Are environmental and geographic effective surrogates for genetic variation in conservation planning?

Jeffrey O. Hanson[1], *Jonathan R. Rhodes*[2], *Cynthia Riginos*[2], *Hugh P. Possingham*[1], *Richard A. Fuller*[1]

[1]*School of Biological Sciences, The University of Queensland, Brisbane, QLD, Australia*

[2]*School of Geography, Planning and Environmental Management, The University of Queensland, Brisbane, QLD, Australia*

Correspondance should be addressed to jeffrey.hanson@uqconnect.edu.au

03 December 2015

Abstract

Insert abstract here.

Contents

Introduction

Methods

Study area

To address the aims of this study, we obtained species distribution and genomic (AFLP) from (Meirmans *et al.* 2011). This dataset was chosen because it provides genomic data for a multitude of species at a high spatial resolution (approx. $20\text{km}^2 \times 22\text{km}^2$).

```
## compile spatial grid data
# load grid cell centroids
grid.DF <- fread(
    ’extdata/Data_Meirmans_et_al_IntrabioDiv/ReadMe.txt’,
    data.table=FALSE,
    skip=’cell\tLong\tLat’
) %>% rename(
        grid.longitude=Long,
        grid.latitude=Lat
) %>% mutate(
    id=seq_along(grid.latitude)
)
```

```
# load in aflp data
spp.aflp.paths <- dir(
    ’extdata/Data_Meirmans_et_al_IntrabioDiv’,
    ’^.*AFLP\\.dat$’,
    full.names=TRUE
)[seq_len(n.spp)]
spp.BayeScanData.LST <- llply(
    spp.aflp.paths,
    read.BayeScanData
)
```

```
## compile species occurence data
# load in data
spp.loc.paths <- dir(
    ’extdata/Data_Meirmans_et_al_IntrabioDiv’,
    ’^.*locations\\.txt$’,
    full.names=TRUE
)[seq_len(n.spp)]
spp.samples.DF <- ldply(
    seq_along(spp.loc.paths),
    .fun=function(i) {
        x <- mutate(
            fread(spp.loc.paths[i], data.table=FALSE),
            species=gsub(’_locations.txt’, ’’, basename(spp.loc.paths[i]), fixed=TRUE)
        ) %>% rename(
            cell=population,
            sample.longitude=longitude,
            sample.latitude=latitude
        )
        return(x[as.numeric(spp.BayeScanData.LST[[i]]@populations),])
    }
) %>% left_join(
        grid.DF,
        by=’cell’
)
# append species data to grid data.frame (wide-format)
for (i in unique(spp.samples.DF$species))
    grid.DF[[i]] <- replace(
        rep(0, nrow(grid.DF)),
        which(grid.DF$cell %in% filter(spp.samples.DF, species==i)$cell),
        1
    )
```

Genomic data

Loci in the AFLP were classified as adaptive or neutral using BayeScan (version 2.1, using a proability threshold of `r`).

```
# assign cells as populations
spp.BayeScanData.LST <- llply(
    seq_along(unique(spp.samples.DF$species)),
    function(i) {
        bd <- spp.BayeScanData.LST[[i]]
        bd@populations <- filter(spp.samples.DF, species==unique(spp.samples.DF$species)[i])$ce
        return(bd)
    }
)
# run BayeScan
spp.BayeScan.LST <- llply(
    spp.BayeScanData.LST,
    run.BayeScan,
    threshold=bs.threshold,
    threads=bs.threads,
    n=bs.n,
    thin=bs.thin,
    nbp=bs.nbp,
    pilot=bs.pilot,
    burn=bs.burn
)
# run MDS
spp.mds.LST <- llply(
    spp.BayeScan.LST,
    function(i) {
        ‘names<-‘(llply(c(’adaptive’, ’neutral’), function(j) {
            if (sum(i@results@fst==j)==0)
                return(NULL)
            return(
                mds(
                    i,
                    metric=’gower’,
                    type=j,
                    k=mds.k,
                    trymax=mds.trymax
                )
            )
        }), c('adaptive','neutral'))
    }
)
```

```
## Warning in daisy(cbind(as.data.frame(x@matrix == 1), 1), metric = metric, :
## at least one binary variable has not 2 different levels.
```

```
## Run 0 stress 0.2366402
## Run 1 stress 0.2399884
## Run 2 stress 0.2414222
## Run 0 stress 0.2161925
## Run 1 stress 0.2163052
## ... procrustes: rmse 0.02846903  max resid 0.1498994
## Run 2 stress 0.2152845
## ... New best solution
## ... procrustes: rmse 0.04697536  max resid 0.1799123
```

```
## Warning in daisy(cbind(as.data.frame(x@matrix == 1), 1), metric = metric, :
## at least one binary variable has not 2 different levels.
```

```
## Run 0 stress 0.2418134
## Run 1 stress 0.2474117
## Run 2 stress 0.2478307
```

```
## Warning in daisy(cbind(as.data.frame(x@matrix == 1), 1), metric = metric, :
## at least one binary variable has not 2 different levels.
```

```
## Run 0 stress 0.2254906
## Run 1 stress 0.2434694
## Run 2 stress 0.2278421
## Run 0 stress 0.2005142
## Run 1 stress 0.2043358
## Run 2 stress 0.2287576
## Run 0 stress 0.2333163
## Run 1 stress 0.242894
## Run 2 stress 0.2362956
```

```
# store mds rotations for each sample
spp.samples.DF <- ldply(seq_along(unique(spp.samples.DF$species)), .fun=function(i) {
    x <- filter(spp.samples.DF, species==unique(spp.samples.DF$species)[i])
    for (j in c('adaptive', 'neutral')) {
        if (!is.null(spp.mds.LST[[i]][[j]])) {
            x <- cbind(
                x,
                ‘names<-‘(
                    as.data.frame(spp.mds.LST[[i]][[j]]$points),
                    paste0(j,’_d’,seq_len(mds.k))
                )
            )
        }
    }
    return(x)
})
# store mds average rotation for each grid
for (i in seq_along(unique(spp.samples.DF$species))) {
    for (j in c(’adaptive’, ’neutral’)) {
        if(!is.null(spp.mds.LST[[i]][[j]])) {
            curr.sub <- filter(spp.samples.DF, species==unique(spp.samples.DF$species)[i])
            for (k in seq_len(mds.k)) {
                curr.vals <- tapply(
                    curr.sub[[paste0(j,’_d’,k)]],
                    curr.sub$cell,
                    FUN=mean
                )
                curr.pos <- match(names(curr.vals), grid.DF$cell)
                grid.DF[curr.pos,paste0(unique(spp.samples.DF$species)[i],’_’,j,’_d’,k)] <- cu
            }
        }
    }
}
```

Surrogate data

```
## create spatial data
# grid data as SpatialPolygonsDataFrame
grid.PTS <- SpatialPoints(as.matrix(grid.DF[,2:3]))
grid.PLY <- grid.PTS %>%
    points2grid(tolerance=0.05) %>%
    as(’SpatialPolygons’)
grid.PLY <- grid.PLY[sapply(gIntersects(grid.PTS, grid.PLY, byid=TRUE, returnDense=FALSE), ‘[[
    spChFIDs(
        as.character(seq_len(nrow(grid.DF)))
    ) %>%
    SpatialPolygonsDataFrame(
        data=grid.DF
    )
grid.PLY@proj4string <- wgs1984
grid.PPLY <- spTransform(grid.PLY, europeEA)
# sample data as SpatialPoints
spp.sample.PTS <- SpatialPointsDataFrame(
    coords=as.matrix(spp.samples.DF[,5:6]),
    data=spp.samples.DF,
    proj4string=wgs1984
)
spp.sample.PPTS <- spTransform(spp.sample.PTS, europeEA)
## extract geographic data
centroids.DF <- gCentroid(grid.PPLY, byid=TRUE) %>% slot(’coords’) %>%
    as.data.frame() %>% ‘names<-‘(paste0(’geo_d’,1:2))
grid.DF <- cbind(grid.DF, centroids.DF)
## extract climatic data
# load climatic data
bioclim.STK <- stack(’extdata/BioClim_variables/bioclim_pca.tif’)
# extract mean for each cell for each principle component
extract.DF <- grid.PPLY %>% rasterize(bioclim.STK, field=’id’) %>%
    zonal(x=bioclim.STK) %>% as.data.frame() %>% select(-1) %>%
    ‘names<-‘(paste0(’env_d’,seq_len(nlayers(bioclim.STK))))
# merge with grid.DF
grid.DF <- cbind(grid.DF, extract.DF)
## update spatial objects
grid.PLY@data <- grid.DF
grid.PPLY@data <- grid.DF
```

Prioritisations

Results

Discussion

Acknowledgements

JOH is funded by an Australian Postgraduate Award (APA) scholarship. RAF has an Australian Research Council Future Fellowship. This work was supported by the Centre of Excellence for Environmental Decisions (CEED) and the Landscape Ecology and Conservation Group (LEC) at The University of Queensland.

References

Supporting Information

Appendix S1: Species distributions

```
## plot map of species distributions
# download basemap
data(countriesHigh)
countries.FPLY <- countriesHigh[
    countriesHigh$ADMIN %in% c(
        ’Italy’, ’Switzerland’, ’France’, ’Austria’,
        ’Germany’, ’Slovenia’, ’Croatia’, ’Hungary’,
        ’Monaco’, ’Germany’
    )
,] %>% spFortify
# fortify data
grid.FPLY <- spFortify(grid.PLY)
spp.grid.FPLY <- ldply(unique(spp.samples.DF$species), function(x) {
        z <- grid.FPLY[,c(’long’, ’lat’, ’group’, x),drop=FALSE]
        names(z)[4] <- ’presence’
        z$species <- gsub(’\\_’, ’ ’, x)
        return(z)
    }
)
# plot species data
ggplot() +
    geom_polygon(data=countries.FPLY, aes(x=long, y=lat, group=group),
        fill=’grey20’, color=’grey80’) +
    geom_polygon(data=spp.grid.FPLY, aes(x=long, y=lat,
        group=group, fill=presence), alpha=0.8, color=’grey10’) +
    theme_classic() +
    guides(fill=guide_legend(title=’Presence’)) +
    theme(axis.ticks=element_blank(), axis.text=element_blank()) +
    coord_cartesian(
        xlim=buffered.range(grid.FPLY$long, 0.05),
        ylim=buffered.range(grid.FPLY$lat, 0.05)
    ) +
    xlab(’’) +
    ylab(’’) +
    facet_wrap(~ species, ncol=4)
```

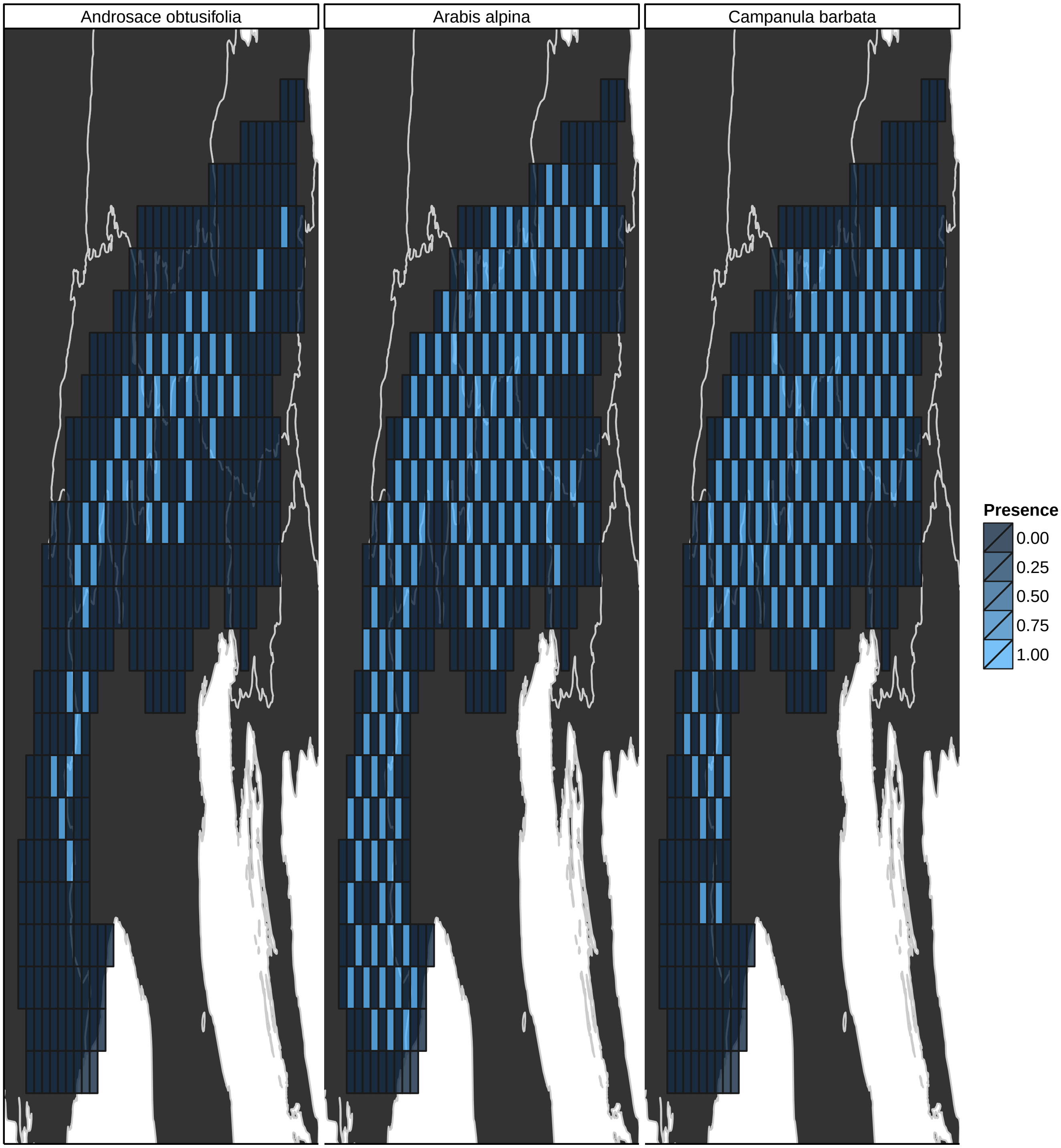

Figure 1 Species distributions. Squares represent planning units. For a given species, planning units that were found to be inhabited are denoted with bright blue.

```
# calculate species richness
grid.PLY$Species_richness <- grid.PLY@data %>%
    select(5:(4+n.spp)) %>% as.matrix() %>% rowSums()

# plot species richness
ggplot() +
    geom_polygon(data=countries.FPLY, aes(x=long, y=lat, group=group),
        fill=’grey20’, color=’grey80’) +
    geom_polygon(data=spFortify(grid.PLY), aes(x=long, y=lat,
        group=group, fill=Species_richness), alpha=0.8, color=’grey10’) +
    guides(fill=guide_legend(title=’Count (#)’)) +
    theme_classic() +
    theme(axis.ticks=element_blank(), axis.text=element_blank()) +
    coord_cartesian(
        xlim=buffered.range(grid.FPLY$long, 0.05),
        ylim=buffered.range(grid.FPLY$lat, 0.05)
    ) +
    xlab(’’) +
    ylab(’’) +
    ggtitle(’Species richness’)
```

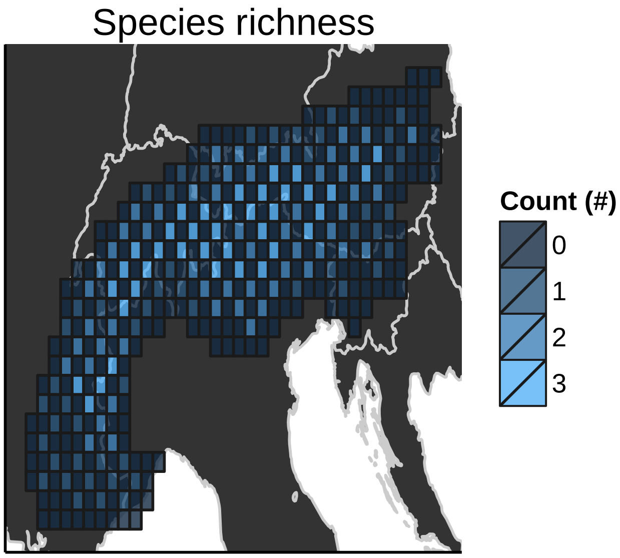

Figure 2 Species richness. Squares denote planning units. Planning units with a brighter color are inhabited by more species.

Appendix S2: Genomic MDS

```
knitr::kable(
    ldply(
        seq_along(unique(spp.samples.DF$species)),
        function(i) {
            ldply(
                seq_along(spp.mds.LST[[i]]),
                function(j) {
                data.frame(
                    Species=paste0('\\textit{',gsub('\\_', ' ', unique(spp.samples.DF$species)
                    Loci=names(spp.mds.LST[[i]])[j],
                    Stress=spp.mds.LST[[i]][[j]]$stress,
                    Converged=spp.mds.LST[[i]][[j]]$converged
                )
            })
        }
    ),
    digits=2,
    caption='Summary of nonmetric-dimensional scaling (MDS) analyses on genetic variation for
)
```

Species	Loci	Stress	Converged
Androsace obtusifolia	adaptive	0.24	FALSE
Androsace obtusifolia	neutral	0.22	FALSE
Arabis alpina	adaptive	0.24	FALSE
Arabis alpina	neutral	0.23	FALSE
Campanula barbata	adaptive	0.20	FALSE
Campanula barbata	neutral	0.23	FALSE

Table 1 Summary of nonmetric-dimensional scaling (MDS) analyses on genetic variation for each species.

Appendix S3: Distribution maps of intra-specific variation

```
plot.spp.mds <- function(i) {
    do.call(
        grid.arrange,
        append(
            unlist(llply(c('adaptive','neutral'), function(g) {
                llply(seq_len(mds.k), function(k) {
                    ggplot() +
                        geom_polygon(data=countries.FPLY, aes(x=long, y=lat, group=group),
                            fill=’grey20’, color=’grey80’) +
                        geom_polygon(data=grid.FPLY, aes_string(x=’long’, y=’lat’,
                            group=’group’, fill=paste0(unique(spp.samples.DF$species)[i], ’_’,
                            alpha=0.8, color=’grey10’) +
                        guides(fill=guide_legend(title=’ ’)) +
                        theme_classic() +
                        theme(axis.ticks=element_blank(), axis.text=element_blank(),
                            plot.margin=unit(c(0,0,0,0),’cm’)) +
                        coord_cartesian(
                            xlim=buffered.range(grid.FPLY$long, 0.05),
                            ylim=buffered.range(grid.FPLY$lat, 0.05)
                        ) +
                        xlab(’’) +
                        ylab(’’) +
                        ggtitle(paste0(g,’ (’,k,’)’))
                })
            }),recursive=FALSE),
            list(ncol=2)
        )
    )
}
```

```
plot.spp.mds(1)
```

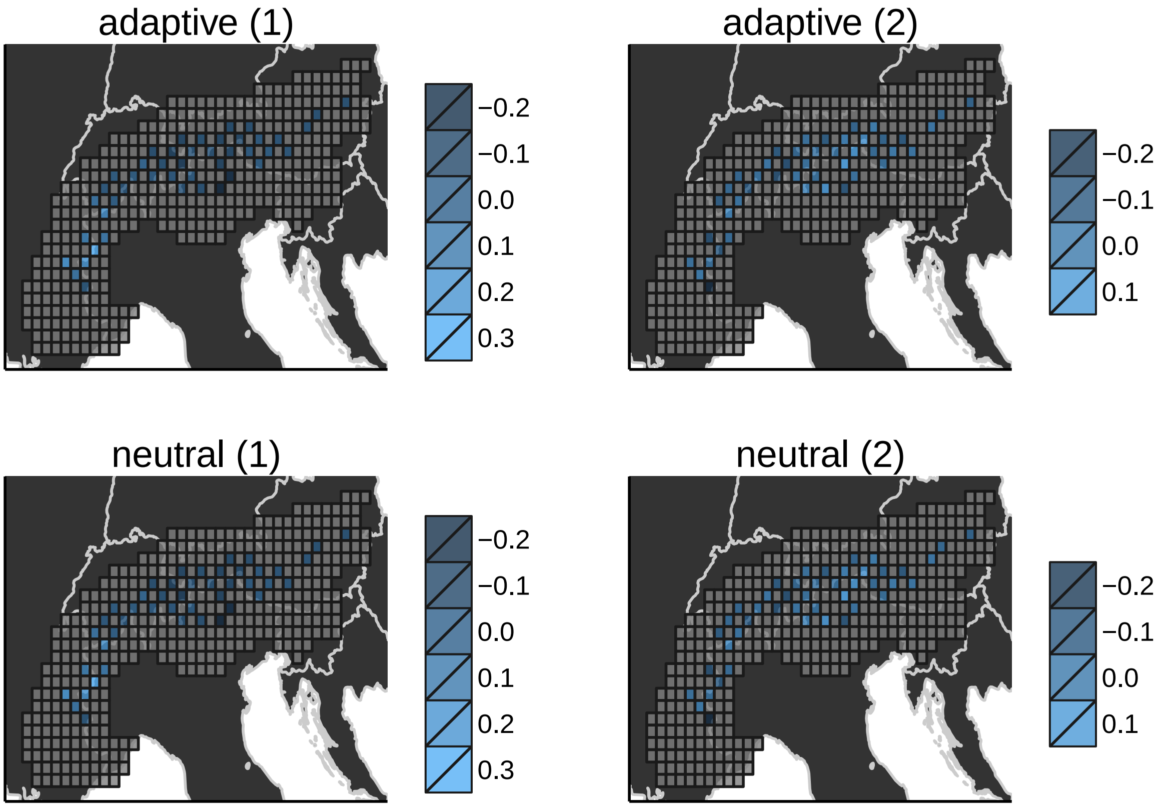

Figure 3 Distribution of adaptive and neutral genetic variation in *Androsace obtusifolia*. Each square represents a planning unit. The color of each planning unit panel corresponds to ordination values. Planning units with similar colors contain individiduals with similar genetic variation.

```
plot.spp.mds(2)
```

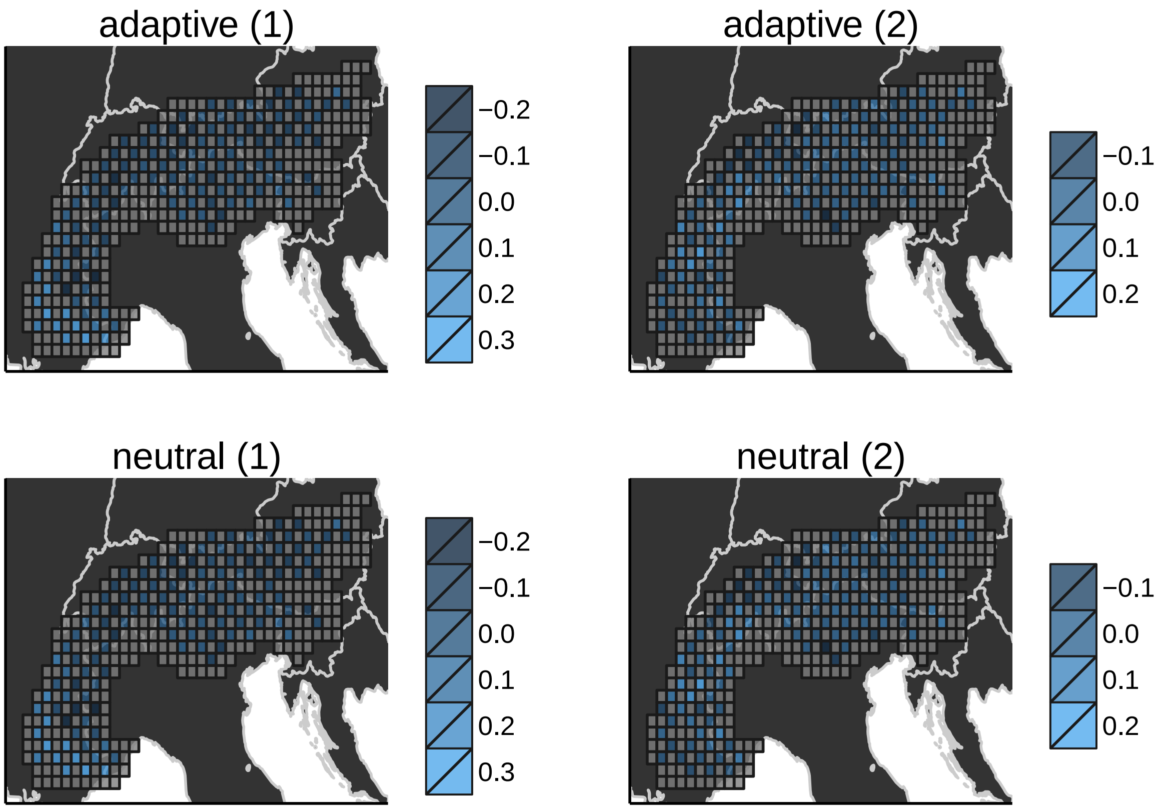

Figure 4 Distribution of adaptive and neutral genetic variation in *Arabis alpina*. See Figure XX caption for conventions.

```r
plot.spp.mds(3)
```

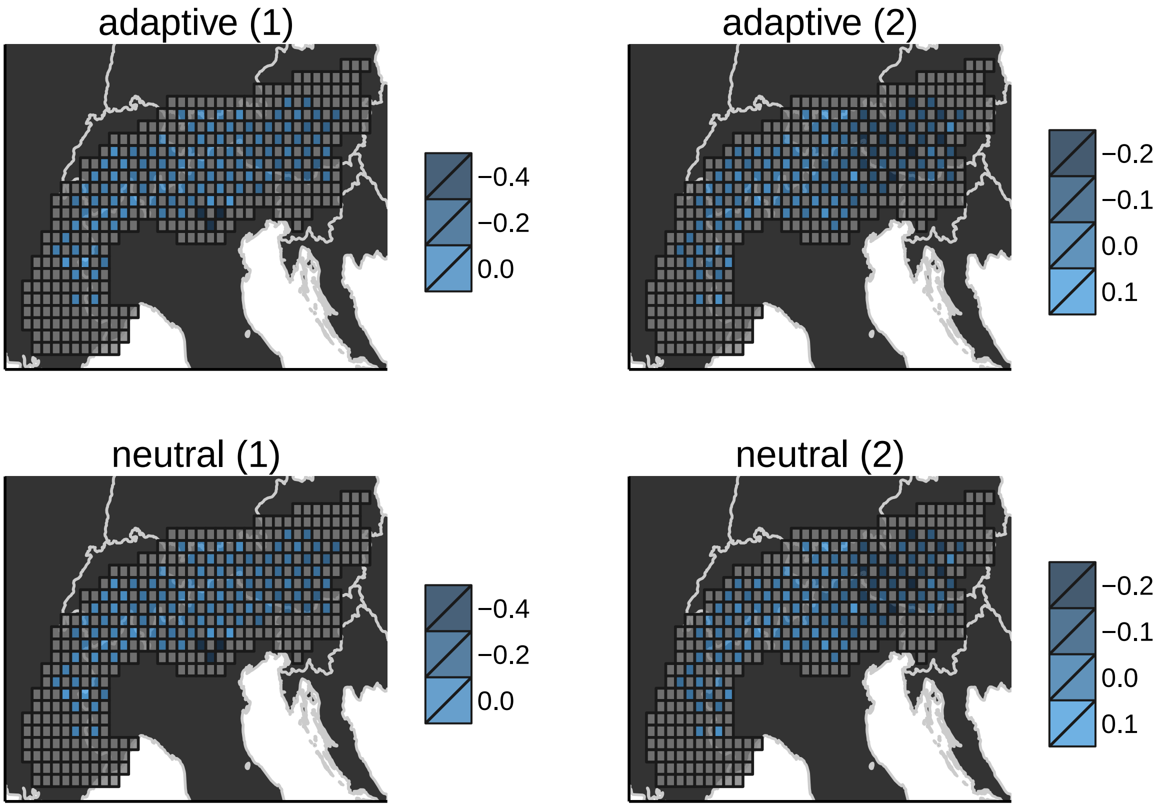

Figure 5 Distribution of adaptive and neutral genetic variation in *Campanula barbata*. See Figure XX caption for conventions.

Appendix S4: Principle components analysis on climatic variation

```
## load pca summary
pca.DF <- read.table(’extdata/BioClim_variables/pca.TXT’, skip=80) %>% ‘names<-‘(
    c(’Principle Component’, ’Eigen Value’, ’Variation explained (%)’,
    ’Accumulative variation explained (%)’)
)
## make results table showing Eigen values
knitr::kable(
    pca.DF,
    digits=2,
    caption=’Summary of priciniple components analysis (PCA) on bioclimatic variation across t
)
```

Principle Component	Eigen Value	Variation explained (%)	Accumulative variation explained (%)
1	216765.14	82.67	82.67
2	38177.84	14.56	97.23

Principle Component	Eigen Value	Variation explained (%)	Accumulative variation explained (%)
3	5356.75	2.04	99.27
4	1216.67	0.46	99.73
5	700.39	0.27	100.00

Table 2 Summary of priciniple components analysis (PCA) on bioclimatic variation across the study area. The first two principle components (PCs) were used for subsequent analysis.

Appendix S5: Maps of climatic variation

```
do.call(
    grid.arrange,
        append(
        llply(grep(’^env\\_.*$’, names(grid.DF), value=TRUE), function(x) {
            ggplot() +
                geom_polygon(data=countries.FPLY, aes(x=long, y=lat, group=group),
                    fill=’grey20’, color=’grey80’) +
                geom_polygon(data=grid.FPLY, aes_string(x=’long’, y=’lat’,
                    group=’group’, fill=x),
                    alpha=0.8, color=’grey10’) +
                guides(fill=guide_legend(title=’ ’)) +
                theme_classic() +
                theme(axis.ticks=element_blank(), axis.text=element_blank(),
                    plot.margin=unit(c(0,0,0,0),’cm’)) +
                coord_cartesian(
                    xlim=buffered.range(grid.FPLY$long, 0.05),
                    ylim=buffered.range(grid.FPLY$lat, 0.05)
                ) +
                xlab(’’) +
                ylab(’’) +
                ggtitle(paste0(’PC ’, substr(x, nchar(x), nchar(x))))
        }),
        list(ncol=2)
    )
)
```

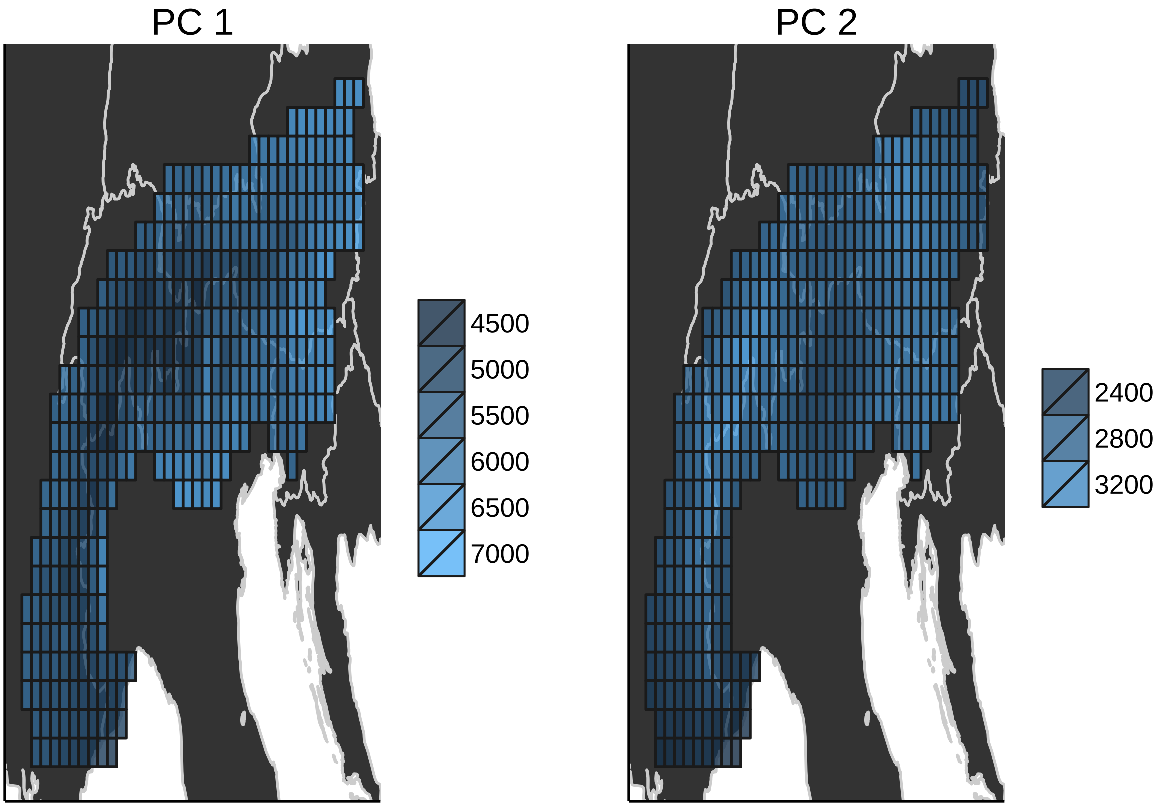

Figure 6 Climatic variation. Each panel depicts variation based on a different principle component (PC). Sqaures represent planning units. The color of each planning unit denotes the average pricinipile component value of pixels inside it. Planning units with more similar colors have more similar climates regimes.

Meirmans, P., Goudet, J., IntraBioDiv Consortium, Gaggiotti, O. (2011) Ecology and life history affect different aspects of the population structure of 27 high-alpine plants. *Molecular Ecology.* **20**, 3144–3155.