# Supplementary Material for Development and assessment of automated forest road projection methods using performance metrics relevant for wildlife

Journal: Landscape Ecology

Authors: Josie Hughes, Sarah Endicott, David Lapins, Kyle Lochhead, and Gregory Paradis

Corresponding author: Josie Hughes; josie.hughes@ec.gc.ca (mailto:josie.hughes@ec.gc.ca); Landscape Science and Technology Division, National Wildlife Research Center, Environment and Climate Change Canada, Ottawa, ON, Canada

# 1 Grade penalty and simple cost methods for setting edge weights

The grade penalty method for calculating edge weights is a simplified version of the approach taken by Anderson and Nelson (2004) that does not distinguish between adverse and favourable grades. We use a modified Digital Elevation Model (DEM) as the input to the edge weight calculation that allows barriers other than slope to be included without incurring additional computation and memory costs. Pixels where roads cannot be built are assigned a value of NA, and negative values indicate costs associated with other features (e.g. stream crossings). Existing roads are assigned a value of 0. All other values in the weight raster are assumed to be elevation and grade is calculated from differences between adjacent locations. Default construction cost values are taken from the BC Interior Appraisal Manual: the base cost of building a road is $16 178 with an additional cost of $504 for every 1% grade and a limit of 20% grade. Edges with greater than 20% grade are assigned either NA or an arbitrarily high value. A value of 65 000 is used in our projections to avoid errors when a landing cannot be accessed without crossing an NA cell. Note that grade penalties are not applied adjacent to existing roads, nor do they apply when connecting to a node with a negative value. In the latter case, the edge weight is set to the highest absolute cost. Weights are adjusted for the horizontal distance between nodes to penalize longer diagonal road segments.

```
gradePenaltyFn <- function(x1, x2, hdistance, baseCost = 16178, limit = 20,
                           penalty = 504, limitWeight = NA){
  #For grade calculation, x1, x2 and hdistance (i.e. horizontal distance between nodes)
  #must have the same units.

  # Don't calculate grade penalty cost if one of the nodes is a barrier.
  cond <- pmin(x1, x2) >= 0
  cond[is.na(cond)] <- F

  # Apply grade penalty if both nodes have elevation values.
  # If one node is a road use base cost.
  grade <- 100 * abs(x1 - x2) * cond / hdistance #This is % (dimensionless)

  slp <- baseCost + grade * penalty * (pmin(x1, x2) > 0) #This is cost per km.
  slp[grade > limit] <- limitWeight

  # If both 0 this is an existing road link. Otherwise it is a barrier.
  slp[!cond] <- abs(pmin(x1, x2))[!cond]

  # Multiply by hdistance to penalize diagonals.
  # Note that multiplying all edge weights by a constant does not change algorithm behaviour,
  # so we ignore the unit of hdistance here.
  slp = slp*hdistance

  return(slp)
}
```

In the simple cost method, we first calculate a slope raster $S$, then calculate an input cost raster $X$ that depends on slope $S$:

$$X = \begin{cases} 0 \text{ if existing road} \\ 65000 \text{ if lake} \\ 16178 + 504S \text{ otherwise} \end{cases}$$

The weight of an edge between two nodes is the mean value of the input cost raster at each of those nodes, adjusted for the euclidean distance between the nodes:

```
simpleCostFn <- function(x1,x2,hdistance){
  # Multiply by hdistance to penalize diagonals.
  # Note that multiplying all edge weights by a constant does not change algorithm behaviour,
  # so we ignore the unit of hdistance here.
  return(hdistance*(x1+x2)/2)
}
```

# 2 Simple cost results

The simple cost method for setting edge weights generally performed worse than the grade penalty and Hardy methods (compare Figures 2 and S1, and Figures 5 and S2).

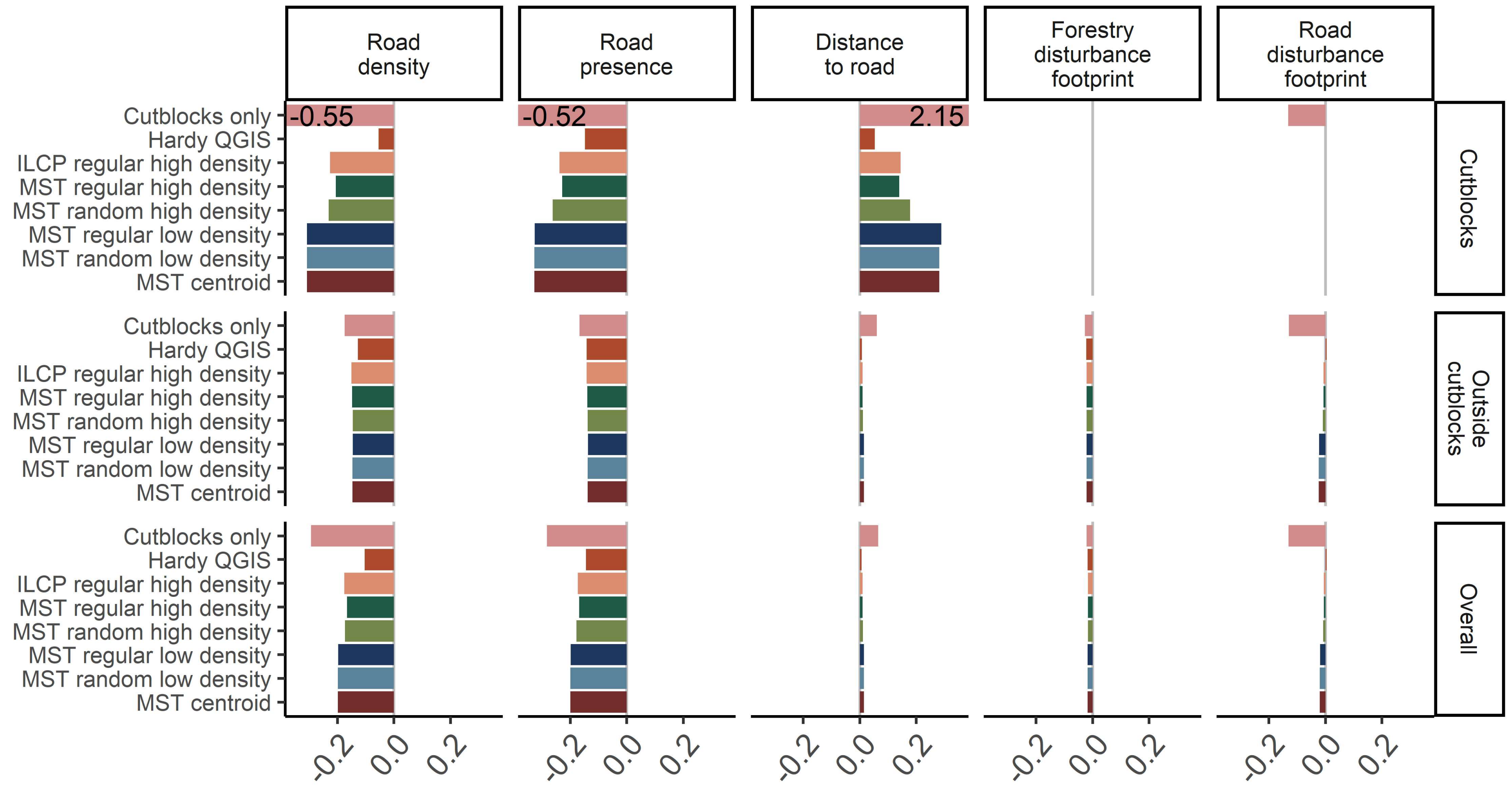

Figure S1: Proportional difference of projected mean from observed mean road metrics using different simple cost projection method variants within cutblocks, outside cutblocks, and overall (1 ha cost raster). Values greater than zero indicate overprojection and negative values indicate underprojection. Cutblocks not accessible via the observed road network were excluded from analysis. Performance varies among metrics (columns), projection method variants (colours), and within and outside cutblocks.

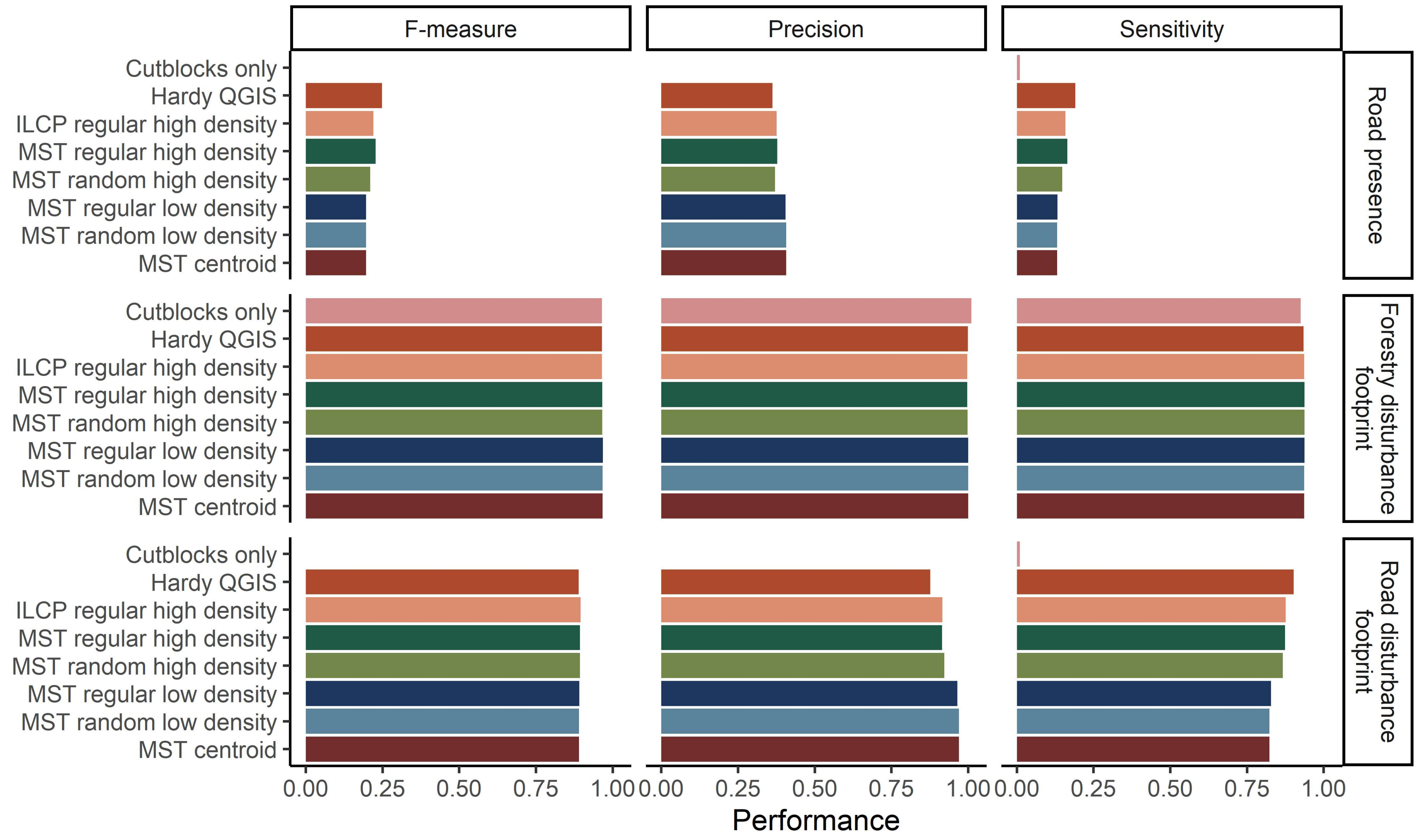

Figure S2: Spatial variation of in performance among simple cost projection method variants (1 ha cost raster). F-measure is the harmonic mean of precision and sensitivity, precision is the proportion of predicted pixels that were observed and sensitivity is the proportion of observed pixels that were correctly predicted. Values closer to one indicate better performance. Cutblocks not accessible via the observed road network were excluded from analysis.

# 3 All cutblocks results

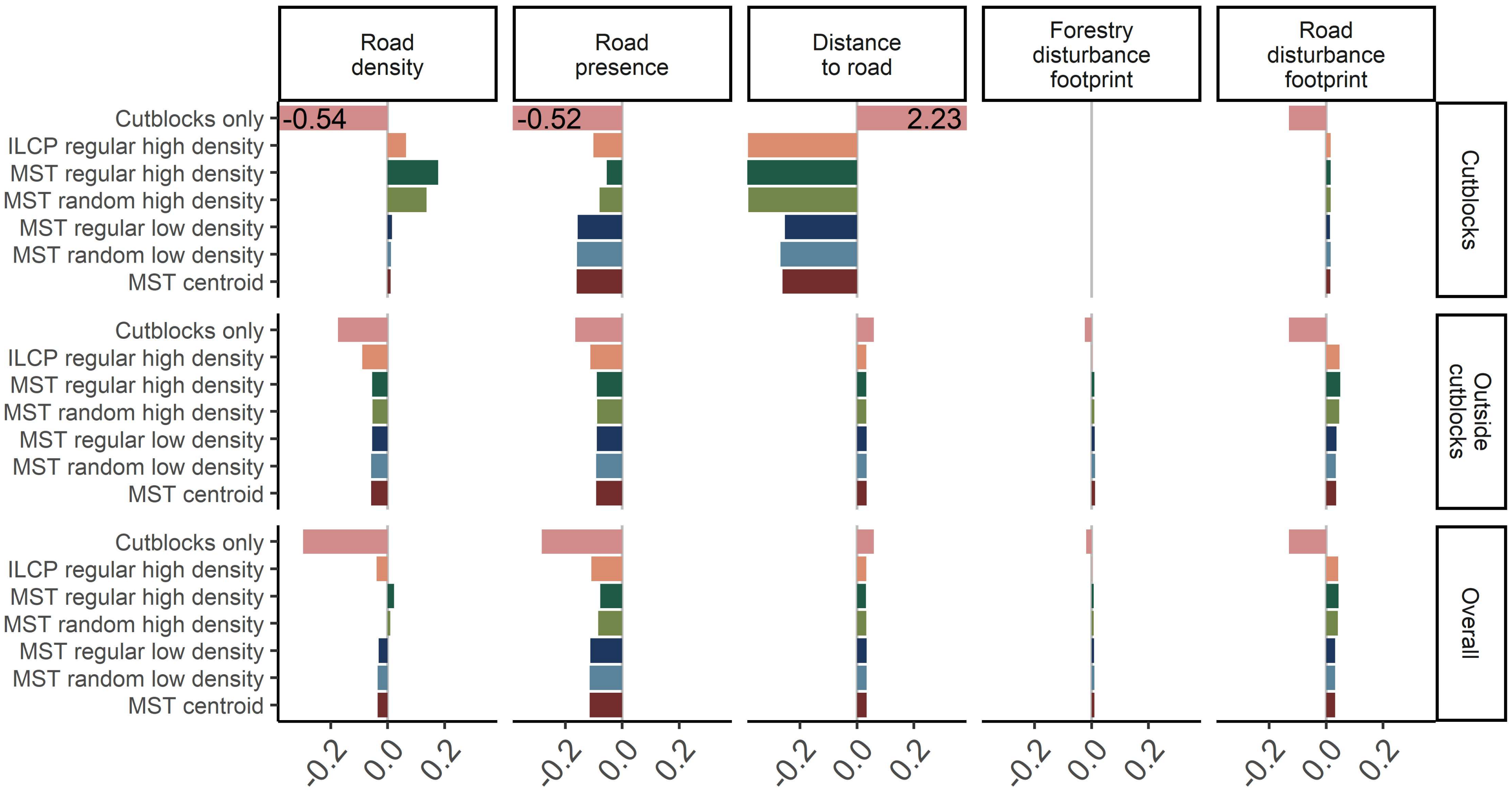

Figure S3: Proportional difference of projected mean from observed mean road metrics using different projection method variants with a 1 ha resolution weight raster in analysis that includes false cutblocks with no observed access roads. Metrics were summarised within cutblocks, outside cutblocks, and overall. Values greater than 0 indicate overprojection of the metric and negative values indicate an underprojection. Note that extreme values for the cutblocks only method within cutblocks are cut off to ensure patterns in the other methods to remain visible. Bars that are cutoff are labeled with the total value.

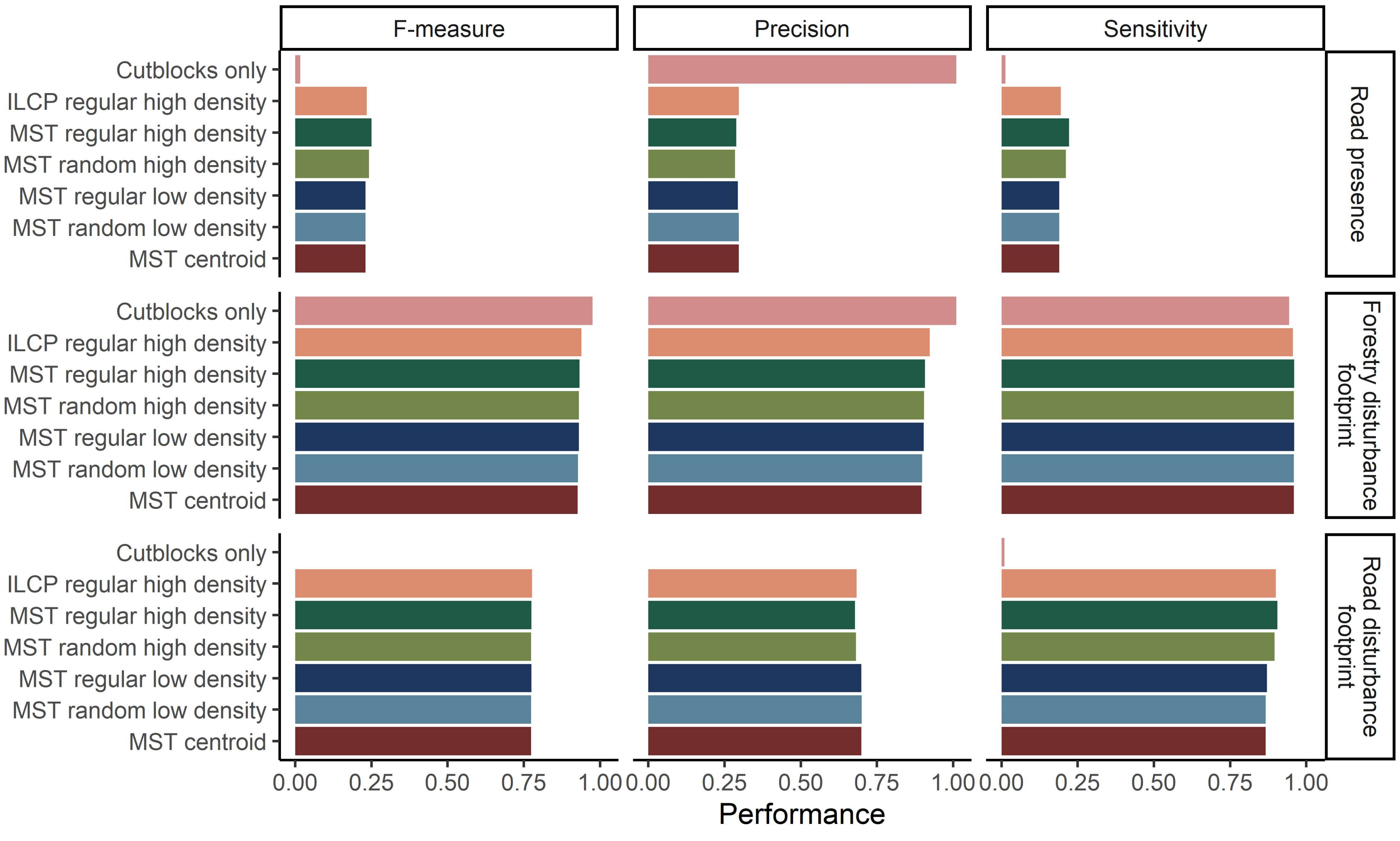

Figure S4: Comparison of performance metrics for each projection method variant using the 1 ha resolution weight raster, including false cutblocks with no observed access roads. F-measure is the harmonic mean of precision and sensitivity, precision is the proportion of predicted roads that are observed roads and sensitivity is the proportion of observed roads that were correctly predicted. Values closer to one indicate better performance.

# 4 Coarse resolution results

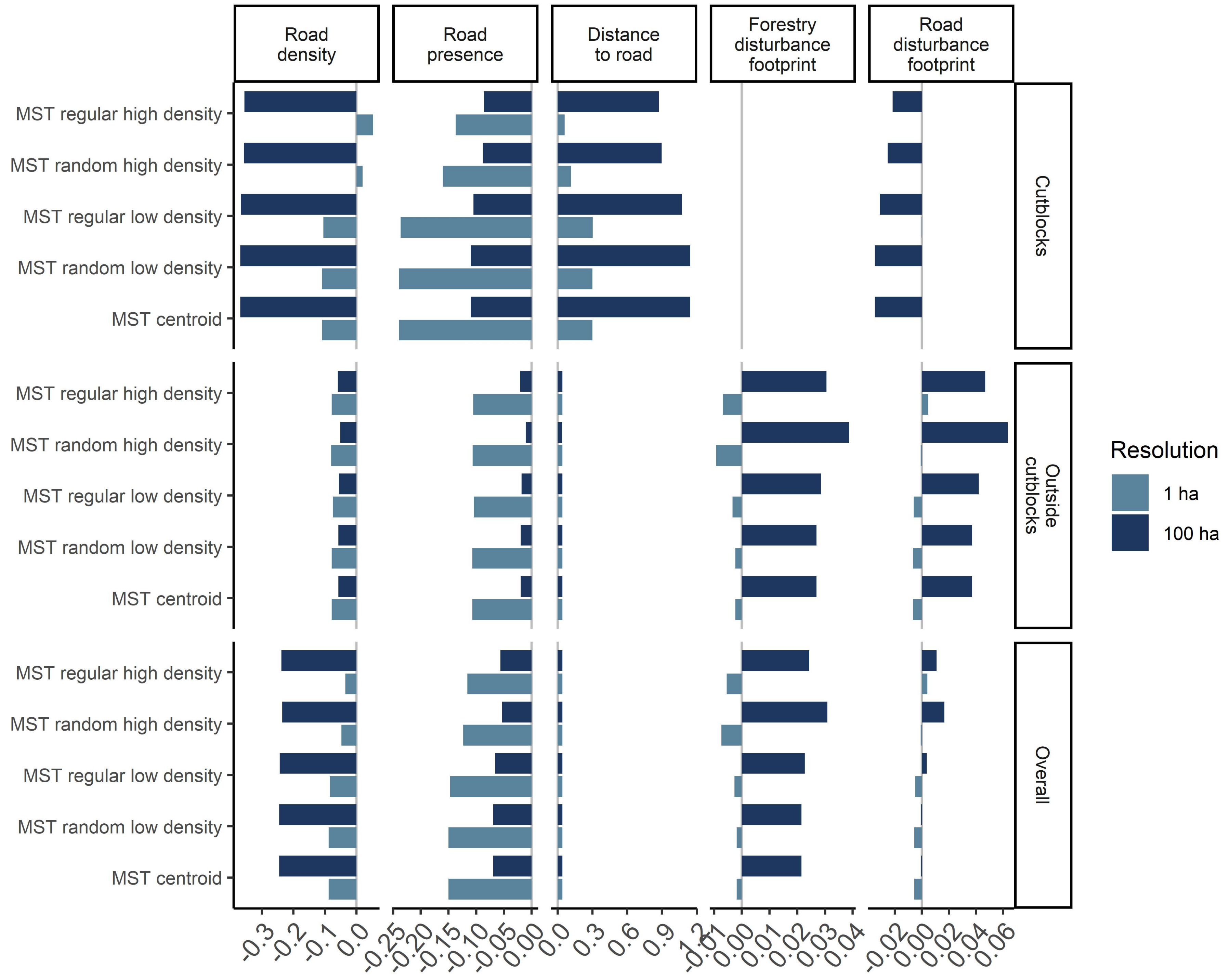

Figure S5: Proportional difference of projected mean from observed mean road metrics using different projection methods for the fine resolution (1 ha) and coarse resolution (100 ha) weight rasters. Metrics were summarised within cutblocks, outside cutblocks, and overall. Values greater than 0 indicate the projection overestimated the metric and negative values indicate an underestimation. Note that each panel has a different scale so bars cannot be compared across metrics.

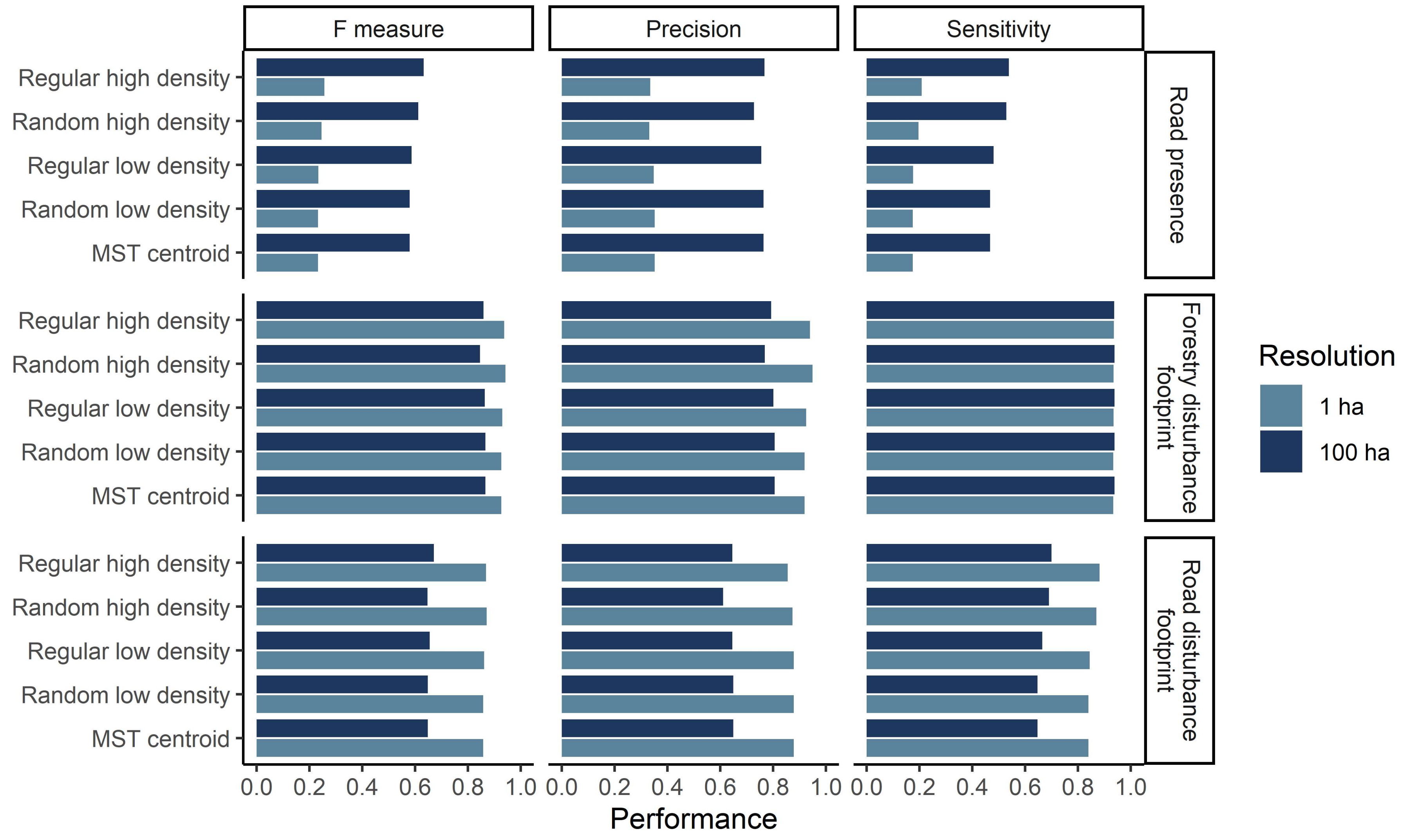

Figure S6: Comparison of performance metrics for each projection method variant for the fine resolutiom (1 ha) and coarse resolution (100 ha) weight rasters. F measure is the harmonic mean of precision and sensitivity, precision is the proportion of predicted roads that are observed roads and sensitivity is the proportion of observed roads that were correctly predicted.

# 5 Benchmarking road projection methods

We compared the speed and peak RAM usage of the MST and ILCP methods with landings regularly sampled at 10 per km$^2$ (high density) or using only the cutblock centroids as landings on weight rasters of 3 different resolutions (1000 m, 500 m, 100 m), giving 18 297, 72 633, and 1 815 825 nodes in the cost graph, respectively. Each cell in the raster becomes a node in the graph used to determine the shortest path which increases the time and memory requirements of the projection. Benchmarking was done using Azure Standard A4 v2 nodes which have 4 cores and 8 GB of RAM, chosen to match a typical laptop. Each projection was run on a separate node and the `peakRAM` package was used to measure the total time to run the projection and the maximum memory used at any point in the analysis (Quinn 2024). We also timed the execution of the Hardy QGIS plugin with the 100 m resolution weights raster.

Execution time and memory requirements increase with the number of nodes in the graph, and the slope of that increase is highest for the MST method (Figure S7). Except on the very small landscape the ILCP method is faster than the MST method. Reducing the number of landings by targeting centroids also reduces execution time (Figure S7). The Hardy method is significantly faster than any of the R package methods.

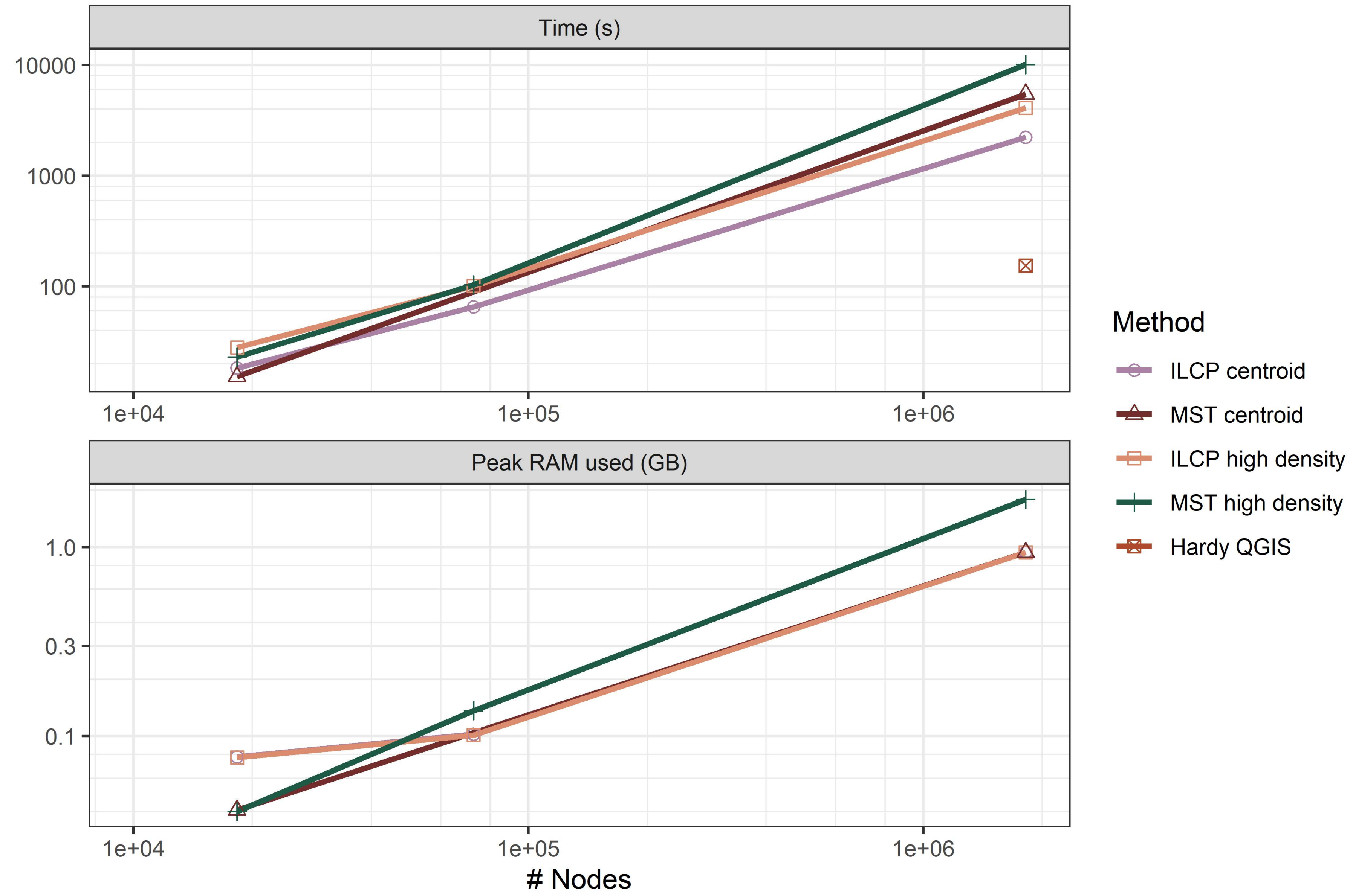

Figure S7: Speed and peak memory usage of the ILCP and MST road projection methods with landings regularly sampled at 10 per km$^2$ (high density) or using cutblock centroids as landings, with weight rasters of three different resolutions (1000 m, 500 m and 100 m) giving 18 297, 72 633 and 1 815 825 nodes in the cost graph, respectively. The time to run the Hardy QGIS plugin with the 100 m raster is also shown. Both number of nodes and time are displayed on log transformed axes.

# 6 Profiling

Profiling with the `Rprof` function shows which parts of a function take the longest. On a test run with 1 million cells in the weight raster and 10 landings we profiled the performance of the ILCP and MST methods. Both methods begin by constructing a full graph, which took 3-5 s. For the ILCP method, iteratively determining the shortest paths is the only other step that takes more than a second to run, and this run time will increase with the number of landings (Figure S8). For the MST method, calculating cost distances among landings and from each landing to the nearest point on the existing road takes the most time, which also increases with the number of landings (Figure S8). Determining the locations of the paths in the minimum spanning tree also takes about 6 seconds. We experimented with using graph construction methods in the package `gdistance` (Etten 2017), but Lochhead et al's (2018; 2022) method is faster (Figure S8).

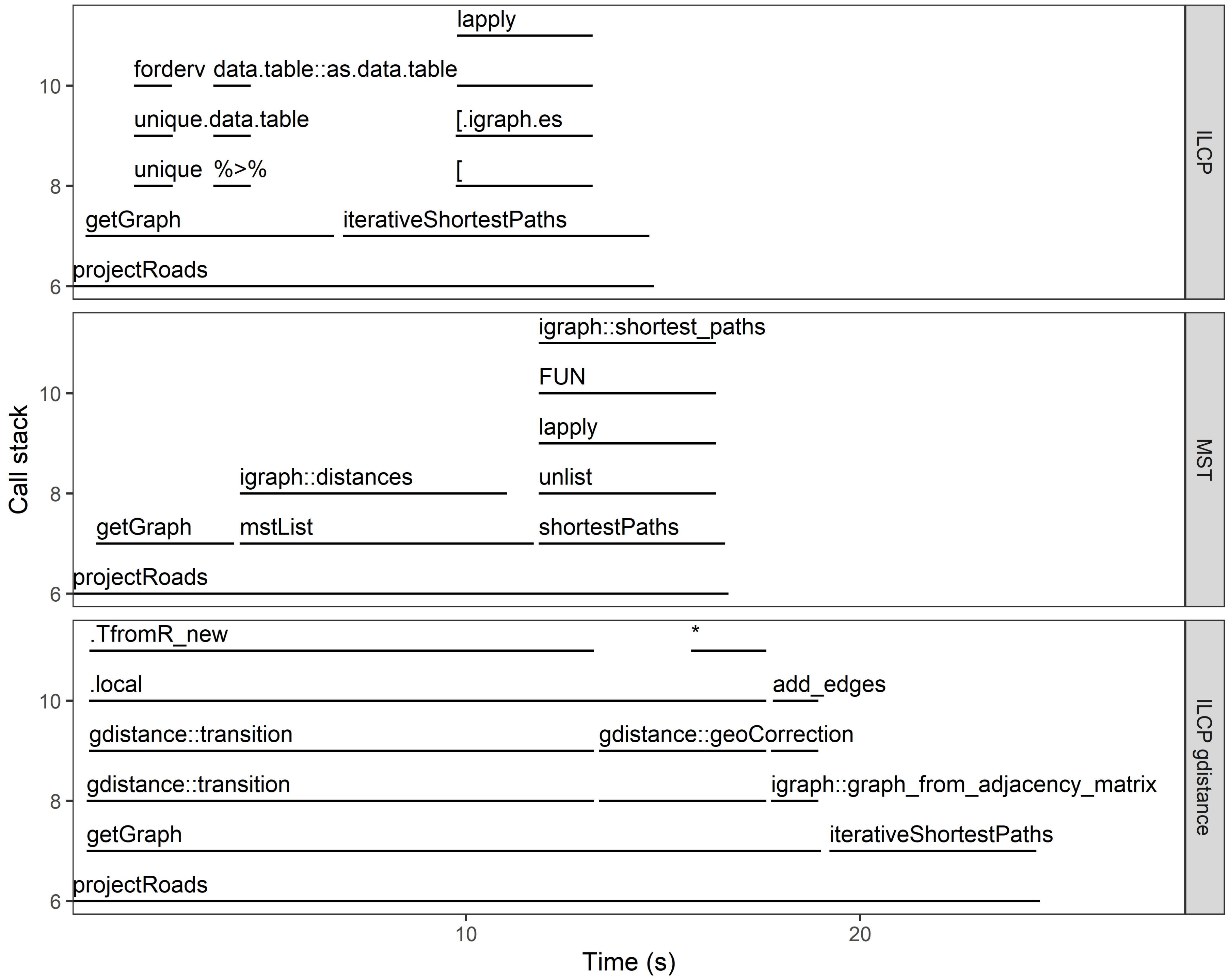

Figure S8: Profiling results for the ILCP, MST, and ILCP using the gdistance package methods, showing functions that take more than one second to run on a cost raster with 1 million cells and 10 landings. Functions higher in the call stack are called by functions lower down.

# 7 Fort Nelson map

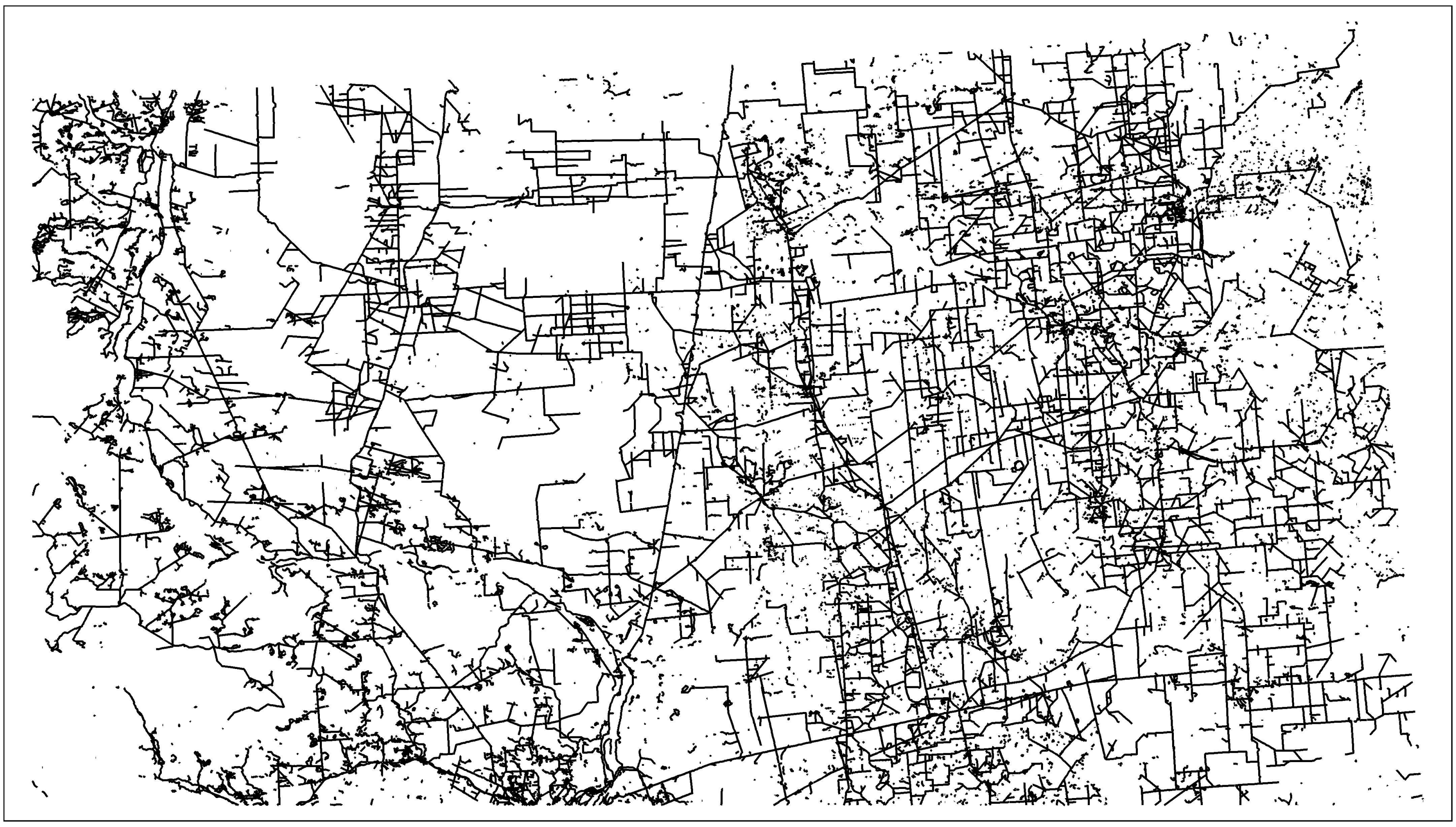

Figure S9: Existing roads in Fort Nelson. Cost based road projection methods are not expected to be useful in areas where roads are built for purposes other than accessing a target location at minimal cost. For example, in the Northeast corner of the Fort Nelson Timber Supply Area seismic lines show a different pattern than foresty roads.

# Literature Cited

Anderson AE, Nelson J (2004) Projecting vector-based road networks with a shortest path algorithm. Canadian Journal of Forest Research 34:1444–1457. https://doi.org/10.1139/x04-030 (https://doi.org/10.1139/x04-030)

Etten J van (2017) R package gdistance: Distances and routes on geographical grids. Journal of Statistical Software 76:1–21. https://doi.org/10.18637/jss.v076.i13 (https://doi.org/10.18637/jss.v076.i13)

Lochhead KD, Kleynhans EJ, Muhly TB (2022) Linking woodland caribou abundance to forestry disturbance in southern British Columbia, Canada. The Journal of Wildlife Management 86:e22149. https://doi.org/10.1002/jwmg.22149 (https://doi.org/10.1002/jwmg.22149)

Lochhead KD, Muhly TB (2018) A demo of road simulation in CLUS (https://github.com/bcgov/clus/blob/master/reports/roads/draft-CLUS-roads.md)

Quinn T (2024) peakRAM: Monitor the Total and Peak RAM Used by an Expression or Function (http://github.com/tpq/peakRAM)

# Colophon

This report was generated on 2025-04-07 14:51:41.592455 using the following computational environment and dependencies:

The current Git commit details are:

```
#> Local:    master C:/Users/EndicottS/Documents/gitprojects/RoadPaper
#> Remote:   master @ origin (https://github.com/LandSciTech/RoadPaper.git)
#> Head:     [ad162e5] 2024-06-13: Merge branch 'master' of https://github.com/LandSciTech/RoadPaper
```