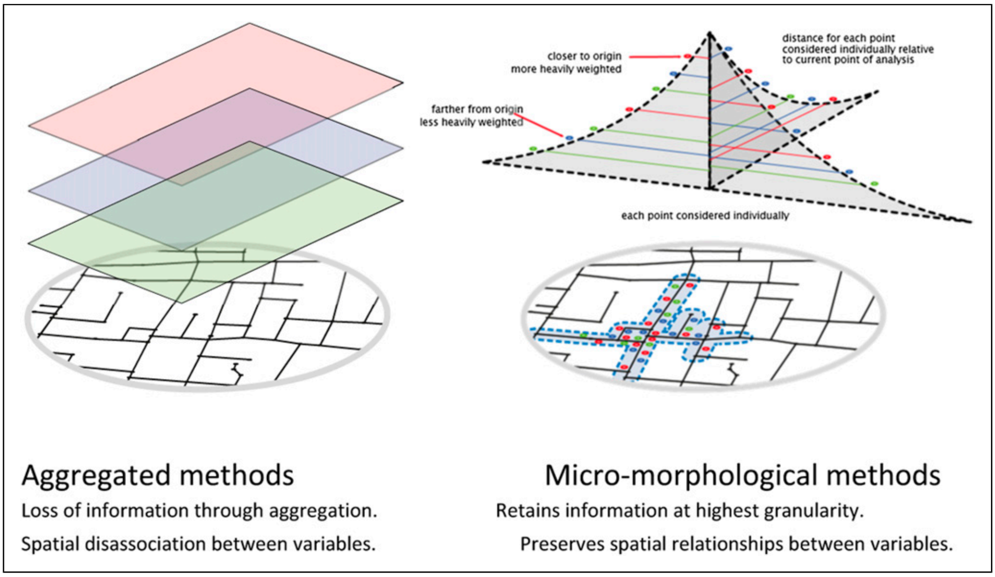

closer to origin
more heavily weighted
distance for each point
considered individually relative
to current point of analysis
farther from origin
less heavily weighted
each point considered individually
Aggregated methods
Loss of information through aggregation.
Spatial disassociation between variables.
Micro-morphological methods
Retains information at highest granularity.
Preserves spatial relationships between variables.

Figure 4. Overarching spatial analysis methods, where variables are aggregated to encompassing grids or zones (left), do not work well for purposes of urban analysis from the perspective of architecture and urbanism. This is because overarching methods collapse spatial information into larger-scale aggregations with the implication that data is no longer contextually relevant to a pedestrian's perception of space as defined by streetfronts. Unlike overarching aggregations, locally windowed methods (right) do not collapse data-points and do not discard relationships between variables relative to a selected location on the street-network. Locations and distances remain anchored relative to street-networks and walking tolerances from a given point in space.

structure of the network, potentially causing calculations or aggregations to become haphazard because longer street segments increasingly intersect or ‘overshoot’ smaller distance thresholds. These scenarios can cause a spatial ‘vacuum’ of information at the periphery of the moving-window and may consequently exacerbate variances in spatial aggregations, with ensuing artefacts in the distributions and a weakening of correlations. Another problem with longer street segments is that observations are not always sufficiently granular to provide intervening information between widely-spaced nodes on either side of longer street segments. Yamada and Thill (2010) applied a decompositional technique to street-networks such that no network edge is longer than a set maximum distance, thereby increasing the resolution of observations (Figure 5). A similar technique is exposed by the cityseer-api package: network edges can be ‘decomposed’ so that no edge is longer than a specified distance. The effect is that topological artefacts at smaller distance thresholds can be forestalled while permitting increasingly precise assignment of data-points (e.g. land-uses or spatially embedded information) to adjacent street edges, thus encapsulating particularities at more finely spaced intervals along street-fronts. The overlapping nature of moving-window analysis means that the resolution of data sampling can be increased through decomposition without changing the spatial units of analysis. Decomposition is further discussed in the next section.

In cityseer-api, data aggregation is routed via the two street-network nodes on either side of the closest adjacent street edge, thus facilitating dynamic selection of the direction and distance of aggregation appropriate to the location of the currently windowed node. The assignment of data-points to adjacent streets is achieved with the use of a winding algorithm (Figure 6), which first selects the closest adjacent node, then attempts to circle the street-network around the point of interest to identify the closest adjacent edge. If encountering a dead-end, the algorithm will backtrack and continue exploring. If exceeding the maximum search distance, it will then explore in the opposite winding direction to confirm whether any other closer edges exist.

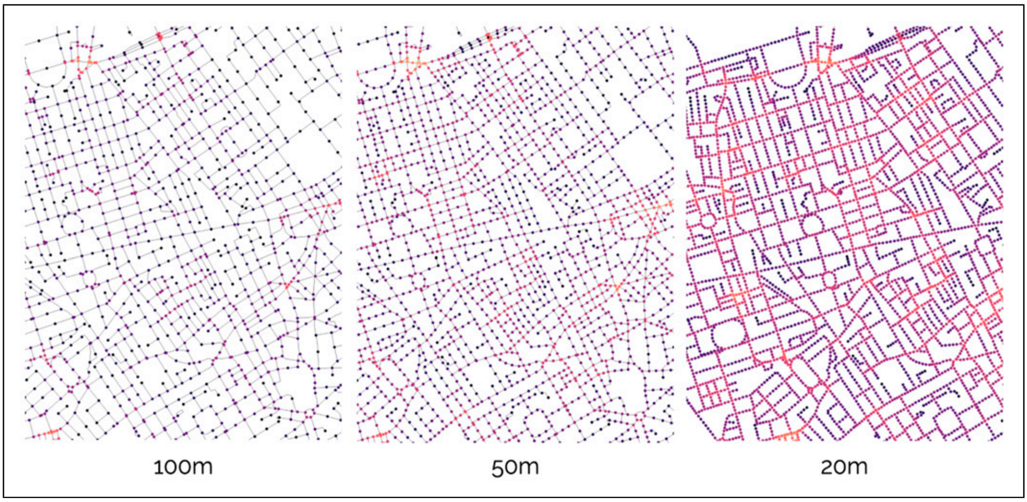
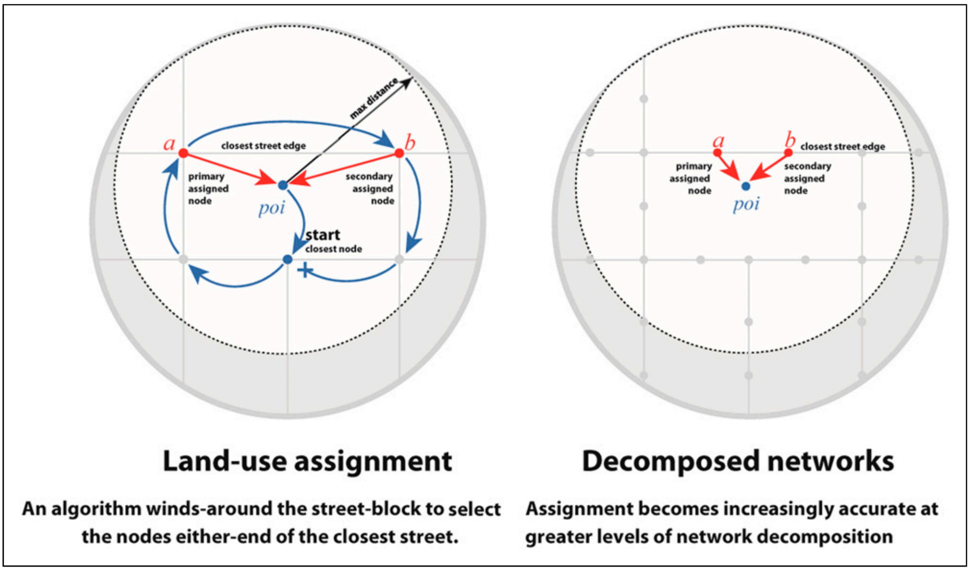


Figure 5. Varying levels of street network decomposition: Increasing the level of decomposition permits a higher resolution of analysis showing the variability of measures along street-fronts, allows for more precise assignments of land-uses to adjacent street segments, and reduces the potential loss of information where longer street segments might otherwise be discarded if intersecting a distance threshold.



Land-use assignment

An algorithm winds-around the street-block to select the nodes either-end of the closest street.

Decomposed networks

Assignment becomes increasingly accurate at greater levels of network decomposition

Figure 6. Datapoints representing land-uses are assigned to the two network nodes on either side of the nearest adjacent street edge. Therefore, accurate distances can be determined from each point of analysis to each datapoint via the closest adjacent node on the street-network while taking the direction of the approach into account. The algorithm 'winds' around the street-network to encircle the datapoint of interest to identify the closest adjacent street edge. The assignment increases in accuracy with increasing levels of network decomposition. In cityseer-api, the distance from the windowed node to a selected data-point includes the distance from the primary /secondary nodes to the datapoint's actual location in space.

Since data-points remain located precisely in space, and distances are known from each windowed node to each data-point's location relative to the street-network and the direction of approach, the contribution of any such data-point can be weighted on an individual basis when calculating land-use accessibilities or other aggregative or statistical measures. Pedestrian walking-distance weighted aggregation methods applied over decomposed networks can, thus, yield a particularly localised emphasis while allowing for the use of equivalently larger distance thresholds that are less likely to incur topological artefacts otherwise encountered by small distance thresholds. Similar to spatial interaction models, cityseer-api models spatial impedances (distance decays) using the negative exponential decay function

$$w = \exp(-\beta \cdot d) \tag{1}$$

which reflects a decreasing willingness for pedestrians to walk correspondingly farther distances. Accordingly, the contribution of a data-point is weighted by weight w as a function of the distance d. The rate at which this willingness to walk decreases is reflected in the strength of the specified β parameter (Figure 7). By default, cityseer -api anchors the strength of β relative to the selected maximum distance threshold of d_{max} using

$$\beta = 4/d_{max} \tag{2}$$

This conversion can be manually specified where greater control is required over the relationship between β and d_{max}; further information is provided in the documentation.

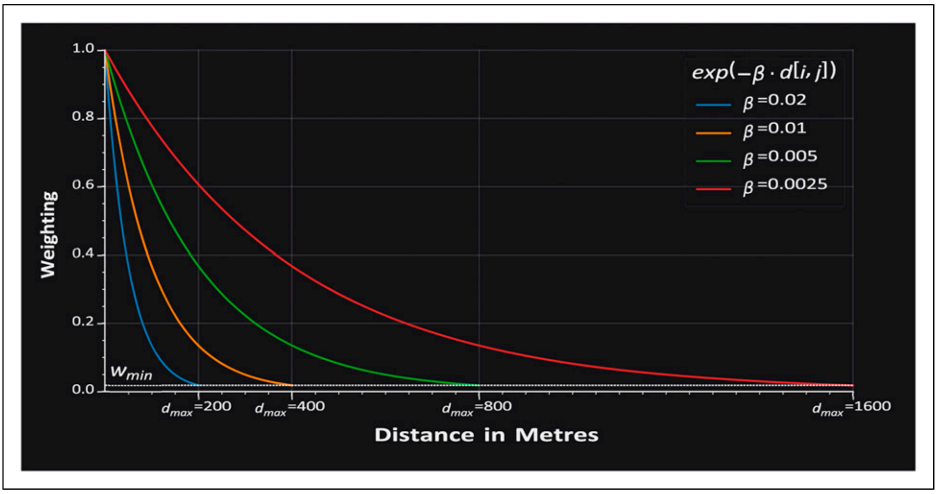


Figure 7. Spatial impedance curves for different β parameters. Nearer locations can be weighted more heavily than farther locations through use of the negative exponential decay (distance decay) function. The rate of the fall-off is controlled by the strength of β, where β is by default selected as $4/d_{max}$. The mapping from β to d_{max} (or vice-versa) can be customised and is explained more comprehensively in the package documentation.

Design decisions

Numba

cityseer-api makes use of Python and NumPy (Harris et al., 2020), with computationally intensive algorithms optimised through use of Numba JIT compilation (Lam et al., 2015). Python is a widespread programming language offering access to an extensive ecosystem of high-quality network (networkX, Hagberg et al., 2008), geospatial (shapely, Gillies, 2007), OpenStreetMap conversion (OSMnx, Boeing 2017), and data manipulation packages (pandas, McKinney, 2010). This facilitates general purpose workflows spanning from data munging and database I/O to interaction with an assortment of powerful data science and machine-learning packages such as sklearn (Pedregosa et al., 2011) and keras (Chollet, 2015).

The ease of use and flexibility of Python also entails a drawback: it offers slower performance when compared to lower-level languages such as C. For this reason, performance-critical Python packages are typically wrappers of code developed in more performant languages, with a prevalent example being the NumPy stack underpinning array-dependent operations central to a wide variety of Python's scientific computing packages. Network-based methods, which depend on loop-intensive low-level algorithms such as Dijkstra's shortest path remain a challenge, and it is for these purposes that the Numba package proves useful. Numba translates Python code into machine code using 'Just In Time' (JIT) compilation, offering performance similar to that of compiled lower-level languages such as C. Use of Numba thus infers convenient access to the Python ecosystem while facilitating experimentation with computationally complex algorithms.

Package composition

The cityseer-api package consists of three sub-packages: algos, consisting of the Numba optimised functions; metrics, consisting of higher-level Python code accessed by the end-user; and tools, a collection of utility modules for purposes such as network preparation using the cityseer.tools.graphs module and the generation and visualisation of mock data used by the unit tests.

The data structures and algorithms utilised by the algos sub-package's modules can be accessed directly; however, it is simpler to interact through the higher-level wrappers in the metrics sub-package. This contains the networks module for building street-networks and calculating street-network centralities as documented at cityseer.benchmarkurbanism.com/metrics/networks/. cityseer-api also broaches themes on land-use accessibilities, the mix of land-uses, and statistical aggregations, with these measures computed using the layers module as documented at cityseer.benchmarkurbanism.com/metrics/layers/. Whereas crude forms of these measures could be calculated using crow-flies distance aggregation methods, these become problematic when working at smaller pedestrian distance thresholds because the network structure can substantially affect distances to surrounding locations. The layers module is consequently underpinned by the same network structures and moving-window workflows utilised by the networks module. When functions contained in the layers module are invoked, data-points will be assigned to the specified street-network by invoking an algorithm that assigns each data-point to the closest adjacent street edge (see Figure 6). Metrics computed by the networks and layers modules are computed relative to the same network structure, with calculations saved to a GeoPandas DataFrame where they can be used for downstream statistical or machine learning analysis.

An important advantage to network-based distance methods and the bidirectional assignment of data-points to network nodes is that comparatively accurate distances are known from any selected network node to any accessible data-point, thus allowing distance-weighted methods to be applied. These techniques are explored in more detail in the accompanying papers on network centrality methods [Simons, n.d.a] and mixed-use methods [Simons, n.d.b].

Decomposition

A conundrum presents when calculating metrics on either a primal or dual network: architects, urban designers, and urbanists are interested in fine-scaled properties of the urban environment and how these properties can vary along street lengths. For example, characteristics at either street corner of a street segment may be notably different from that of the midpoint. One strategy may involve the interpolation of metrics to intervening locations, but this can be problematic for similar reasons; for example, if either end of a street segment has higher mixed-uses than the midpoint, interpolation will still give misleading results. The cityseer-api package therefor incorporates the optional use of network decomposition (Yamada and Thill, 2010). Each segment (edge) can be decomposed to a set maximum length (Figure 5) so that longer street segments are broken down into smaller sections. This strategy confers some advantages when working at small distance thresholds: measurement can be performed at a higher-resolution of analysis and becomes more contextual; data-points can be assigned to the network more precisely; and longer street segments are no longer problematic if intersecting a distance threshold. Whereas the decomposed version entails additional computational demands, the benefit is a greater number of sampled points at a finer resolution.

From the perspective of classic forms of network analysis — such as social networks, economic networks, web URL links, or citation networks — the idea of decomposition may seem nonsensical. However, this works for urban analysis because the nodes and edges are not being used as fundamentally discrete units of analysis in the same sense of individual persons, businesses, URLs, or publications. These are, instead, used in a murkier sense as proxies mapping to the adjacently accessible street-network. As such, these forms of urban analysis are not, per se, about 'intersections' as discrete points in space, but are instead about the availability of street frontages to pedestrians in a more continuous sense, and the resultant potential for social and economic activities as a function of the configuration of the street-network. This intuition is borne out in a conceptual sense by Space Syntax's (Hillier and Hanson, 1984) use of the dual instead of the primal network representation, and in a practical sense by the use of street-length weighted (Turner, 2007) or building weighted analysis (Sevtsuk and Mekonnen, 2012). In short, longer streets or more granular street-front typologies represent larger generators of activity.

When applying network decomposition within the context of urban analysis, measures such as land-use accessibilities and mixed-uses increase in resolution and become more accurate but are not otherwise substantially affected because these are simple distance-weighted aggregations from a selected point on the street-network. The benefit of decomposition is that distance calculations and land-use assignments become increasingly precise, and the number of intervening nodes otherwise does not affect the calculations. Topological network centrality measures, such as network cycles, are likewise not greatly affected, though are now sampled at more frequent intervals on the street-network. On the other hand, aggregative node-based network centrality measures, such as closeness centrality and betweenness centrality, do behave differently on decomposed networks but not necessarily in a problematic sense:

1. Additional nodes result in additional summations, with the implication that the measures are not comparable across different levels of decomposition;
2. Decomposition behaves as an implicit form of length-normalisation akin to weighting by street lengths because longer segments will yield a greater number of decomposed nodes, and therefor a greater number of summations. This effect is beneficial because nodes will be spread more evenly across the network, thereby tempering distortions introduced by varying concentrations of nodes on messier network representations;
3. Decomposition introduces $degree = 2$ nodes, and this can result in changes in the output distributions for aggregative centrality measures when calculated for small distance thresholds less than $200m$. See [Simons, n.d.a] for further discussion.

An alternative to node-based centrality measures is segmentised (continuous) forms of centrality adapted from their node-based equivalents. These are explored in [Simons, n.d.a]. Segmentised measures explicitly acknowledge street-networks as a continuous rather than discretised form of analysis and remain stable when the network is decomposed; however, as with node-based measures, the distributions of the measures are affected by the introduction of $degree = 2$ nodes for small distance thresholds.

Example workflows

Example workflows are provided and maintained on the examples page of the documentation, with links to Jupyter notebooks.

Current examples include:

1. A getting started guide.
2. An example workflow for cleaning and preparing OpenStreetMap data for analysis.
3. An example for how OpenStreetMap data can be imported and converted from OSMnx.
4. A demonstration showing how to compute network centralities for London (OpenStreetMap and OS Open Roads data).
5. A demonstration showing how to compute pub accessibility for London.

Graph cleaning and preparation

The tools.graphs module contains a collection of convenience functions for the preparation and conversion of networkX MultiGraphs, i.e., undirected networks allowing for multiple edges in cases where divergent street segments connect the same nodes.

These functions are designed to work with raw shapely Linestring geometries assigned to edges as geom attributes. The benefit of this approach is that the geometry of the network remains decoupled from the topology: the topology is consequently free from distortions in node intensities which would otherwise confound centrality measures.

When creating a street-networks for analysis, two common scenarios might include:

1. Source datasets that keep the topology of the network separate from a street's geometry. This is the ideal case and the network can be constructed directly from the topology while assigning the roadway geometries to the respective edges spanning the nodes. *Ordnance Survey* Open Roads is an example of this type of dataset. Assigning the geometries to an edge involves firstly casting the geometry to a shapely Linestring, then assigning this geometry to the respective edge as a 'geom' attribute. i.e., G[start_node][end_node][edge_idx]['geom'] = linestring_geom.
2. Data sources that represent roadway geometries by adding additional nodes to the topological network. This is not desirable because this technique introduces topological distortions. In these cases, the Graph Cleaning guide should be followed: the graphs.nx_simple_geoms function can be used to generate street geometries and then several functions can be applied to further clean and prepare the network for analysis, including nx_wgs_to_utm for WGS to UTM coordinate conversions; nx_remove_dangling_nodes to remove roadway stubs and disconnected portions of the network, nx_remove_filler_nodes to strip out unnecessary filler nodes, and nx_consolidate_nodes to consolidate nodes.

Related examples are provided in the documentation examples, with example images shown in Figure 8.

Computationally efficient analysis

After network preparation and cleaning is completed, the networkX network can be transformed into the data structures used by the networks and layers modules for efficiently computing centralities, land-use measures, and statistical aggregations. This is done by calling the graphs.network_structure_from_nx function to convert a networkX network into a GeoPandas GeoDataFrame representing the data state for each node, and a structures.NetworkStructure containing detailed information of the network for use by underlying algorithms.

The networks.node_centrality and networks.segment_centrality methods wrap underlying Numba optimised functions for computing a range of available centrality methods. Specified measures and distance thresholds are computed simultaneously to reduce the time required for multi-variable and multi-scalar workflows. The results of the computations are written to the nodes GeoDataFrame for downstream analysis.

Land-use and statistical measures require a GeoPandas GeoDataFrame representing data-points. cityseer-api automatically routes the location of each data-point through the two closest network nodes, one in either direction, as determined from the closest adjacent street edge. This permits cityseer-api to use dynamic spatial aggregation methods that more accurately describe distances from the perspective of pedestrians travelling over the network, and relative to the direction of approach.

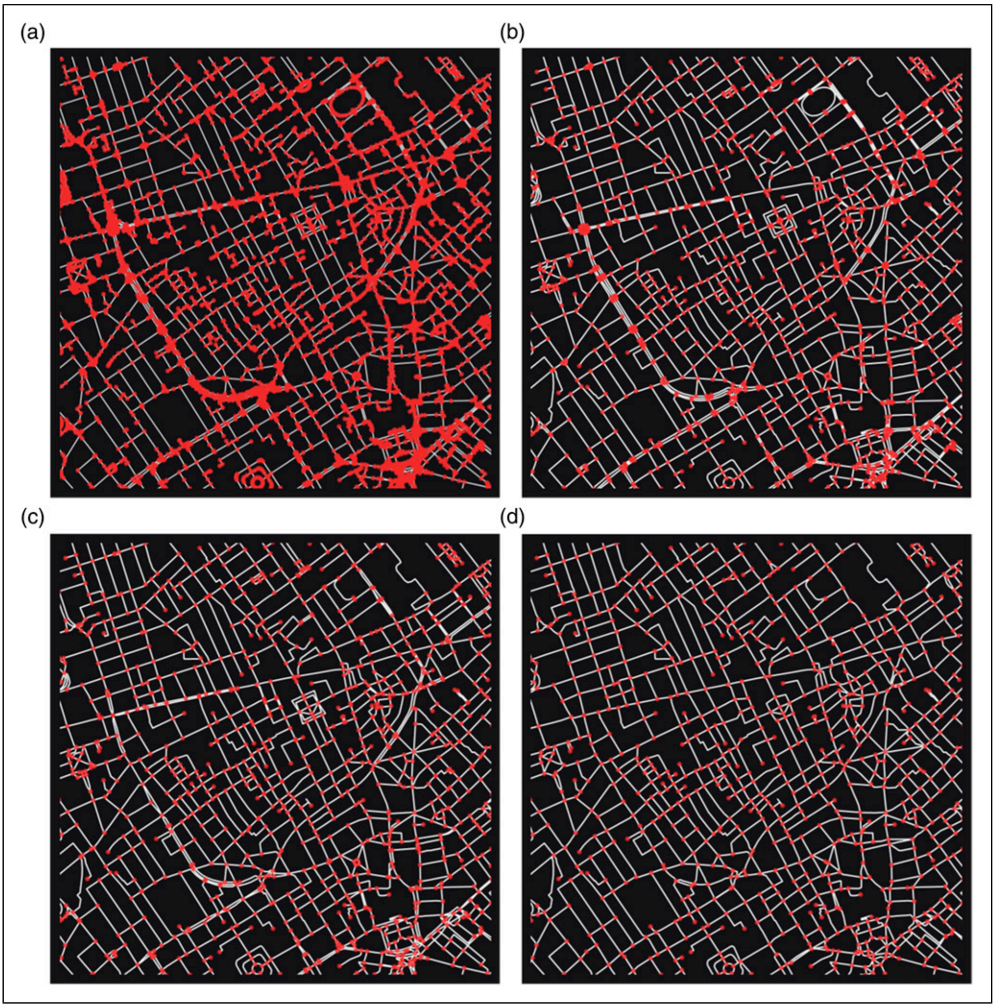


Figure 8. Examples from typical network cleaning and preparation workflows. See the documentation examples for links to Jupyter notebooks. (a) Raw network data as downloaded from OpenStreetMap prior to cleaning. (b) Graph data after removal of redundant 'filler' and 'dangling' nodes (c) After an initial pass of network consolidation showing simplified intersections. (d) After a second pass of consolidation with parallel roadways removed.

The layers.compute_accessibilities and layers.compute_mixed_uses functions are used for the calculation of land-use accessibility and mixed-use measures. In these cases, GeoDataFrame columns are used to represent categorical land-use information (e.g. 'pub', 'shop', 'school'). As with the centrality methods, land-use measures are computed simultaneously for all selected forms of analysis. Stand-alone methods are also available for some of the more commonly used mixed-use methods.

The layers.compute_stats function is used for statistical aggregations. In this case, GeoDataFrame columns are used to represent numerical information.

Land-use metrics and statistical aggregations are computed over the street-network relative to the network, with results written to each node. The mixed-use, landuse accessibility, and statistical

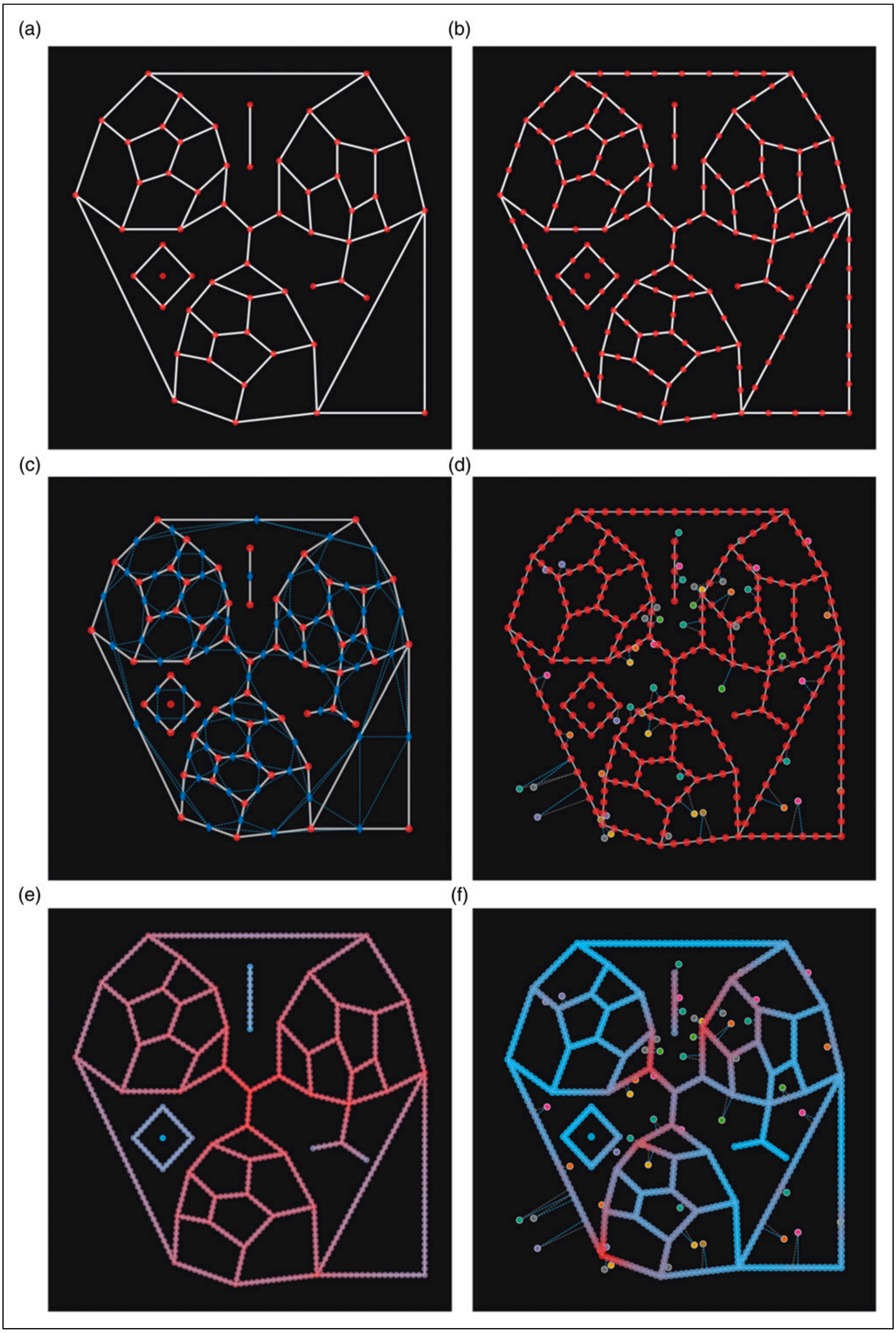


Figure 9. See the documentation examples for links to Jupyter notebooks. (a) A mock network for demonstration. (b) The network after decomposition (c) The network after primal to dual conversion. (d) Bi-directional assignment of data-points to adjacent edges. (e) 800 m harmonic centrality (f) 400 m distance-weighted mixed-uses.

aggregations can therefor be compared directly to centrality computations performed from the same locations, which can then be fed to downstream statistical or machine-learning analysis. Data derived from the cityseer.metrics package can be converted back into a NetworkX network using the graphs.nx_from_network_structure function, which can also overlay computed metrics onto the original network if provided as a nx_multigraph parameter to the function.

Related examples are provided in the documentation examples, with example images from the getting started guide shown in Figure 9.

Summary

cityseer-api contributes a synthesis of computational techniques to support granular forms of network-based spatial analysis from the perspective of pedestrians:

1. High-resolution workflows using localised moving-window analysis with strict network-based walking distance thresholds; spatially precise assignment of landuses or other data-points to adjacent street-fronts for improved contextual sensitivity; dynamic aggregation workflows which aggregate data-points and compute distances on-the-fly from any selected point on the network to any accessible land-use or data-point within a selected distance threshold; facilitation of workflows eschewing intervening steps of aggregation and associated issues such as ecological correlations; and the optional use of network decomposition to increase the resolution of the analysis.
2. Computation of network centralities using either shortest or simplest path heuristics on either primal or dual networks, including tailored methods such as harmonic closeness centrality, which behaves more suitably than traditional variants of closeness, and segmentised versions of centrality, which convert centrality methods from a discretised to an explicitly continuous form, see [Simons, n.d.a].
3. Land-use accessibilities and mixed-use calculations incorporate dynamic and directional aggregation workflows with the optional use of spatial-impedance weighted forms. These can likewise be applied with either shortest or simplest path heuristics and on either primal or dual networks, see [Simons, n.d.b].
4. Network centralities dovetailed with land-use accessibilities, mixed-uses, and general statistical aggregations from the same points of analysis to generate multi-scalar and multi-variable datasets facilitating downstream data science and machine-learning workflows, see [Simons, n.d.c] and [Simons, n.d.d].
5. The inclusion of network cleaning methods reducing topological distortions for high quality network analysis and aggregation workflows while accommodating workflows bridging the wider Numpy and GeoPandas ecosystem of scientific and geospatial packages.
6. Numba JIT compilation of underlying loop-intensive algorithms allows for these methods to be applied to large and, optionally, decomposed networks, which have greater computational demands.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Gareth Simons https://orcid.org/0000-0003-3790-0638

References

Araldi A and Fusco G (2016) Urban form from the pedestrian point of view: spatial patterns on a street network. In: SiTI, sui Sistemi Territoriali per l'Innovazione ISMB IS and di Torino ISMBDP (ed) 9th International Conference on Innovation in Urban and Regional Planning (INPUT 2016), Turin, Italy, 14–15 September 2016, pp. 32–38. fr/hal-01417484. Series Title: E-agora for the transition towards resilient communities` - INPUT 2016, Conference Proceedings. https://hal.archives-ouvertes

Araldi A and Fusco G (2019) From the street to the metropolitan region: pedestrian perspective in urban fabric analysis. *Environment and Planning B: Urban Analytics and City Science* 46(7): 1243–1263. DOI: 10.1177/2399808319832612

Boeing G (2017) OSMnx: new methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems* 65: 126–139. DOI: 10.1016/j.compenvurbsys.2017.05.004

Chollet F (2015) Keras. https://keras.io

Cooper CHV (2015) Spatial localization of closeness and betweenness measures: a selfcontradictory but useful form of network analysis. *International Journal of Geographical Information Science* 29(8): 1293–1309. DOI: 10.1080/13658816.2015.1018834

Cressie N (1998) Aggregation and interaction issues in statistical modeling of spatiotemporal processes. *Geoderma* 85(2): 133–140. DOI: 10.1016/S00167061(98)00017-2

Csardi G and Nepusz T (2006) The igraph software package for complex network research. InterJournal Complex Systems: 1695. https://igraph.org

depthmapX development team (2017) depthmapX. https://github.com/SpaceGroupUCL/depthmapX/

Duque JC, Laniado H and Polo A (2018) S-maup: statistical test to measure the sensitivity to the modifiable areal unit problem. *PLOS ONE* 13(11): e0207377. DOI: 10.1371/journal.pone.0207377

Fleischmann M (2019) momepy: urban morphology measuring toolkit. *Journal of Open Source Software* 4(43): 1807. DOI: 10.21105/joss.01807

Fotheringham AS and Wong DWS (1991) The modifiable areal unit problem in multivariate statistical analysis. *Environment and Planning A: Economy and Space* 23(7): 1025–1044. DOI: 10.1068/a231025

Foti F, Waddell P and Luxen D (2012) *A Generalized Computational Framework for Accessibility: From the Pedestrian to the Metropolitan Scale*. Karlsruhe, Germany: Karlsrühe Institute of Technology.

Gillies S (2007) Shapely: manipulation and analysis of geometric objects. https://github.com/Toblerity/Shapely

Hagberg AA, Schult DA and Swart PJ (2008) Exploring network structure, dynamics, and function using NetworkX. In: Varoquaux G, Vaught T and Millman J (eds) Proceedings of the 7th Python in Science Conference. Pasadena, CA, 19–24 August 2008, pp. 11–15.

Harris CR, Millman KJ, van der Walt SJ, et al. (2020) Array programming with NumPy. *Nature* 585(7825): 357–362. DOI: 10.1038/s41586-020-2649-2

Hillier B and Hanson J (1984) *The Social Logic of Space*. Cambridge, UK: Cambridge University Press.

Jordahl K, Bossche JVd, Fleischmann M, et al. (2020) geopandas/geopandas: v0.8.1. DOI: 10.5281/zenodo.3946761

Lam SK, Pitrou A and Seibert S (2015) Numba: a LLVM-based Python JIT compiler. In: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, Austin, TX, 15 November 2015. ACM Press, pp. 1–6. DOI: 10.1145/2833157.2833162

Logan TM, Williams TG, Nisbet AJ, et al. (2017) Evaluating urban accessibility: leveraging open-source data and analytics to overcome existing limitations. *Environment and Planning B: Urban Analytics and City Science* 46(5): 897–913. DOI: 10.1177/2399808317736528

McKinney W (2010) Data structures for statistical computing in python. In: van der Walt S and Millman J (eds) Proceedings of the 9th Python in Science Conference, Austin, TX, 28 June –3 July 2010, pp. 51–56.

Pedregosa F, Varoquaux G, Gramfort A, et al. (2011) Scikit-learn: machine learning in python. *Journal of Machine Learning Research* 12: 2825–2830.

Peixoto TP (2014) The graph-tool python library. figshare. DOI: 10.6084/m9.figshare.1164194

Porta S, Crucitti P and Latora V (2006) The network analysis of urban streets: a primal approach. *Environment and Planning B: Planning and Design* 33: 705–725. DOI: 10.1068/b32045

Reynolds HD (1998) *The Modifiable Area Unit Problem, Empirical Analysis by Statistical Simulation*. PhD Thesis. Toronto, Canada: University of Toronto.

Robinson AH (1956) The necessity of weighting values in correlation analysis of areal data. *Annals of the Association of American Geographers* 46(2): 233–336. DOI: 10.1111/j.1467-8306.1956.tb01507.x

Robinson WS (2009) Ecological correlations and the behavior of individuals. *International Journal of Epidemiology* 38(2): 337–341. DOI: 10.1093/ije/dyn357

Sevtsuk A and Mekonnen M (2012) Urban network analysis: a new toolbox for ArcGIS. *RIG* 22: 287–305.

Simons GD (n.d.a) Network centrality measures and their correlation to mixed-uses at the pedestrian-scale. http://arxiv.org/abs/2106.14040. ArXiv: 2106.14040

Simons GD (n.d.b) The application of mixed-use measures at the pedestrian-scale. https://arxiv.org/abs/2106.14048. ArXiv: 2106.14048.

Simons GD (n.d.c) Untangling urban data signatures: unsupervised machine learning methods for the detection of urban archetypes at the pedestrian scale. https://arxiv.org/abs/2106.15363. ArXiv: 2106.15363.

Simons GD (n.d.d) Prediction of 'artificial' urban archetypes at the pedestrian-scale through a synthesis of domain expertise with machine learning methods. https://arxiv.org/abs/2106.15364. ArXiv: 2106.15364.

Thomas EN and Anderson DL (1965) Additional comments on weighting values in correlation analysis of areal data. *Annals of the Association of American Geographers* 55(3): 492–505.

Turner A (2007) From axial to road-centre lines: a new representation for space syntax and a new model of route choice for transport network analysis. *Environment and Planning B: Planning and Design* 34: 539–555. DOI: 10.1068/b32067

Yamada I and Thill JC (2010) Local indicators of network-constrained clusters in spatial patterns represented by a link attribute. *Annals of the Association of American Geographers* 100(2): 269–285. DOI: 10.1080/00045600903550337

Gareth Simons is an urbanist, urban data scientist, and geospatial software developer. His interests centre on the development of computational techniques for urban morphological analysis. He presently works for the UCL Energy Institute's Building Stock Lab where he develops building stock models at a national scale for energy analytics.