Supplementary Information

Supplementary Table 1. All surveyed settlements and their distance from modeled August microenvironments.

Site Code	Period	Description	Distance from Modeled August Microenvironment (m)
KKT08	Unknown	Semi-subterranean structure	76.609697
KKT09	Unknown	Semi-subterranean structure	215.131741
KKT10	Unknown	Semi-subterranean structure	201.967965
KKT12	Multi-period	Multi-complex site	168.57826
KKT14	Multi period (Bronze Age and Turkic)	Multi-complex site with large burial ground with many graves, possibly undisturbed	48.055783
KKT15	Bronze/Iron Age	Multi-complex consisting of kurgans, burials and structures	215.947044
KKT19	Multi-period	Multi-complex site consisting of semi-circular enclosures	174.267563
KKT23-26	Multi-period	4 structures, semi-subterranean square form	117.502748
KKT27	Multi-period	2 structures	88.17544
KKT28	Multi-period	Structure	39.566334
KKT32	Multi-period	Circular house with surrounding fence	31.042814
KKT33	Multi-period	Multi-period settlement.	66.61917
KKT34	Multi-period	Multi-complex site of houses next to cemetery	0
KKT36	Turkic or later	Semi-circular multi-terraced stone structure	0
KKT37	Multi-period	Multi-complex site	9.922755
KKT39	Multi-period	Multi-complex site consisting of at least 7 burials	139.586128
KKT40	Iron Age	Multi-complex site consisting of at least 13 kurgans	119.528954
KKT43	Multi-period	Multi-complex site of burials, settlements, shrines	5.172643
KKT48	Multi-period	Multiple structures of a settlement	94.617246

KKT50	Multi-period	Multi-complex site	37.986764
SMT01	Bronze and Iron Age	Multi-complex site consisting of at least 7 kurgans and stone structures from Bronze Age and later	142.627721
SMT02	Bronze and Iron Age	Multi-complex site consisting of different kurgans and stone fence BA burial	122.044359
SMT08	Medieval	Caravanserai with a double wall enclosure	2086.025653
SMT09	Multi-period	Multi-complex site consisting of settlement and burials robbed around 5 years ago.	193.809811
SMT14	Medieval	Caravansarai or fort with a double wall enclosure	92.835308
SMT15	Multi-period	Multiple stone structures over a hill from SMT14 and alongside a jasper stone source with some stones showing potential evidence of human modification.	219.278033
SMT16	Medieval	At least 12 subterranean structures with heavy overgrowth located next to a spring and next to SMT14	11.220368

Supplementary Table 2. Statistics across 3 indices for the entire study area, and for modeled microenvironments (using the August extent).

Microenvironment			
May			
	CIRE	NDVI	Thermal
mean	1.541839	0.598347	300.71525
min	-0.99749	-0.9959	292.06123
max	5.361526	0.875776	322.69694
STD	0.593803	0.132741	5.142115
August			
mean	1.930881	0.699758	305.99966
min	0.562528	0.39531	299.93635
max	9.226837	0.974478	317.69638
STD	0.655054	0.094979	3.140003

Landscape			
May			
mean	0.726222	0.374778	308.16459
min	-0.99914	-0.99834	292.06123
max	5.361526	0.876285	325.11348
STD	0.344391	0.130921	5.309164
August			
mean	0.582071	0.315547	312.5912
min	-0.99917	-0.99858	299.93635
max	9.08052	0.94	319.00206
STD	0.458314	0.115288	3.132602

Supplementary Text 1: GEE script

```
// Minimum distance classification to identify microenvironments
// Add an 8-band PlanetScope image into the imports tab, and name it "img"
// Add the following feature classes as assets (training data):
// "fc1" - microenvironments
// "fc2" - terrestrial non-microenvironment
// "fc3" - water

// If you are using a >10GB image (such as our 35,000 sqkm arid microenvironment image), upload it as a cloud-optimized geotiff to a google cloud account
// Then, un-comment the following two lines:
// var img = ee.Image.loadGeoTIFF("gs://Put your bucket name here/Raster_name.tif");
// Map.addLayer(img);

//If you are using an image of less than 10GB, ignore the above and proceed
//Assign planet bands (assumed names b1..b8)
var bands = ['b1','b2','b3','b4','b5','b6','b7','b8'];

// Here you can add class properties if you already have them named, otherwise leave it as "null"
var existingClassProperty = null;

// Class labels / IDs / colors (3 classes)
var classNames = ['Microenvironment','Terrestrial','Water'];
var classIds   = [1, 2, 3];
// Optional, assign colors below, such that microenvironments = green terrestrial non-microenvironments = red, and water = blue
var palette    = ['#4daf4a','#e41a1c','#377eb8'];

// Determine sattelite image horizontal sampling size/pixel size and set these values appropriately
var sampleScale = 3;
```

```
var exportScale = 3;

// Center the map on your image
var roi = img.geometry();

// Select only the bands used for classification
img = img.select(bands);

// Stamp classes onto your "fc1, fc2, fc3" vector shapefiles
function stampClass(fc, clsId) {
  return existingClassProperty
    ? fc
    : fc.map(function (f) { return f.set('class', clsId); });
}
fc1 = stampClass(fc1, classIds[0]);
fc2 = stampClass(fc2, classIds[1]);
fc3 = stampClass(fc3, classIds[2]);
var allFc = fc1.merge(fc2).merge(fc3);
var classProp = existingClassProperty || 'class';

// Assign a variable for your training samples
var samples = img.sampleRegions({
  collection: allFc,
  properties: [classProp],
  scale: sampleScale,
  geometries: false
});

// Begin creating your pixel classifier below
```

```
// OPTIONAL: to enable these optional classifier parameters below (if you wish to tweak the classifier), un-comment the following three lines
// var METRIC = 'mahalanobis';    // options: 'euclidean', 'cosine', 'mahalanobis', 'manhattan'
// var K= 1;

// Make the classifier below
var classifier = ee.Classifier
// If you rnabled the tunable factors above, insert (METRIC, K) at the end of the next line
  .minimumDistance()
  .train({
    features: samples,
    classProperty: classProp,
    inputProperties: bands
  });

var classified = img.classify(classifier);

Map.centerObject(roi, 11);

// To display your PlanetScope image in RGB
Map.addLayer(
  img,
  {bands: ['b4','b3','b2'], min: 0, max: 3000},
  'Input (RGB)'
);

Map.addLayer(
  classified,
  {min: classIds[0], max: classIds[classIds.length - 1], palette: palette},
  'Minimum Distance Classification'
```

```
);

// Export your classified image to Google Drive, be sure to pick an appropriate name for each line of the script
Export.image.toDrive({
  image: classified.toInt(),
  description: 'name',
  folder: 'name',
  fileNamePrefix: 'predfix',
  region: roi,
  scale: exportScale,
  maxPixels: 1e13
});

// Below is a seperate GEE script window to generate spectral indices
// Load PlanetScope  8-band image, and name it "PLANET"
// Apply band names
var bands = {
  blue: 'b2',
  green: 'b3',
  red: 'b6',
  redEdge: 'b7',
  nir: 'b8'
};

// Calculate NDVI
var ndvi = PLANET.normalizedDifference([bands.nir, bands.red]).rename('NDVI');
Map.addLayer(ndvi, {min: -1, max: 1, palette: ['blue', 'white', 'green']}, 'NDVI');

// Calculate CIRE
```

```
var ciRedEdge = PLANET.expression(
  '(NIR / RE) - 1', {
    'NIR': PLANET.select(bands.nir),
    'RE': PLANET.select(bands.redEdge)
}).rename('CI_red_edge');

Map.addLayer(ciRedEdge, {min: 0, max: 3, palette: ['white', 'yellow', 'green']}, 'CI Red Edge');

// Center the map on the image
Map.centerObject(PLANET, 12);

//Export to Drive, choose geometry as image footprint
var exportRegion = PLANET.geometry();

// 3m per pixel, the same resolution as planet imagery
var exportScale = 3;

// Export NDVI to drive
Export.image.toDrive({
  image: ndvi,
  description: 'PLANET_NDVI_toDrive',
  fileNamePrefix: 'PLANET_NDVI',
  region: exportRegion,
  scale: exportScale,
  maxPixels: 1e13
});

// Export CIRE to Drive
Export.image.toDrive({
  image: ciRedEdge,
```

```
  description: 'PLANET_CIredEdge_toDrive',
  fileNamePrefix: 'PLANET_CIredEdge',
  region: exportRegion,
  scale: exportScale,
  maxPixels: 1e13
});



// The following script can be used separately to generate land surface temperature (LST) in a study area of your choosing
// Set your AOI in terms of a mimumum/maximum longitude followed by minimum/maximum lattitude
var region = (typeof region === 'undefined')
  ? ee.Geometry.Rectangle([62.0, 51.5, 62.6, 52.0])  //
  : region;
// Apply Landsat L2 and C2 scale factors before anything else
function applyScaleFactors(img) {
  var sr = img.select('SR_B.*').multiply(0.0000275).add(-0.2);
  var stK = img.select('ST_B10').multiply(0.00341802).add(149.0);
  return img.addBands(sr, null, true)
            .addBands(stK.rename('ST_B10'), null, true);
}

function addLST(img) {
  var lstK = img.select('ST_B10').rename('LST'); // already Kelvin
  return img.addBands([lstK]);
}

// Find monthly LST
function processMonth(year, month) {
```

```
  var startDate = ee.Date.fromYMD(year, month, 1);
  var endDate   = startDate.advance(1, 'month');  // exclusive end

  // Produce a merged L8+L9 L2 collection
  var l8 = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2');
  var l9 = ee.ImageCollection('LANDSAT/LC09/C02/T1_L2');

  var col = l8.merge(l9)
    .filterBounds(region)
    .filterDate(startDate, endDate)
    .filter(ee.Filter.lte('CLOUD_COVER', 80)) // looser to avoid empty months
    .filter(ee.Filter.listContains('system:band_names', 'ST_B10'))
    .map(applyScaleFactors)
    .map(addLST);

  // If empty, return a masked LST so .select('LST') is always safe.
  var out = ee.Image(ee.Algorithms.If(
    col.size().gt(0),
    col.median().clip(region),
    ee.Image.constant(0).rename('LST').updateMask(ee.Image(0)).clip(region)
  ));

  return out;
}

// Pick your year and month below
var year = 2022, month = 5;
var lstImg = processMonth(year, month);

// Ensure that there are actually scenes present
```

```
ee.ImageCollection('LANDSAT/LC08/C02/T1_L2').merge(ee.ImageCollection('LANDSAT/LC09/C02/T1_L2'))
  .filterBounds(region)
  .filterDate(ee.Date.fromYMD(year, month, 1), ee.Date.fromYMD(year, month, 1).advance(1, 'month'))
  .size()
  .evaluate(function(n){
    if (n > 0) {
      Map.addLayer(lstImg.select('LST'), {
        min: 285.35, max: 314.85, // Kelvin equivalents
        palette: [
          '040274','040281','0502a3','0502b8','0502ce','0502e6',
          '0602ff','235cb1','307ef3','269db1','30c8e2','32d3ef',
          '3be285','3ff38f','86e26f','3ae237','b5e22e','d6e21f',
          'fff705','ffd611','ffb613','ff8b13','ff6e08','ff500d',
          'ff0000','de0101','c21301','a71001','911003'
        ]
      }, 'LST_' + year, true);
    } else {
      print('No L8/L9 scenes available in this time and AOI!');
    }
  });

// center map
Map.centerObject(region, 10);

// Prepare export
var out = lstImg.select('LST')
  .unmask(-9999)
  .rename('LST');
```

```
var exportScale = 30;
var exportCrs   = 'EPSG:4326';

// Export to Drive, be sure to pick appropriate names.
Export.image.toDrive({
  image: out,
  description: 'name',
  fileNamePrefix: 'prefix',
  region: region,
  scale: exportScale,
  crs: exportCrs,
  maxPixels: 1e13,
  fileFormat: 'GeoTIFF',
  formatOptions: {
    cloudOptimized: true,
  }
});
```