

Friday, May 16, Practical Session V: Publishing (and Connecting) Your Network Analysis

Ryan Horne



Advanced Research Computing, UCLA
ryan.matthew.horne@gmail.com



Final Countdown



Agenda

- Measurements we missed
- Neo4j, Linked Open Data, and the software / scholarly ecosystem
 - Plus a thing or two from me
 - It will be fun, honest!



Components

- Natural breaks for connected portions of a graph
- Connected component of a graph
 - Every node in the subset has a path to each other

Giant Component

Informal definition: Connected component that contains a significant fraction of all the nodes

- Game of Thrones: People who want Joffrey dead. Not **everybody**, but close!
- Most networks only have one giant component



How to Find the Giant Component in Neo4j GDS

First, run weakly connected components (WCC)

WCC is an algorithm that identifies clusters (also called components) in a directed graph, where every node in a cluster is connected to every other node if you ignore the direction of relationships.

- WCC is used to:
 - Identify islands or clusters of connectivity in a graph.
 - Detect isolated groups of nodes.
 - Understand fragmentation of a network.



WCC

```
1 CALL gds.wcc.stream('movies-graph-undirected')
2 YIELD nodeId, componentId
3 RETURN *,gds.util.asNode(nodeId).name AS name,componentId
4 ORDER BY componentId
5 LIMIT 50;
```

Write to node

```
1 CALL gds.wcc.write('movies-graph', {  
2   writeProperty: 'componentId'  
3 });
```

Get the size of each component

```
1 MATCH (n)
2 WITH n.componentId AS component, count(*) AS size
3 RETURN component, size
4 ORDER BY size DESC;
```

Assuming 0 is the largest

```
1 MATCH (n)
2 WHERE n.componentId = 0
3 RETURN n
```

Merger of Giant Components

- Only one connection merges giant components into one
 - In history: Sudden, often catastrophic change
 - Think of 1492 C.E.
 - Disease
 - Political change
 - Previous contacts were not sustained
- Issue of time

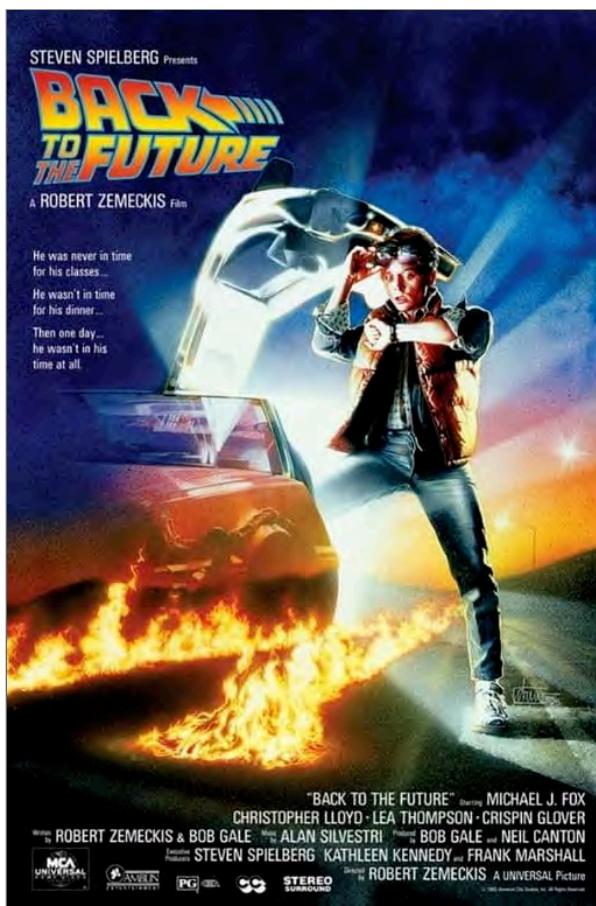
Where to Go From Here

- What can we *do* with these networks?
- How is LOD used?
- More information about all of this?



Think About the Future

- What do you want to do with network analysis?
- Or Digital Humanities?
- Do you want to:
 - Use tech *as is* to provide insights?
 - Learn enough to be dangerous, but remain mostly within traditional Byzantine studies?
 - Jump in all the way and learn the nuts and bolts of the tech?
- Think about where and what you want to do



Well, Maybe Don't Think Too Much About the Future



It is ok to just have a pretty picture!

- Maybe you just want a visual of the structure of your network
- Perhaps to see how concepts are related, flow, etc

Part of a Digital Ecosystem

- Think of Neo4j as a tool to store networks and nodes, not tables
 - Although we can export the data as tables!
- Return to Linked Open Data
- How can we get the data, links, and ontologies into Neo4j?



Neo4j is a Database

- We looked at some stats, graphs, and exports
- This does not have to be done *in* Neo4j
- We can use it as just a database and export or transfer the data to other programs

Neo4j is a Database

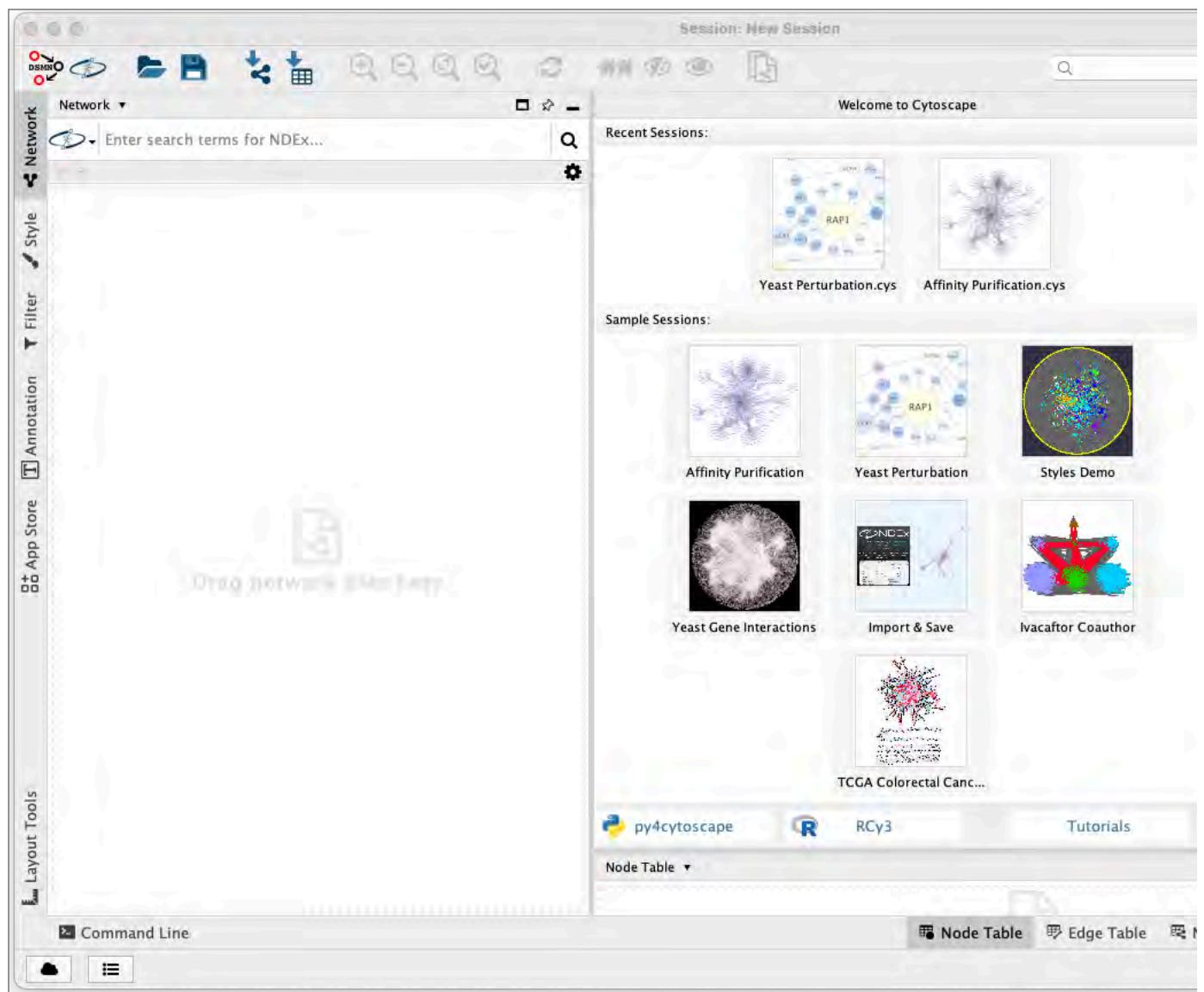
- We looked at some stats, graphs, and exports
- This does not have to be done *in* Neo4j
- We can use it as just a database and export or transfer the data to other programs

A Quick Overview

Export Target	Advantages	Disadvantages
Cytoscape	<ul style="list-style-type: none">- Visual graph exploration & styling- Easy Neo4j plugin- Layout & analysis tools	<ul style="list-style-type: none">- Not suited for very large graphs- Plugin may have compatibility limits
Gephi	<ul style="list-style-type: none">- High-performance large graph visualization- Dynamic filtering & layout- Graph statistics	<ul style="list-style-type: none">- Requires export to GraphML/CSV first- Less user-friendly than Cytoscape
CSV (Excel, Pandas)	<ul style="list-style-type: none">- Easy to share and manipulate- Good for tables and summaries- Compatible with many tools	<ul style="list-style-type: none">- Loses graph structure (e.g., path traversal)- Manual relationship parsing

Export Target	Advantages	Disadvantages
GraphML / GEXF	<ul style="list-style-type: none"> - Preserves node-link structure - Works with Gephi, Cytoscape, etc. - Good for offline storage 	<ul style="list-style-type: none"> - Requires APOC plugin - Limited metadata if not customized
JSON	<ul style="list-style-type: none"> - Ideal for web apps/APIs - Compatible with D3.js, Python, JavaScript - Retains flexible structure 	<ul style="list-style-type: none"> - More complex to transform manually - Large JSON can be hard to debug
Python (via Py2neo, Neo4j Driver)	<ul style="list-style-type: none"> - Fully programmatic control - Integration with data science tools - Automate queries and transformations 	<ul style="list-style-type: none"> - Requires coding skills - Slower than native Cypher for bulk exports
RDF (via Neosemantics)	<ul style="list-style-type: none"> - Semantic web & ontology-friendly - Interoperable with triplestores (e.g., Virtuoso) 	<ul style="list-style-type: none"> - Requires Neosemantics plugin - Complexity of RDF model vs. property graphs

Neo4j to Cytoscape



A Plugin

- <https://apps.cytoscape.org/apps/cytoscapeneo4jplugin>
- It is broken :(
- Can still export a network as nodes and edges tables
 - Also how to bring it into Gephi
 - Also as a .csv
- What are some of the possible issues here?



CSV to RDF, etc

- There are utilities / functions that can do this
- Also can be done by hand....

Next Idea

- You can use OpenRefine
- <https://openrefine.org/>
- Free, opensource tool that has reconciliation capabilities!



Another Plugin for Doing This Directly

- There is the neosemantics plugin
- <https://neo4j.com/labs/neosemantics/>
- It is not compatible with the current version of neo4j!!!



Just export a csv

```
1 MATCH (m:Movie)
2 RETURN m.title AS title, m.released AS released, m.tagline
3
4
5 MATCH (p:Person)-[r:ACTED_IN]->(m:Movie)
6 RETURN p.name AS actor, m.title AS movie, r.roles AS roles
7
8
```

Go A Little Deeper

- Why convert when you can just call the data?
- You saw R in colab
- You can use Python to tie together the loose ends



Python

```
1  from neo4j import GraphDatabase
2
3
4 # Connection parameters
5 uri = "bolt://localhost:7687" # or your remote bolt URI
6 username = "neo4j"
7 password = "your_password"
8
9 # Create a driver instance
10 driver = GraphDatabase.driver(uri, auth=(username, password))
11
12 # Function to run a simple query
13 def get_movies():
14     with driver.session() as session:
15         result = session.run("MATCH (m:Movie) RETURN m.title")
16         for record in result:
17             print(record["title"])
18
19 # Run function
```

Python

- Requires some programming knowledge
- Many powerful libraries
- You can use Python to combine GIS and network data in the same area



Example: Networks of Coins

- Data sources
 - Pleiades = spatial data for cities
 - American Numismatic Society (ANS): MANTIS, NOMISMA, Coins, spatial information, data structures, some hoard data
 - Museums using ANS data structures
 - Ancient World Mapping Center = spatial data (roads, etc)



id	hoard_id	title	ex_start_date	ex_end_date	buried	b_start_date	b_en
1	ch1,1	Asia Minor		1974	-550-545		-550.0
2	ch1,2	Hellespont		1969	c. 550 B.C.?		-550.0
3	ch1,3	Asia Minor		1935/1940	c. 525 B.C.		-525.0
4	ch1,4	South Anatolia		1971	c. 500 B.C.		-500.0
5	ch1,5	Aegina		1973	c. 500 B.C.?		-500.0
...
2365	ch10,247	Unknown findspot		1993			NaN
4843	igch2254	Cibali, Catana district		1907			NaN
4845	igch2256	Bronte		1906			NaN
4846	igch2257	Lipari		1920			NaN
4847	igch2258	Termini Imerese environs		1958			NaN

[Home](#)[Die Study / Jupyter Notebooks](#)[GitHub Site](#)[Coin C](#)

Double Click to open coin information

Show **10** ▾ entries

Coin ID	Mint	Type ID	Weight	S
1	550908	1676	16.53	
100	550908	1676		
1000	550756	1648	16.91	
1001	550908	1689	16.72	
1002	550908	1680	16.82	
1003	550908	1689	16.54	
1004	550908	1686	16.66	
1005	550908	1678	16.29	



1007	550908	1687	15.5
1008	550908	1680	16.78



File Edit View Insert Cell Kernel Widgets Help

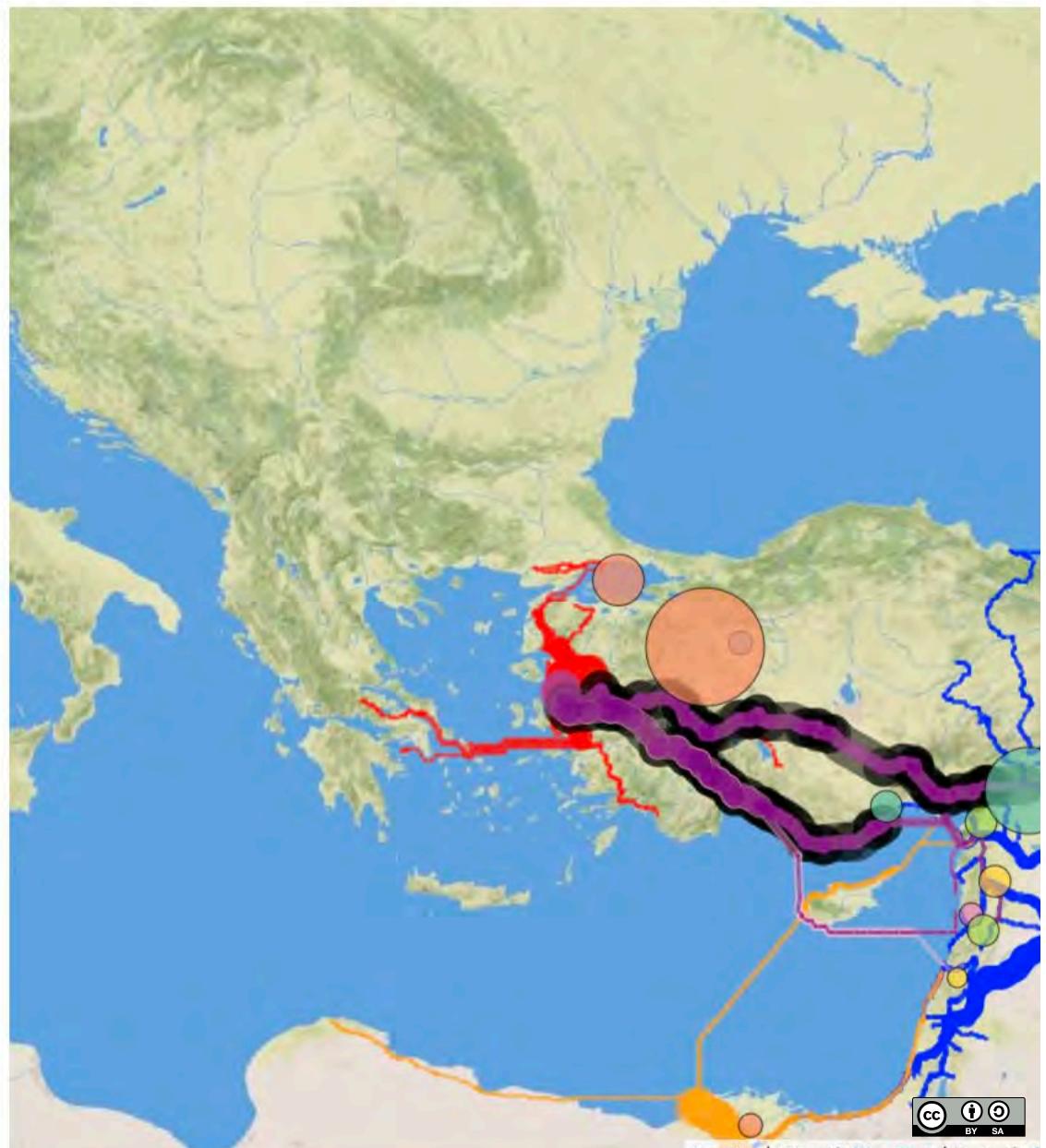


```
addNetworkRoutesFromFile(temnos_routes['url'], temnos_routes['  
addNetworkRoutesFromFile(myrina_routes['url'], myrina_routes['  
addNetworkRoutesFromFile(kyme_routes['url'], kyme_routes['feat
```

```
addFeatureGroupToMap(temnoshardsdf, hoardsfg, 30, 5, 'degree'  
m.add_child(folium.map.LayerControl())
```

```
m
```

Out[33]:



The extensive Temnian minting during the time, especially the very large production Attalids to draw upon in support of their own political projects. One attractive possit soldiers for hire would presumably want some form of wealth that was transferable clearly shown in an earlier mutiny of mercenaries at Philateria and Attalea against the ending the disturbances. Euromos swore to provide his rioting troops with fixed price

Tyre

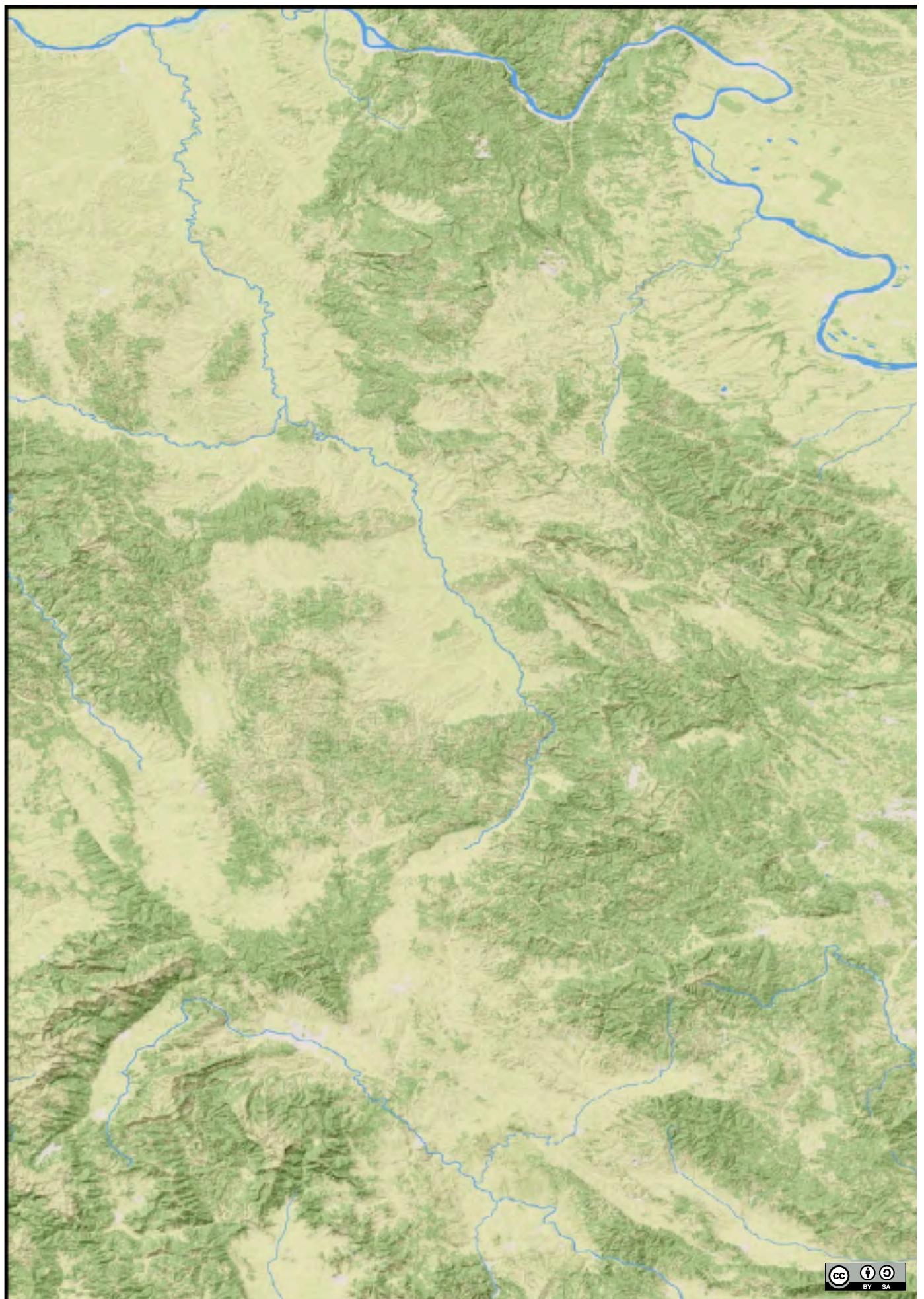
Sidon

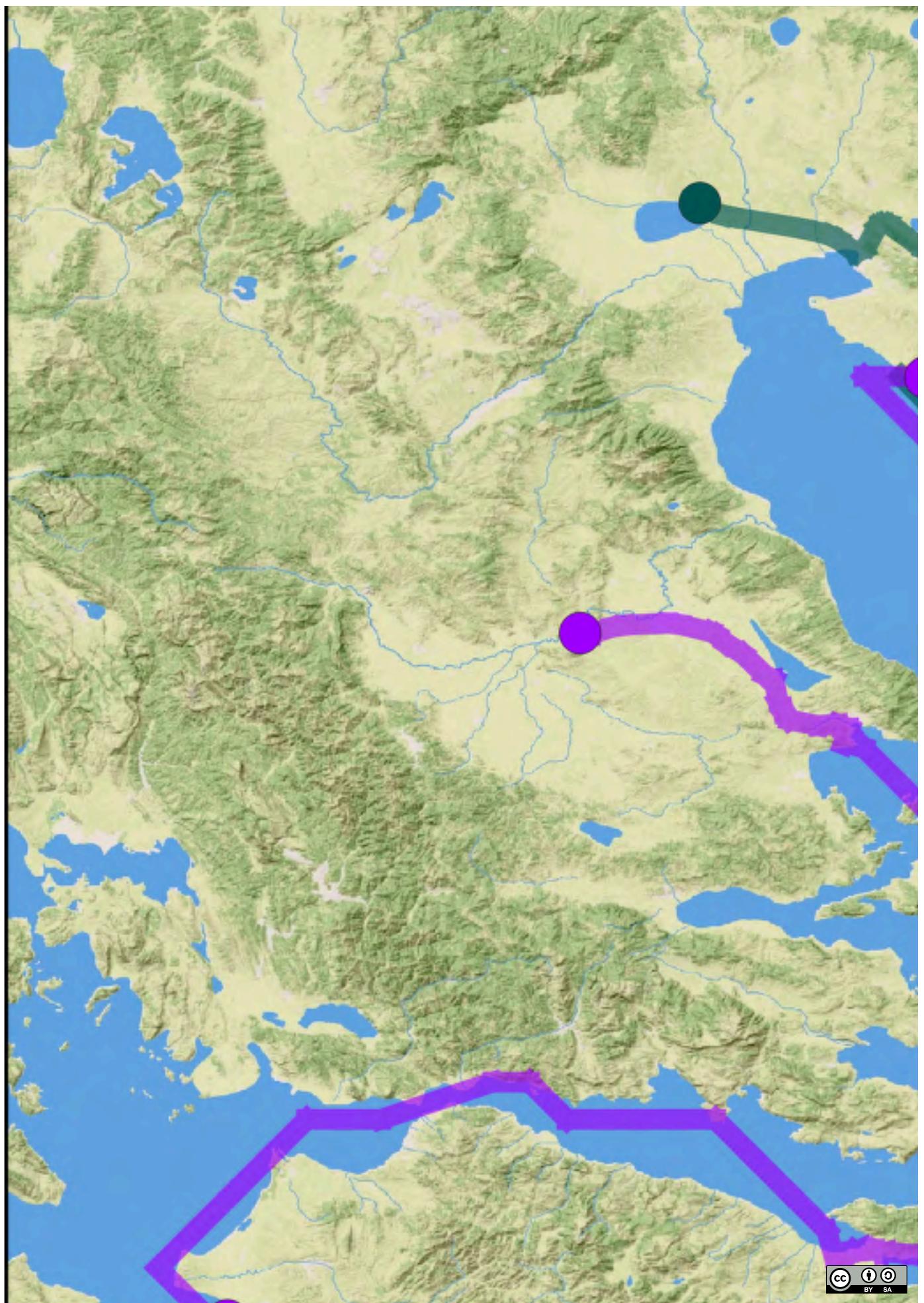
Gaza

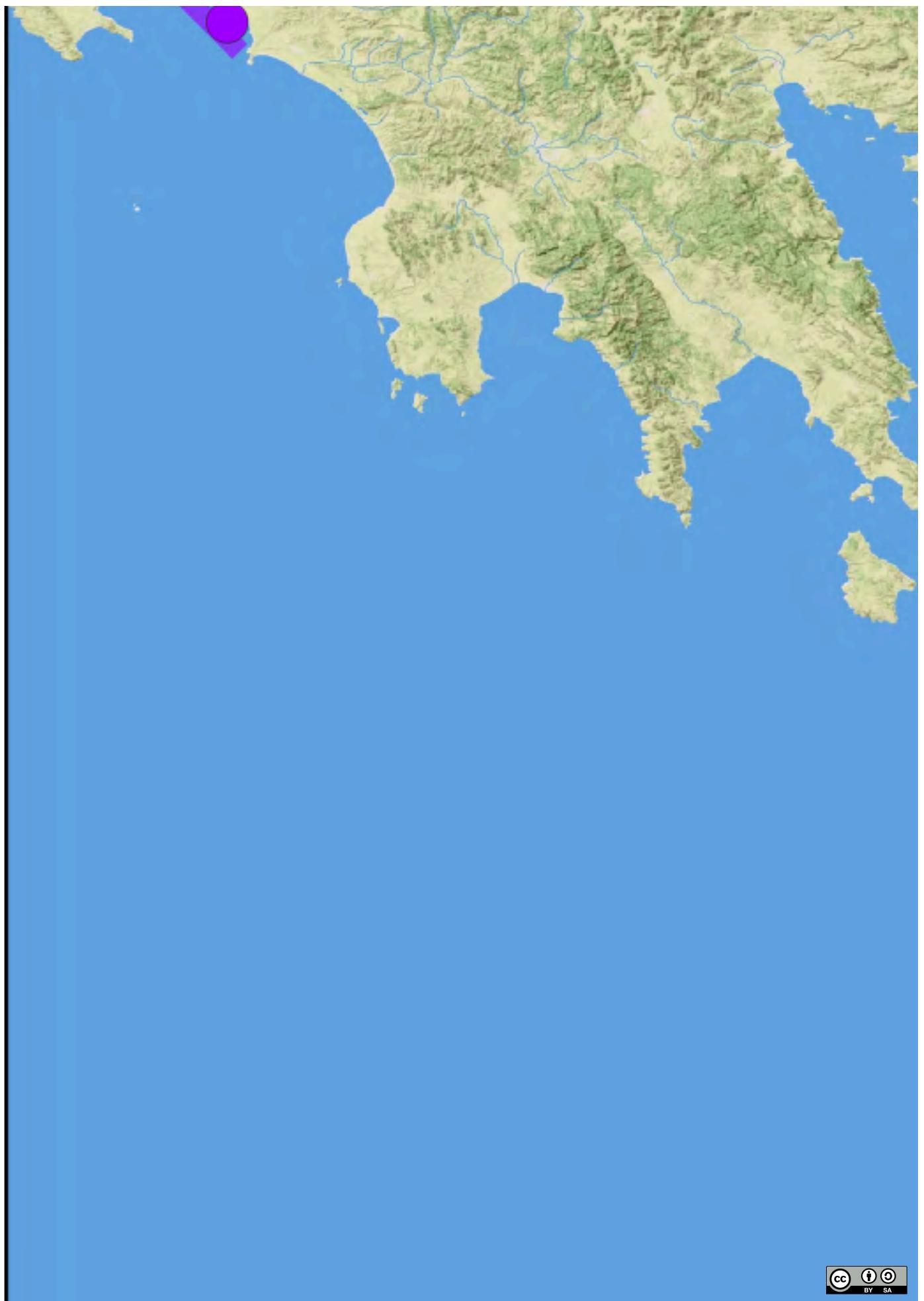
Jerusa

Ptolemaic Dynasty
Ep

Alexar

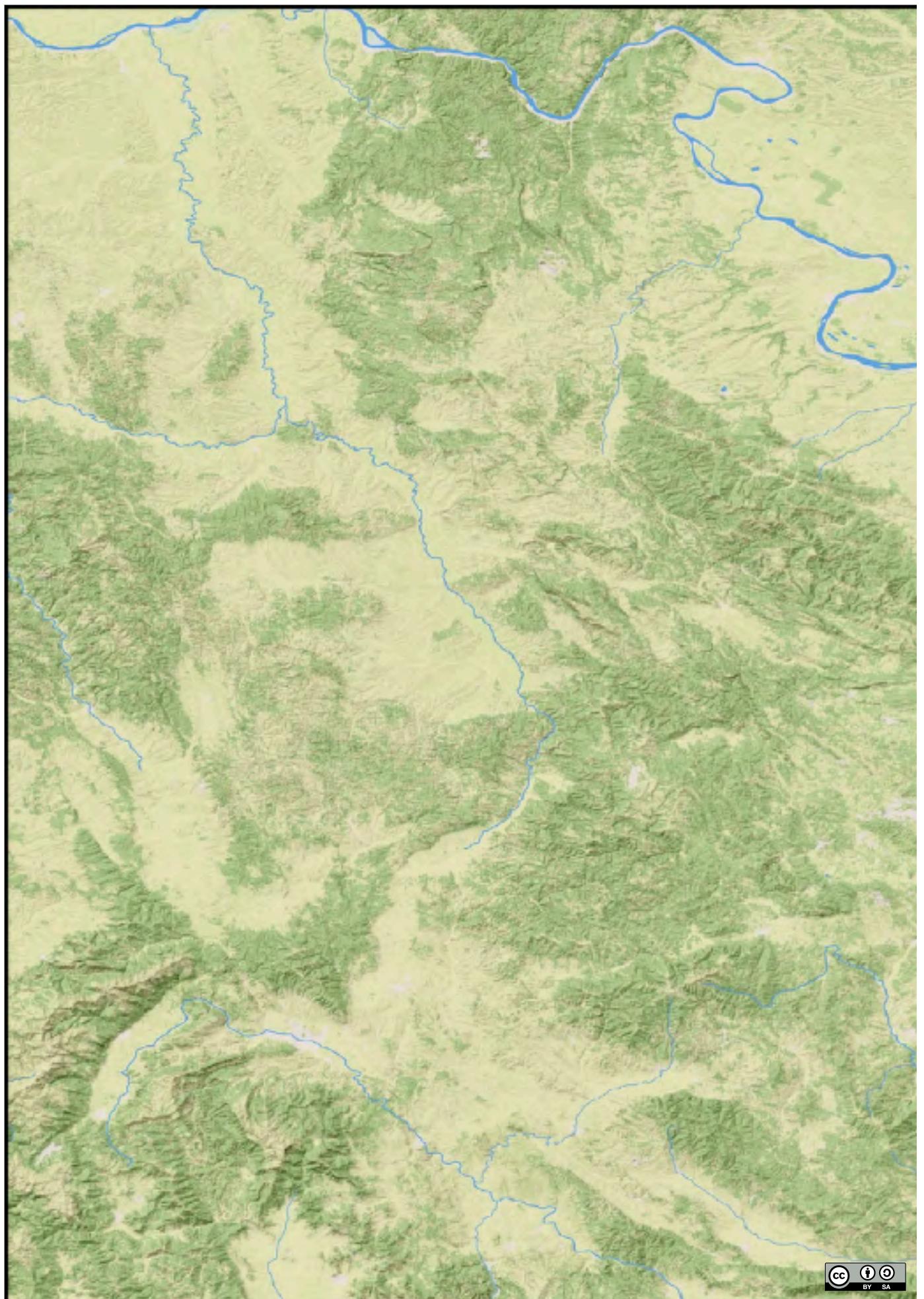


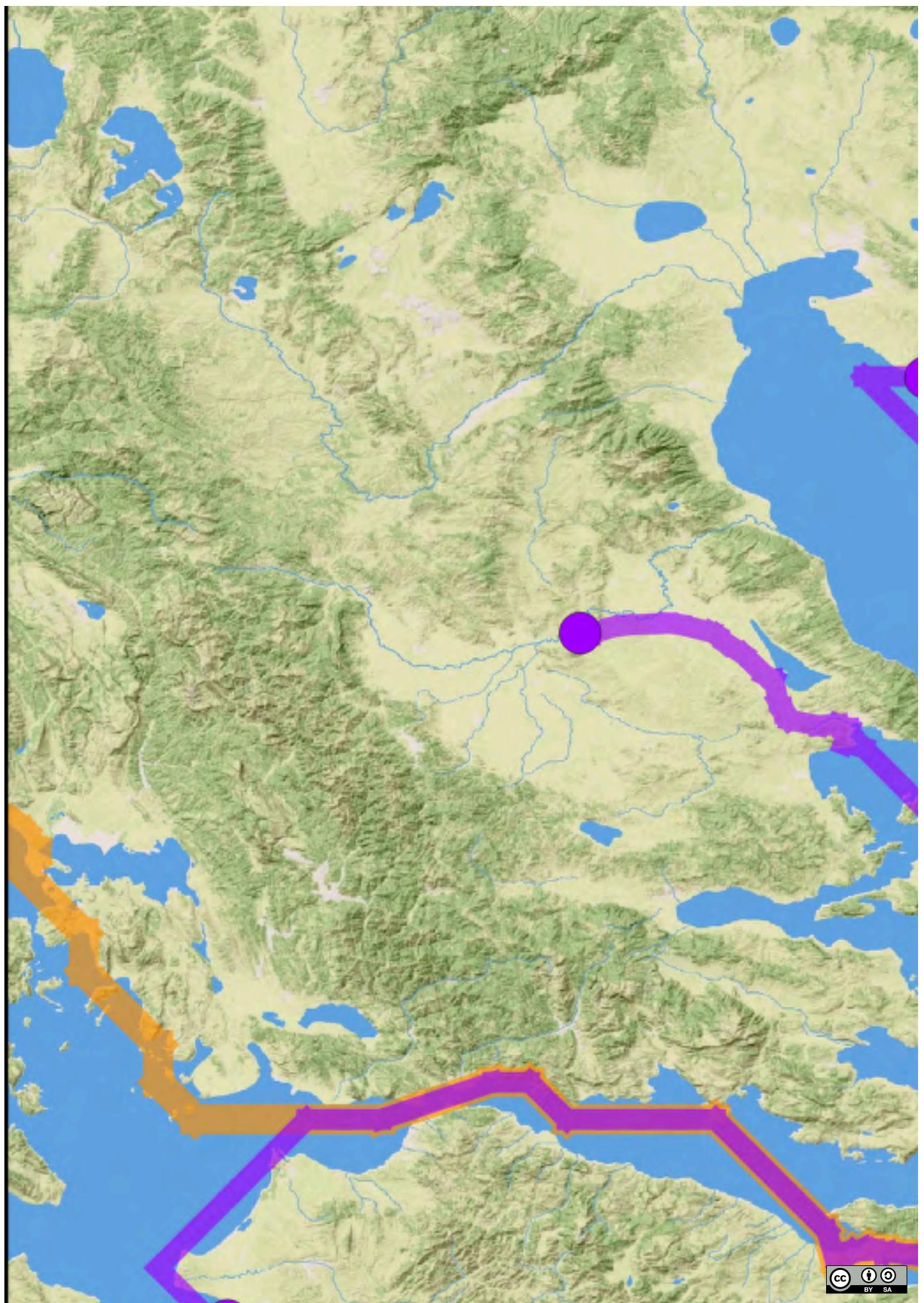


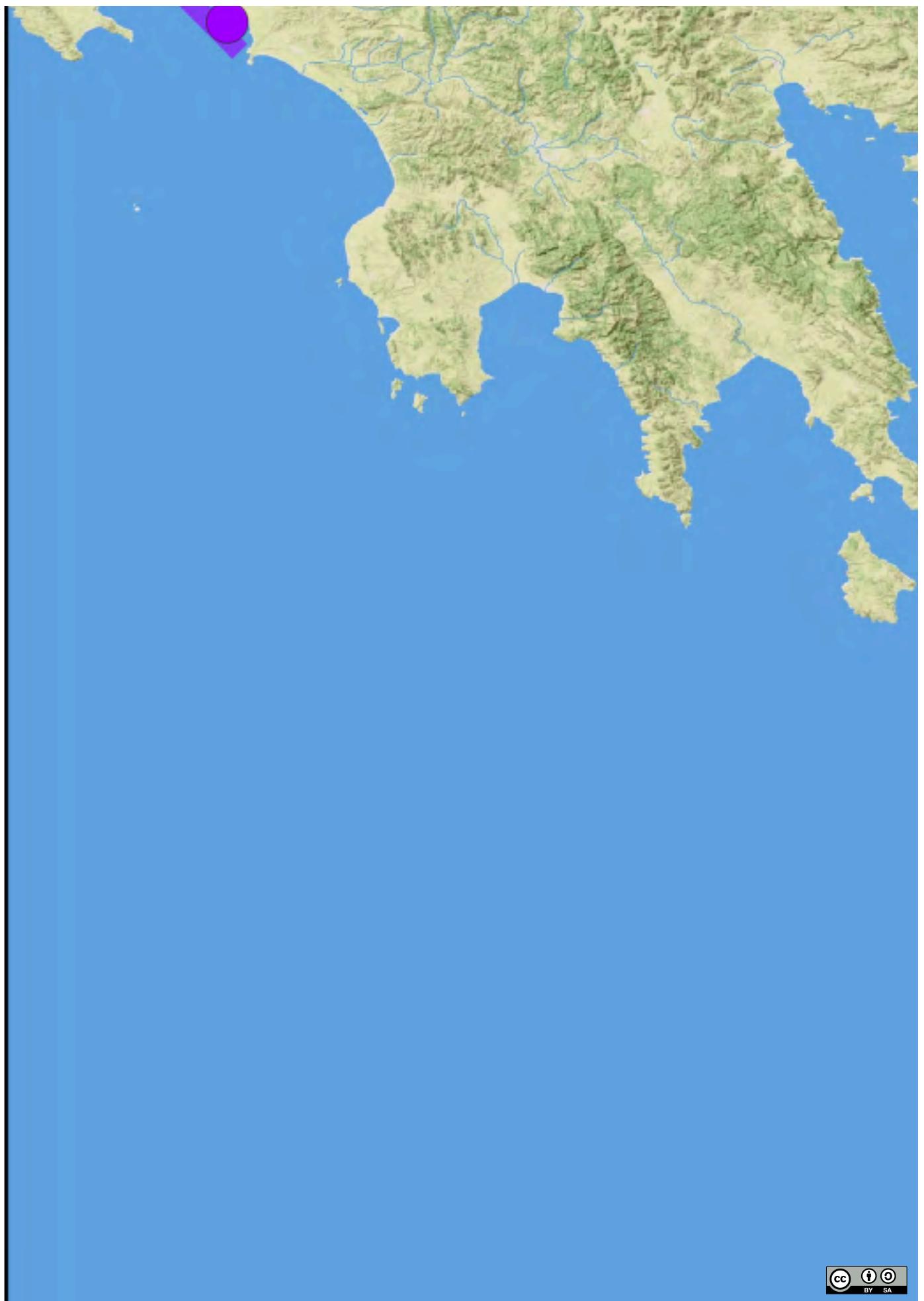


0

200

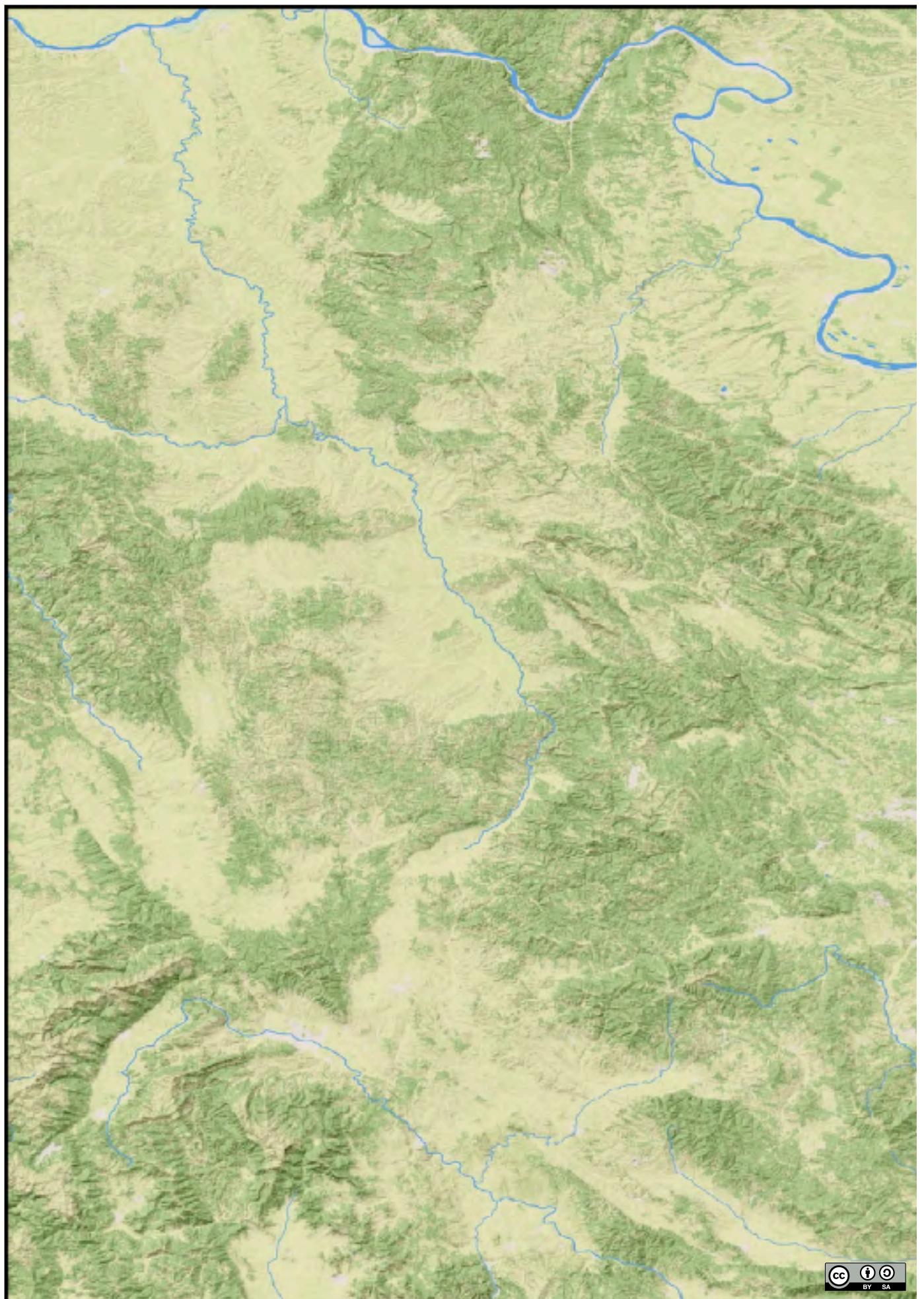


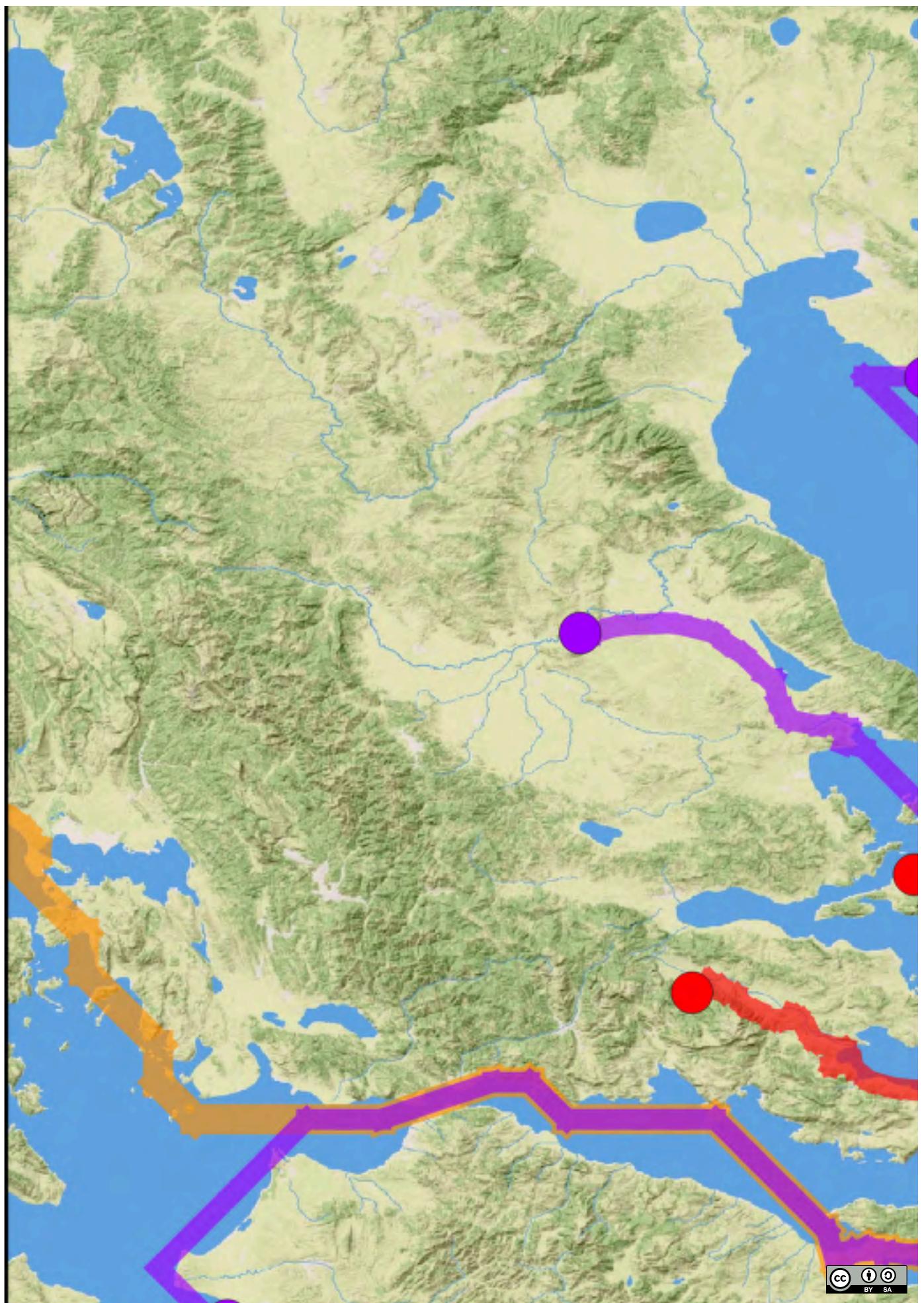


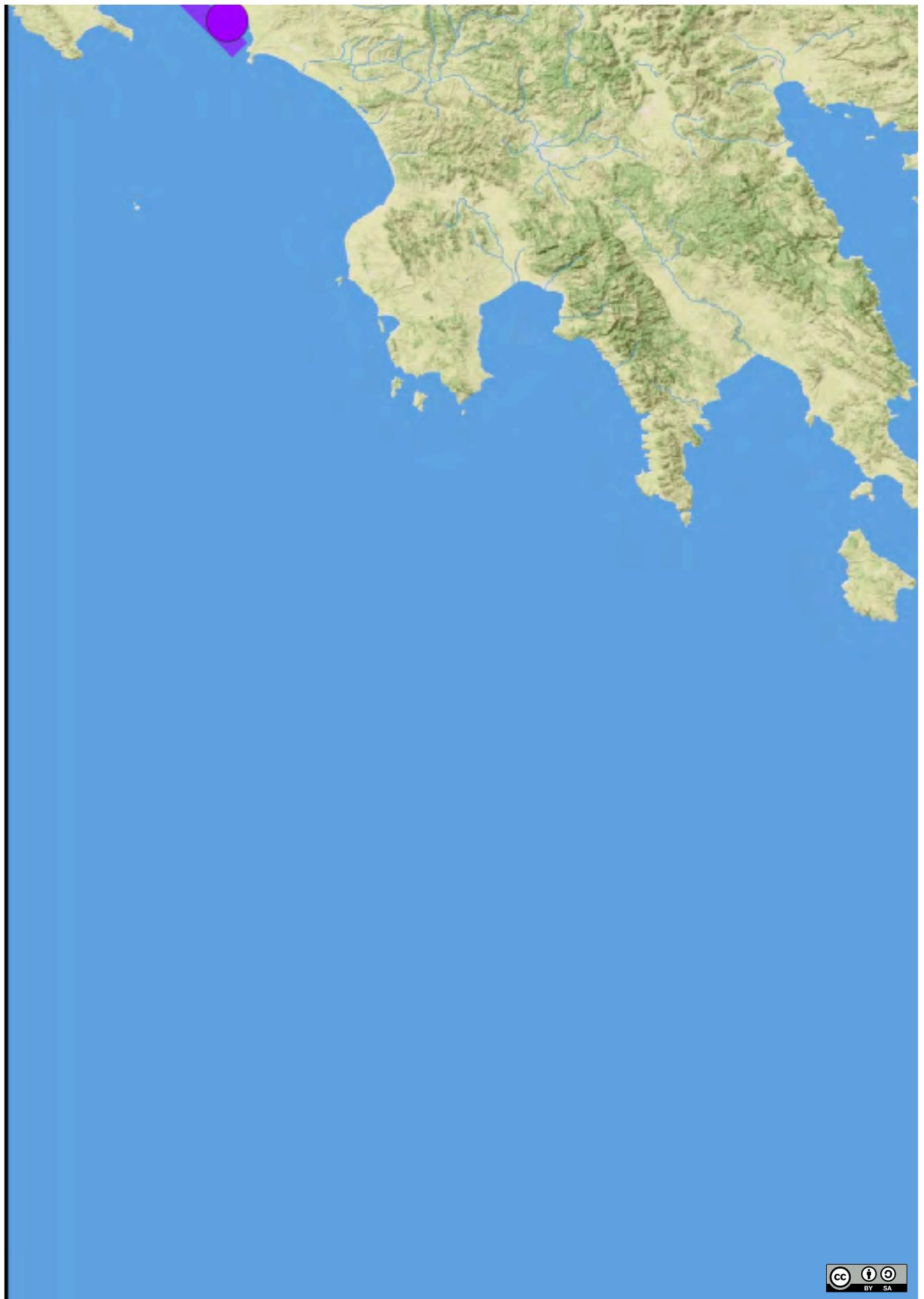


0

200

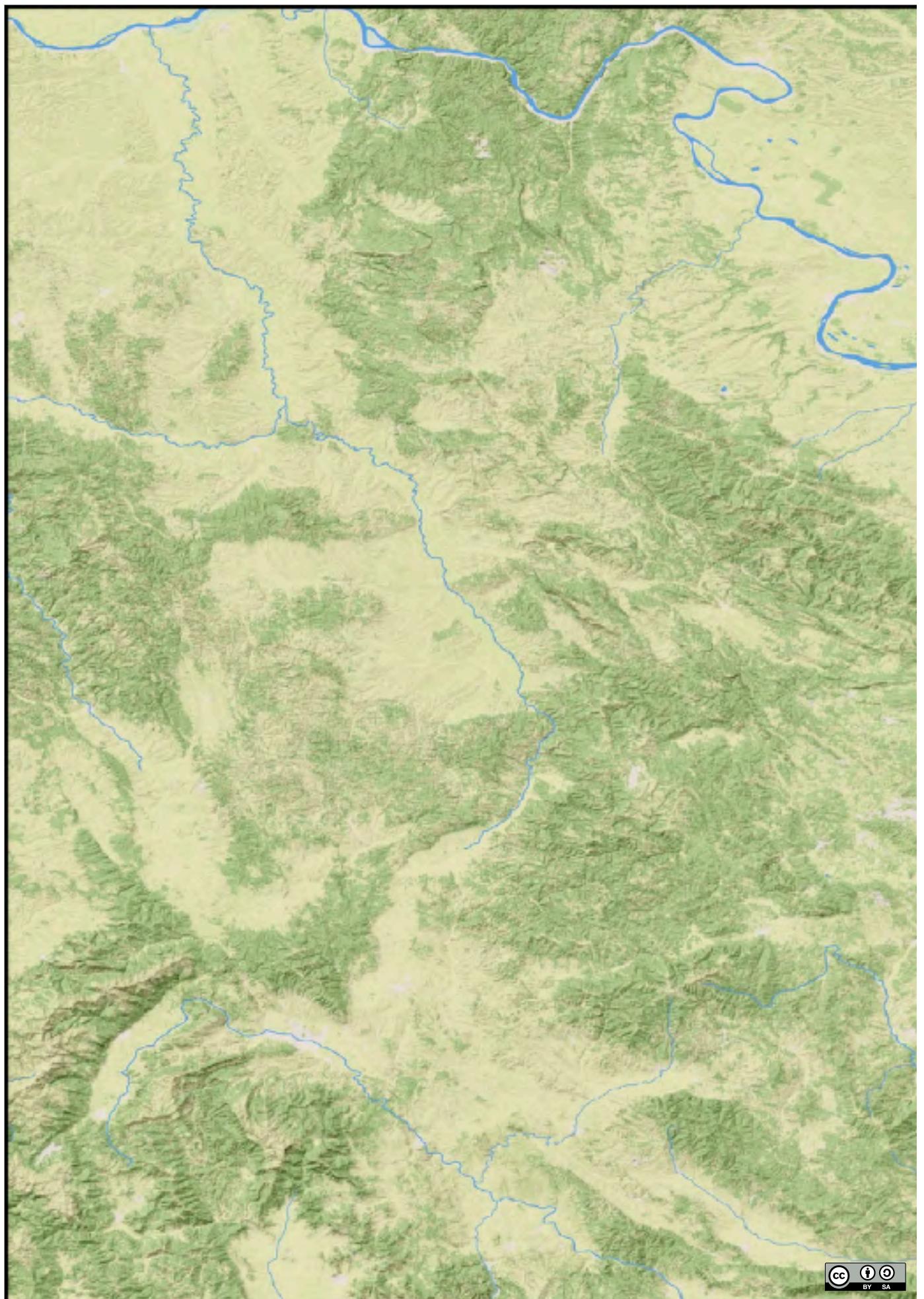


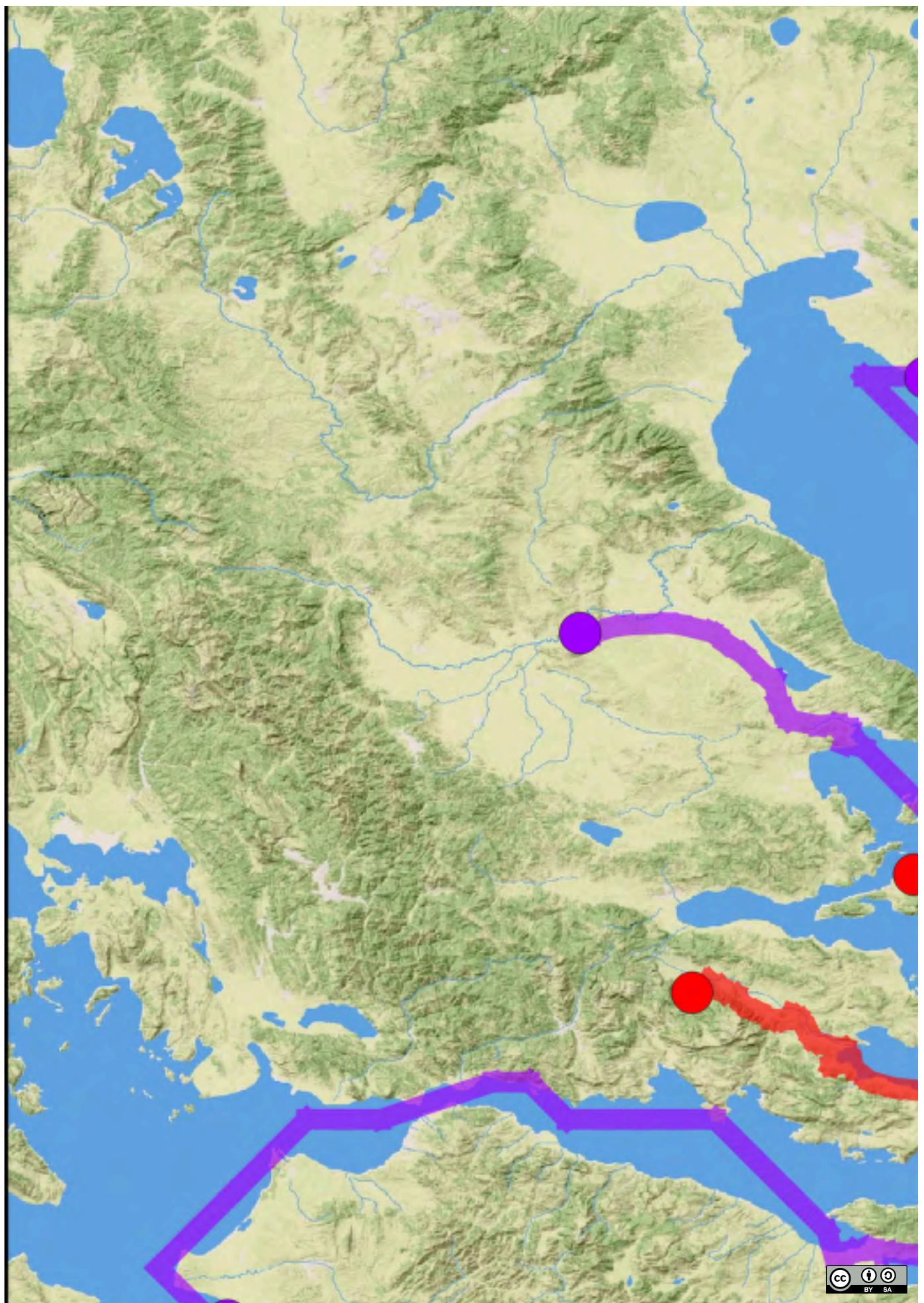


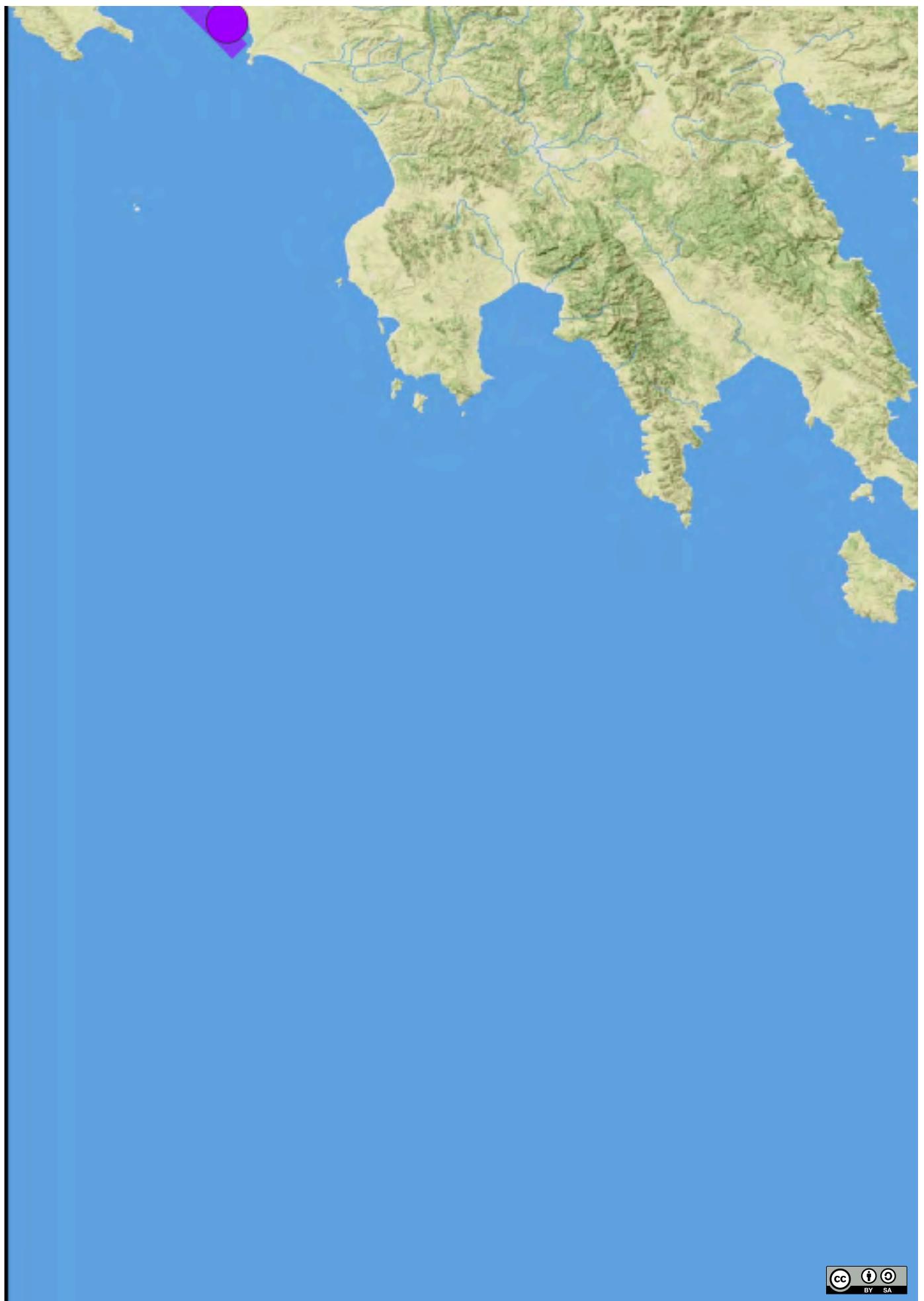


0

200

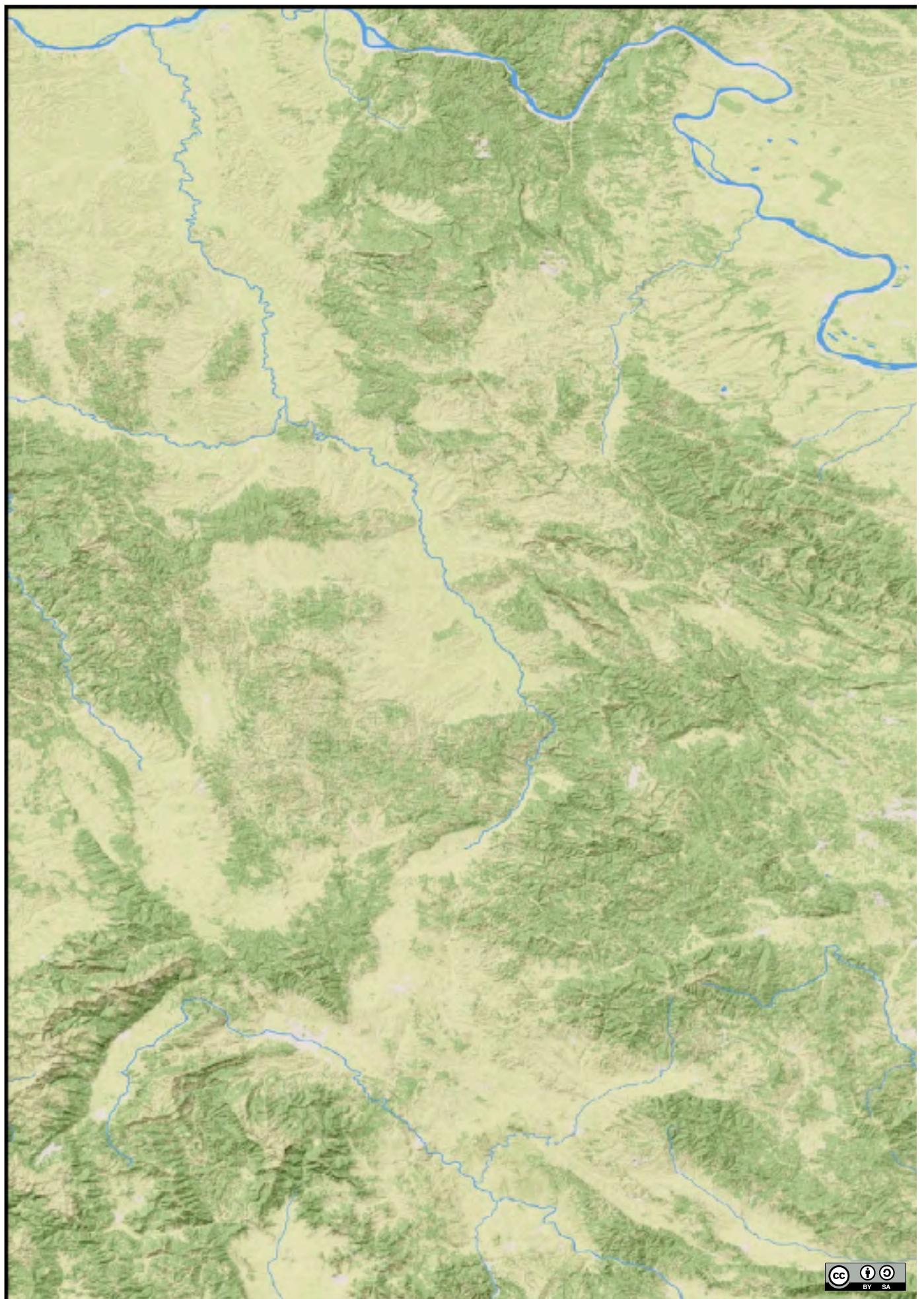




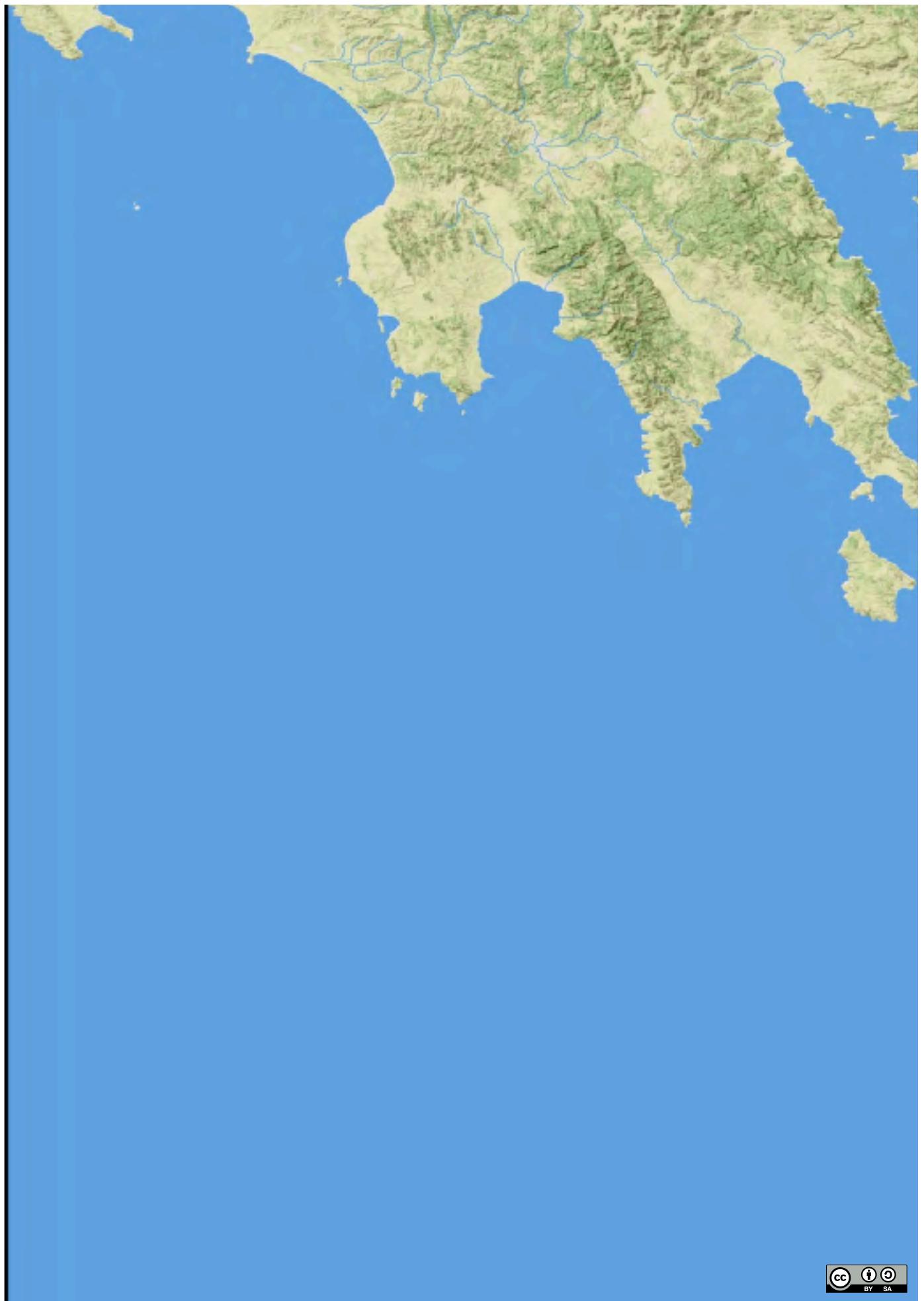


0

200







0

200



