**Supplemental 1**

The following Google Earth Engine script was used to determine the percent ice cover for all lakes in the analysis.

// This script calculated lake ice monthly statistics based on MODIS daily 500m snow cover product

// This script was written to process lake ice data for large lakes (defined as area >= 64km^2) and small

// lakes (area < 64 km^2).

// create monthly groups

var startDate = ee.Date.fromYMD(2000, 3, 1);

var list = ee.List.sequence(0, 261, 1);

var monthly = ee.FeatureCollection(list.map(function(l) {

return(ee.Feature(null, {

beginDate: startDate.advance(l, 'month').format('YYYY-MM-dd'),

endDate: startDate.advance(ee.Number(l).add(1), 'month').format('YYYY-MM-dd'),

year: startDate.advance(l, 'month').get('year'),

month: startDate.advance(l, 'month').get('month'),

}))}));

// prepare MODIS snow cover raster as two bands:

// 1. original fractional snow cover for each pixel (0-100);

// 2. binary snow cover using threshold of 15%

var modisTerraSnow = modisTerra.select(['NDSI\_Snow\_Cover']);

var modisTerraSnowFraction = modisTerraSnow.map(function(i) {

return(ee.Image(i

.addBands(i.gte(15).multiply(100).rename(['snowCoverBinary']))

.addBands(i.unmask(ee.Image.constant(-1), true).eq(-1).rename(['missingData']))

.updateMask(i.gte(0).and(i.lte(100)))));

});

print(modisTerraSnowFraction.first());

// Functions to calculate monthly mean, median, min, max ice cover, as well as the number

// of valid daily observations per month.

// Each statistics was calculated for both the NDSI\_Snow\_Cover and snowCoverBinary bands

// month with greater than 20% of missing daily data were not included in the output.

// Two functions are identical except that the scale at which to carry out the spatial aggregation

// was given differently (lines 46 and 82) to use a coarser resolution for large lakes to

// speed up the calculations.

var calcIceFractionOneLargeLake = function(f) {

var dailyIceStatsOneLake = modisTerraSnowFraction

.map(function(i) {

var dailyIce = i.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: f.geometry(),

scale: f.getNumber('Lake\_area').divide(200).multiply(1000000).sqrt().round(),

bestEffort: false,

tileScale: 1});

return(ee.Feature(null, dailyIce).copyProperties(i, ['system:time\_start']));

});

var columnReducer = ee.Reducer.minMax()

.combine(ee.Reducer.count(), '', true)

.combine(ee.Reducer.mean(), '', true)

.combine(ee.Reducer.median(), '', true);

var combinedReducer = columnReducer.forEach(['NDSI\_Snow\_Cover', 'snowCoverBinary']);

var monthlyIceStatsOneLake = monthly

.map(function(g) {

var monthlyIceStats = dailyIceStatsOneLake

.filterDate(g.getString('beginDate'), g.getString('endDate'))

.filterMetadata('missingData', 'not\_greater\_than', 0.2)

.reduceColumns(combinedReducer, ['NDSI\_Snow\_Cover', 'snowCoverBinary']);

return(ee.Feature(null, monthlyIceStats)

.copyProperties(f, ['Hylak\_id'])

.copyProperties(g, ['year', 'month']));

});

return(monthlyIceStatsOneLake);

};

var calcIceFractionOneSmallLake = function(f) {

var dailyIceStatsOneLake = modisTerraSnowFraction

.map(function(i) {

var dailyIce = i.reduceRegion({

reducer: ee.Reducer.mean(),

geometry: f.geometry(),

scale: 500,

bestEffort: false,

tileScale: 1});

return(ee.Feature(null, dailyIce).copyProperties(i, ['system:time\_start']));

});

var columnReducer = ee.Reducer.minMax()

.combine(ee.Reducer.count(), '', true)

.combine(ee.Reducer.mean(), '', true)

.combine(ee.Reducer.median(), '', true);

var combinedReducer = columnReducer.forEach(['NDSI\_Snow\_Cover', 'snowCoverBinary']);

var monthlyIceStatsOneLake = monthly

.map(function(g) {

var monthlyIceStats = dailyIceStatsOneLake

.filterDate(g.getString('beginDate'), g.getString('endDate'))

.filterMetadata('missingData', 'not\_greater\_than', 0.2)

.reduceColumns(combinedReducer, ['NDSI\_Snow\_Cover', 'snowCoverBinary']);

return(ee.Feature(null, monthlyIceStats)

.copyProperties(f, ['Hylak\_id'])

.copyProperties(g, ['year', 'month']));

});

return(monthlyIceStatsOneLake);

};

// The code below scale up the analysis to each of the 2607 lakes in HydroLakes dataset with surface area larger or equal than 64 km^2

// Due to the large size of these lakes, spatial aggregation of ice cover for these lakes were done using a lower resolution

// (than the 500m resolution of the MODIS image) adaptive to the size of the lake to make the calculation possible.

// Ice data for these lakes were calculated with multiple tasks submitted to GEE server, with each one contains a few hundreds of

// lakes.

var hl = ee.FeatureCollection('users/michaelfrederickmeyer/HydroLAKES\_whole');

var hlLarge = hl.filterMetadata('Lake\_area', 'not\_less\_than', 64).map(function(f) {return(f.buffer(90, 500))});

print(hlLarge.size(), 'No. of large lakes');

var outputLarge = hlLarge.limit(500).map(calcIceFractionOneLargeLake).flatten();

print(outputLarge.first(), 'example output for large lakes');

Export.table.toDrive(outputLarge, 'large\_lake\_', '', 'large\_lake\_', 'csv',

['Hylak\_id', 'year', 'month', 'NDSI\_Snow\_Cover\_min', 'NDSI\_Snow\_Cover\_max', 'NDSI\_Snow\_Cover\_mean', 'NDSI\_Snow\_Cover\_median', 'snowCoverBinary\_min', 'snowCoverBinary\_max', 'snowCoverBinary\_mean', 'snowCoverBinary\_median', 'NDSI\_Snow\_Cover\_count']);

// The code below scale up the analysis to each of the 1425081 lakes in HydroLakes dataset with surface area smaller than 64 km^2

// For these smaller lakes, spatial aggregation of ice cover were done using the highest spatial resolution available (500 meters).

// Ice data for these lakes were calculated with multiple tasks submitted to GEE server.

var hl = ee.FeatureCollection('users/michaelfrederickmeyer/HydroLAKES\_whole');

var hlSmall = hl.filterMetadata('Lake\_area', 'less\_than', 64).map(function(f) {return(f.buffer(90, 500))});

print(hlSmall.size(), 'No. of small lakes');

var outputSmall = hlSmall.limit(500).map(calcIceFractionOneSmallLake).flatten();

print(outputSmall.first(), 'example output for small lakes')

Export.table.toDrive(outputSmall, 'small\_lake\_', '', 'small\_lake\_', 'csv',

['Hylak\_id', 'year', 'month', 'NDSI\_Snow\_Cover\_min', 'NDSI\_Snow\_Cover\_max', 'NDSI\_Snow\_Cover\_mean', 'NDSI\_Snow\_Cover\_median', 'snowCoverBinary\_min', 'snowCoverBinary\_max', 'snowCoverBinary\_mean', 'snowCoverBinary\_median', 'NDSI\_Snow\_Cover\_count']);