

¹ eo-tides: Tide modelling tools for large-scale satellite Earth observation analysis

³ **Robbi Bishop-Taylor**  ¹¶, **Claire Phillips**  ¹, **Stephen Sagar**  ¹, **Vanessa Newey**¹, and **Tyler Sutterley**  ²

⁵ 1 Geoscience Australia, Australia  ² University of Washington Applied Physics Laboratory, United States of America  ¶ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Open Journals](#) ↗

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright¹⁸ and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).¹⁹²⁰

Summary

The eo-tides package provides powerful parallelized tools for integrating satellite Earth observation (EO) data with ocean tide modelling. The package provides a flexible Python-based API that facilitates the modelling and attribution of tide heights to a time-series of satellite images based on the spatial extent and acquisition time of each satellite observation (Figure 1).

eo-tides leverages advanced tide modelling functionality from the pyTMD tide prediction software (Sutterley et al., 2017), combining this fundamental tide modelling capability with EO spatial analysis tools from odc-geo ([odc-geo development team, 2024](#)). This allows tides to be modelled in parallel automatically using over 50 supported tide models, and returned in standardised pandas ([McKinney, 2010](#); [pandas development team, 2020](#)) and xarray ([Hoyer & Joseph, 2017](#)) data formats for further analysis.

Tools from eo-tides are designed to be applied directly to petabytes of freely available satellite data loaded from the cloud using Open Data Cube's odc-stac or datacube packages (e.g. using [Digital Earth Australia](#) or [Microsoft Planetary Computer's SpatioTemporal Asset Catalogue](#)). Additional functionality enables evaluating potential satellite-tide biases, and validating modelled tides using external tide gauge data — both important considerations for assessing the reliability and accuracy of coastal EO workflows. In combination, these open source tools support the efficient, scalable and robust analysis of coastal EO data for any time period or location globally.

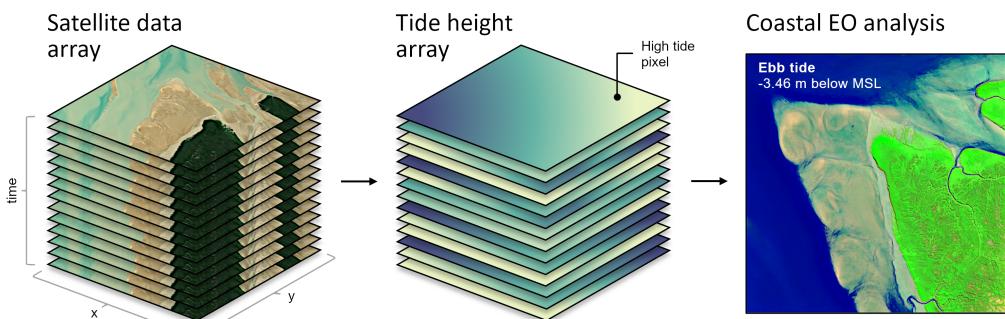


Figure 1: An example of a typical eo-tides coastal EO workflow, with tide heights being modelled into every pixel in a spatio-temporal stack of satellite data (for example, from ESA's Sentinel-2 or NASA/USGS Landsat), then combined to derive insights into dynamic coastal environments.

27 Statement of need

28 Satellite remote sensing offers an unparalleled method to view and examine dynamic coastal
29 environments over large temporal and spatial scales (Turner et al., 2021; Vitousek et al.,
30 2023). However, the variable and sometimes extreme influence of ocean tides in these regions
31 can complicate analyses, making it difficult to separate the influence of changing tides from
32 patterns of true coastal change over time (Vos et al., 2019). This is a particularly significant
33 challenge for continental- to global-scale coastal EO analyses, where failing to account for
34 complex tide dynamics can lead to inaccurate or misleading insights into coastal processes
35 observed by satellites.

36 Conversely, information about ocean tides can also provide unique environmental insights that
37 can greatly enhance the utility of coastal EO data. Conventionally, satellite data dimensions
38 consider the geographical “where” and the temporal “when” of data acquisition. The addition
39 of tide height as a new analysis dimension allows data to be filtered, sorted and analysed with
40 respect to tidal processes, delivering a powerful re-imagining of traditional multi-temporal EO
41 data analysis (Sagar et al., 2017). For example, satellite data can be analysed to focus on
42 specific ecologically-significant tidal stages (e.g. high, low tide, spring or neap tides) or on
43 particular tidal processes (e.g. ebb or flow tides).

44 This concept has been used to map tidally-corrected annual coastlines from Landsat satellite
45 data at continental scale (Bishop-Taylor et al., 2021), generate maps of the extent and elevation
46 of the intertidal zone (Bishop-Taylor et al., 2019; Murray et al., 2012; Sagar et al., 2017), and
47 create tidally-constrained imagery composites of the coastline at low and high tide (Sagar et
48 al., 2018). However, these approaches have been historically based on bespoke, closed-source
49 or difficult to install tide modelling tools, limiting the reproducibility and portability of these
50 techniques to new coastal EO applications. To support the next generation of coastal EO
51 workflows, there is a pressing need for new open-source approaches for combining satellite data
52 with tide modelling.

53 The `eo-tides` package aims to address these challenges by providing a set of performant
54 open-source Python tools for attributing satellite EO data with modelled ocean tides. This
55 functionality is provided in five main analysis modules (`utils`, `model`, `eo`, `stats`, `validation`)
56 which are described briefly below.

57 Setting up tide models

58 A key barrier to utilising tide modelling in EO workflows is the complexity and difficulty of
59 initially setting up global ocean tide models for analysis. To address this, the `eo_tides.utils`
60 module contains useful tools for preparing tide model data files for use in `eo-tides`. This
61 includes the `list_models` function that provides visual feedback on the tide models a user has
62 available in their system, while highlighting the naming conventions and directory structures
63 required by the underlying pyTMD tide prediction software (Figure 2).

64 Running tide modelling using the default tide modelling data provided by external providers can
65 be slow due to the large size of these files — especially for recent high-resolution models like
66 FES2022 (Carrere et al., 2022). To improve tide modelling performance, it can be extremely
67 useful to clip tide model files to a smaller region of interest (e.g. the extent of a country
68 or coastal region). The `clip_models` function can be used to automatically clip all suitable
69 NetCDF-format model data files to a user-supplied bounding box, potentially improving tide
70 modelling performance by over an order of magnitude.

71 These tools are accompanied by comprehensive documentation explaining how to set up several
72 of the most commonly used global ocean tide models, including details on how to download or
73 request access to model files, and how to uncompress and arrange the data on disk.

	Model	Expected path
	EOT20	tide_models/EOT20/ocean_tides
	FES2014	tide_models/fes2014/ocean_tide
	HAMTIDE11	tide_models/hamtide
	TPXO9.1	tide_models/TPXO9.1/DATA
...

Summary:

Available models: 2/50

Figure 2: An example output from `list_tides`, providing a useful summary table which clearly identifies available and supported tide models.

74 Modelling tides

75 The `eo_tides.model` module builds upon advanced tide modelling capability provided by
76 the `pyTMD` tide prediction software ([Sutterley et al., 2017](#)). `pyTMD` is an open-source tidal
77 prediction software that aims to simplify the calculation of ocean and earth tides. Tides are
78 frequently decomposed into harmonic constants, or constituents, associated with the relative
79 positions of the sun, moon and Earth. For ocean tides, `pyTMD.io` contains routines for reading
80 major constituent values from commonly available tide models, and interpolating those values
81 to spatial locations. Information for each of the supported tide models is stored within a
82 JSON database that is supplied with `pyTMD`. `pyTMD.astro` contains routines for computing the
83 positions of celestial bodies for a given time. Namely for ocean tides, `pyTMD` computes the
84 longitudes of the sun (S), moon (H), lunar perigee (P), ascending lunar node (N) and solar
85 perigee (PP). `pyTMD.arguments` contains routines for combining the astronomical coefficients
86 with the “Doodson number” of each constituent, along with routines for adjusting the amplitude
87 and phase of each constituent based on their modulations over the 18.6 year nodal period.
88 Finally, `pyTMD.predict` uses results from those underlying functions to predict tidal values at
89 a given location and time.

90 To support integration with satellite EO data, the `model_tides` function from `eo_tides.model`
91 takes tides predicted by `pyTMD` and returns them in a standardised pandas data format containing
92 information about the tide model, location and time period of each modelled tide. This allows
93 large analyses to be broken into smaller discrete chunks that can be processed in parallel before
94 being combined as a final step. Parallelisation in `eo-tides` is automatically optimised based
95 on the number of available workers and the number of requested tide models and analysis
96 points. This built-in parallelisation can significantly improve tide modelling performance for
97 EO analyses involving large spatial extents or multiple tide models ([Table 1](#)).

Table 1: An example benchmark comparison of tide modelling performance with parallelisation on vs. off. This comparison was performed across an 8-core and 32-core Linux machine, for an example large-scale analysis involving a month of hourly tides modelled across 10,000 spatial points and three tide models (FES2022, TPXO10, GOT5.6)

Cores	Parallelisation	No parallelisation	Speedup
8	180.2 s ± 119 ms	357.4 s ± 235 ms	183%
32	54.2 s ± 276 ms	360.3 s ± 835 ms	342%

98 Tide modelling functionality in the `model_tides` function is primarily intended to support more
 99 complex EO-related capability in the downstream `eo_tides.eo` module. However it can also
 100 be used independently of EO data, for example for any application that requires a time series
 101 of modelled tide heights. In addition to modelling tide heights, the `model_phases` function
 102 allows users to calculate tidal phases at any location and time. This can be used to classify
 103 tides into high and low tide observations, or determine whether the tide was rising (i.e. flow
 104 tide) or falling (i.e. ebb tide) at any point in time.

105 Combining tides with satellite data

106 The `eo_tides.eo` module contains the package's core functionality, focusing on tools for
 107 attributing satellite data with modelled tide heights. For tide attribution, eo-tides offers two
 108 approaches that differ in complexity and performance: `tag_tides` and `pixel_tides` (Table 2).
 109 The `tag_tides` function provides a fast and efficient method for small scale applications where
 110 tides are unlikely to vary across a study area. This approach allocates a single tide height
 111 to each satellite data timestep, based on the geographic-centroid of the dataset and the
 112 acquisition time of each image. Having tide height as a variable allows the selection and
 113 analysis of satellite data based on tides. For example, all available satellite observations for an
 114 area of interest could be sorted by tide height, or used to extract and compare the lowest and
 115 highest tide images in the time series.
 116 However, in reality tides vary spatially – potentially by many metres in height in areas of
 117 complex and extreme tidal dynamics. This means that an individual satellite image can capture
 118 a range of contrasting tide conditions. For larger scale coastal EO analysis, the `pixel_tides`
 119 function can be used to seamlessly model tides through both time and space, producing three-
 120 dimensional “tide height” datacube that can be integrated with satellite data. For efficient
 121 processing, `pixel_tides` ‘models tides into a customisable low resolution grid surrounding
 122 each satellite image in the time series. These modelled tides are then re-projected back into
 123 the original resolution of the input satellite image, returning a unique tide height for every
 124 individual satellite pixel through time (Figure 3).

Table 2: Comparison of the `tag_tides` and `pixel_tides` functions.

tag_tides	pixel_tides
<ul style="list-style-type: none"> - Assigns a single tide height to each timestep/satellite image - Ideal for local or site-scale analysis - Fast, low memory use - Single tide height per image can produce artefacts in complex tidal regions 	<ul style="list-style-type: none"> - Assigns a tide height to every individual pixel through time to capture spatial tide dynamics - Ideal for regional to global-scale coastal product generation - Slower, higher memory use - Produce spatially seamless results across large extents by applying analyses at the pixel level

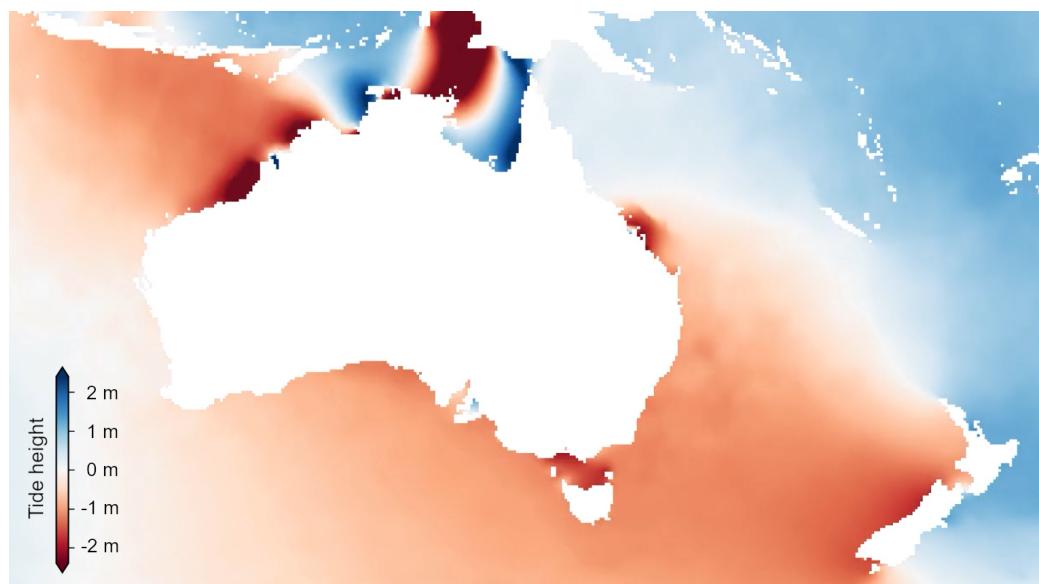


Figure 3: An example tide height output produced by the `pixel_tides` function, showing spatial variability in tides across Australasia for a single timestep.

125 Calculating tide statistics and satellite biases

126 The `eo_tides.stats` module contains tools for calculating statistics describing local tide
 127 dynamics, as well as biases caused by interactions between tidal processes and satellite orbits.
 128 Complex tide aliasing interactions between temporal tide dynamics and the regular overpass
 129 timing of sun-synchronous satellite sensors mean that satellites often do not always observe
 130 the entire tidal cycle (Elefeldt et al., 2014). Biases in satellite coverage of the tidal cycle can
 131 mean that tidal extremes (e.g. the lowest or highest tides at a location) or particular tidal
 132 processes may either never be captured by satellites, or be over-represented in the satellite
 133 record. Local tide dynamics can cause these biases to vary greatly both through time and space
 134 (Bishop-Taylor et al., 2019), making it challenging to compare coastal processes consistently -
 135 particularly for large-scale coastal EO analyses.

136 To ensure that coastal EO analyses are not inadvertently affected by tide biases, it is important
 137 to understand and compare how well the tides observed by satellites match the full range of
 138 modelled tides at a location. The `tide_stats` function compares the subset of tides observed
 139 by satellite data against the full range of tides modelled at a regular interval through time
 140 across the entire time period covered by the satellite dataset. This comparison is used to
 141 calculate several useful statistics that summarise how well a satellite time series captures the
 142 full range of real-world tidal conditions (Bishop-Taylor et al., 2019). These statistics include:

- 143 1. Spread: The proportion of the modelled astronomical tidal range that was observed by
 144 satellites. A high value indicates good coverage of the tide range.
- 145 2. High-tide offset: The proportion of the highest tides never observed by satellites, relative
 146 to the modelled astronomical tidal range. A high value indicates that the satellite data
 147 never captures the highest tides.
- 148 3. Low-tide offset: The proportion of the lowest tides never observed by satellites, relative
 149 to the modelled astronomical tidal range. A high value indicates that the satellite data
 150 never captures the lowest tides.

151 A satellite tide bias investigation for a coastal area of interest will return an automated report
 152 and plot (Figure 4), adding insightful tide-based context to a coastal EO analysis:

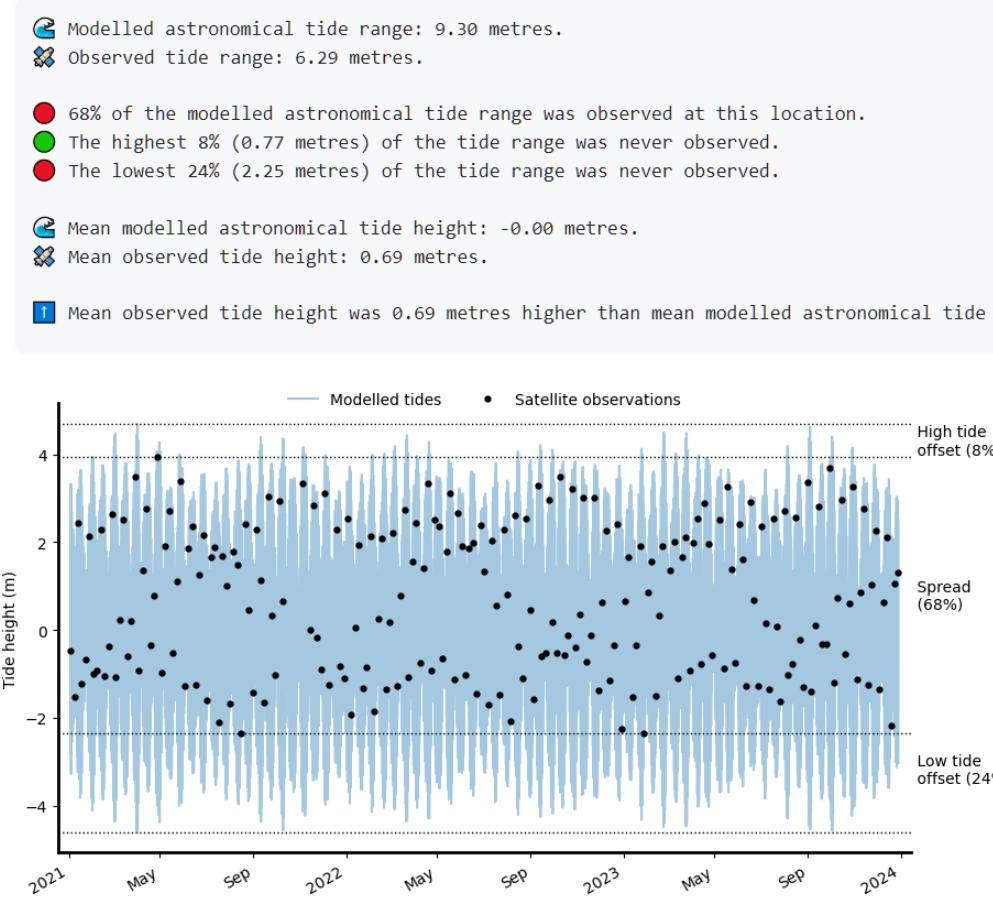


Figure 4: In this example satellite time series, the data captured a biased proportion of the tide range: only observing ~68% of the modelled astronomical tide range, and never observing the lowest 24% of tides. The plot visually demonstrates the relationships between satellite observed tide heights (black dots) and modelled astronomical tide height (blue lines) at this location.

153 Validating modelled tide heights

154 The `eo_tides.validation` module contains tools for validating modelled tides against observed
 155 sea level data. The tide models supported by `eo-tides` can vary significantly in accuracy
 156 across the world's coastlines. Evaluating the accuracy of modelled tides is critical for ensuring
 157 that resulting marine or coastal EO analyses are reliable and useful.

158 Validation functionality in `eo-tides` provides a convenient tool for loading high-quality sea-level
 159 measurements from the GESLA Global Extreme Sea Level Analysis (Haigh et al., 2023) archive
 160 – a global dataset of almost 90,713 years of sea level data from 5,119 records across the world.
 161 The `load_gauge_gesla` function allows GESLA data to be loaded for the same location and
 162 time period as a satellite time series. Differences between modelled and observed tide heights
 163 can then be quantified through the calculation of accuracy statistics that include the Root
 164 Mean Square Error (RMSE), Mean Absolute Error (MAE), R-squared and bias ([Figure 5](#)).

165 Furthermore, different ocean tide models perform differently in different coastal locations.
 166 `eo-tides` allows multiple tide models to be compared against GESLA data simultaneously
 167 ([Figure 5](#)), empowering users to make informed decisions and choose the optimal tide model
 168 that best suits their specific location or application.



Figure 5: An example comparison of modelled tides from multiple global ocean tide models (EOT20, GOT5.5, HAMTIDE11) against observed sea level data from the Broome 62650 GESLA tide gauge, Western Australia.

169 Research projects

170 Early versions of functions provided in `eo-tides` has been used for continental-scale modelling
 171 of the elevation and exposure of Australia's intertidal zone ([Bishop-Taylor et al., 2024](#)), and to
 172 support tide correction for satellite-derived shorelines as part of the `CoastSeg` Python package
 173 ([Fitzpatrick et al., 2024](#)).

174 Acknowledgements

175 Functions from `eo-tides` were originally developed in the Digital Earth Australia Notebooks
 176 and Tools repository ([Krause et al., 2021](#)). The authors would like to thank all DEA Notebooks
 177 contributors and maintainers for their invaluable assistance with code review, feature suggestions
 178 and code edits. This paper is published with the permission of the Chief Executive Officer,
 179 Geoscience Australia.

180 References

- 181 Bishop-Taylor, R., Nanson, R., Sagar, S., & Lymburner, L. (2021). Mapping Australia's
 182 Dynamic Coastline at Mean Sea Level using Three Decades of Landsat Imagery. *Remote
 183 Sensing of Environment*, 267, 112734. <https://doi.org/10.1016/j.rse.2021.112734>
- 184 Bishop-Taylor, R., Phillips, C., Newey, V., & Sagar, S. (2024). *Digital Earth Australia Intertidal*.
 185 Commonwealth of Australia (Geoscience Australia). <https://doi.org/10.26186/149403>
- 186 Bishop-Taylor, R., Sagar, S., Lymburner, L., & Beaman, R. J. (2019). Between the tides:
 187 Modelling the elevation of australia's exposed intertidal zone at continental scale. *Estuarine,
 188 Coastal and Shelf Science*, 223, 115–128. <https://doi.org/10.1016/j.ecss.2019.03.006>
- 189 Carrere, L., Lyard, F., Cancet, M., Allain, D., Dabat, M.-L., Fouchet, E., Sahuc, E., Faugere,
 190 Y., Dibarboire, G., & Picot, N. (2022). A new barotropic tide model for global ocean:
 191 FES2022. *2022 Ocean Surface Topography Science Team Meeting*, 43.
- 192 Eleveld, M. A., Van der Wal, D., & Van Kessel, T. (2014). Estuarine suspended partic-
 193 uulate matter concentrations from sun-synchronous satellite remote sensing: Tidal and
 194 meteorological effects and biases. *Remote Sensing of Environment*, 143, 204–215.
- 195 Fitzpatrick, S., Buscombe, D., Warrick, J. A., Lundine, M. A., & Vos, K. (2024). *CoastSeg*: An
 196 accessible and extendable hub for satellite-derived-shoreline (SDS) detection and mapping.
 197 *Journal of Open Source Software*, 9(99), 6683. <https://doi.org/10.21105/joss.06683>

- 198 Haigh, I. D., Marcos, M., Talke, S. A., Woodworth, P. L., Hunter, J. R., Hague, B. S.,
199 Arns, A., Bradshaw, E., & Thompson, P. (2023). GESLA version 3: A major update to
200 the global higher-frequency sea-level dataset. *Geoscience Data Journal*, 10(3), 293–314.
201 <https://doi.org/https://doi.org/10.1002/gdj3.174>
- 202 Hoyer, S., & Joseph, H. (2017). xarray: N-d labeled arrays and datasets in python. *Journal of*
203 *Open Research Software*, 5(1). <https://doi.org/10.5334/jors.148>
- 204 Krause, C., Dunn, B., Bishop-Taylor, R., Adams, C., Burton, C., Alger, M., Chua, S., Phillips, C.,
205 Newey, V., Kouzoubov, K., Leith, A., Ayers, D., & Hicks, A. (2021). *Digital Earth Australia*
206 *notebooks and tools repository*. <https://github.com/GeoscienceAustralia/dea-notebooks/>;
207 Commonwealth of Australia (Geoscience Australia). <https://doi.org/10.26186/145234>
- 208 McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van
209 der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference*
210 (pp. 56–61). <https://doi.org/10.25080/Majora-92bf1922-00a>
- 211 Murray, N. J., Phinn, S. R., Clemens, R. S., Roelfsema, C. M., & Fuller, R. A. (2012).
212 Continental scale mapping of tidal flats across east asia using the landsat archive. *Remote*
213 *Sensing*, 4(11), 3417–3426.
- 214 Sagar, S., Phillips, C., Bala, B., Roberts, D., Lymburner, L., & Beaman, R. J. (2018).
215 Generating continental scale pixel-based surface reflectance composites in coastal regions
216 with the use of a multi-resolution tidal model. *Remote Sensing*, 10(3), 480. <https://doi.org/10.3390/rs10030480>
- 217 Sagar, S., Roberts, D., Bala, B., & Lymburner, L. (2017). Extracting the intertidal extent and
218 topography of the australian coastline from a 28 year time series of landsat observations.
219 *Remote Sensing of Environment*, 195, 153–169.
- 220 Sutterley, T. C., Alley, K., Brunt, K., Howard, S., Padman, L., & Siegried, M. (2017). *pyTMD:*
221 *Python-based tidal prediction software*. Zenodo. <https://doi.org/10.5281/zenodo.5555395>
- 222 team, odc-geo development. (2024). Opendatacube/odc-geo. In *GitHub repository*. GitHub.
<https://github.com/opendatacube/odc-geo>
- 223 team, pandas development. (2020). *Pandas-dev/pandas: pandas* (latest). Zenodo. <https://doi.org/10.5281/zenodo.3509134>
- 224 Turner, I. L., Harley, M. D., Almar, R., & Bergsma, E. W. J. (2021). Satellite optical imagery
225 in Coastal Engineering. *Coastal Engineering*, 167, 103919. <https://doi.org/10.1016/j.coastaleng.2021.103919>
- 226 Vitousek, S., Buscombe, D., Vos, K., Barnard, P. L., Ritchie, A. C., & Warrick, J. A. (2023).
227 The future of coastal monitoring through satellite remote sensing. *Cambridge Prisms:*
228 *Coastal Futures*, 1, e10. <https://doi.org/10.1017/cft.2022.4>
- 229 Vos, K., Splinter, K. D., Harley, M. D., Simmons, J. A., & Turner, I. L. (2019). CoastSat: A
230 Google Earth Engine-enabled Python toolkit to extract shorelines from publicly available
231 satellite imagery. *Environmental Modelling & Software*, 122, 104528. <https://doi.org/10.1016/j.envsoft.2019.104528>
- 232
- 233
- 234
- 235
- 236