

Wildfire Recovery Playbook Review - 2024

**Prepared for
Pacific Salmon Foundation**

**Prepared by
Al Irvine, B.Sc., R.P.Bio.
New Graph Environment Ltd.**

Version 0.0.1 2024-03-11



new graph environment

Table of Contents

Acknowledgement	iv
Introduction	1
Praise	3
A Few Thoughts	5
Reproducibility	5
Tools for Reproducibility	5
Open Source - Iterative Reporting	5
Tables and Figures	6
Conclusion	9
References	11
Session Info	13

Acknowledgement

Modern civilization has a long journey ahead to acknowledge and address the historic and ongoing impacts of colonialism that have resulted in harm to the cultures and livelihoods living interconnected with our ecosystems for many thousands of years.

Introduction

This report is available as pdf and as an online [interactive report](https://newgraphenvironment.github.io/wildfire_recovery_playbook_review_2024/) at https://newgraphenvironment.github.io/wildfire_recovery_playbook_review_2024/. We recommend viewing online as the web-hosted html version contains more features and is more easily navigable.

Thank you for the opportunity to review the Playbook to Guide Landscape Recovery Strategies & Priorities for Salmon Habitat Following Major Wildfires (EDI Environmental Dynamics Inc. 2023). We are very impressed with the work that has been done!

Praise

1. The playbook is well organized, well written and easy to follow. It must have taken forever to write and all the contributors should be showered in praise for their efforts and should be super proud of the final product.
2. The playbook is comprehensive and covers most the necessary topics. The content is well researched and we are confident the information we are familiar with is accurate.
3. The playbook is a valuable resource for anyone involved in wildfire recovery with the links to external resources are particularly helpful.

A Few Thoughts

Reproducibility

There does not seem to be a lot of information regarding the value of reproducibility for restoration planning and from our perspective that is a major barrier to progress... For example - the use of scripts to automate workflows and the sharing of workflows online. From our perspective, there is extraordinary value in using scripts and packages to automate workflows and promote collaboration. Rather than each professional doing their own analysis on their own custom datasets with many points and clicks it should be encouraged that professionals contribute to open-source programming that allows centralization of workflows. Examples of the types of packages that can be used to automate workflows are presented [here](#).

Tools for Reproducibility

Although many of the resources in the above linked csv are reports there are also many tools. An example of a key tool is [bcdata](#) which is a package that allows the user to access data from the BC Data Catalogue. We use this package to build a mirror of government data that is used in our projects. Because the loading of data to a postgres database is done automatically and the names of the data schemas and tables are equivalent to those served out by the province our GIS analysis is portable and analysis can be rerun with fresh data over time. We can collaborate on a level that is not achievable without much more effort and skill than is required when not using a tool such as this that facilitates reproducibility. Other key tools include [fwapg](#) and [bcfishpass](#) among others.

There are many wonderful conceptual workflows presented in this report and some text that indicates that investments should be made so these workflows can be automated and shared openly to the greatest extent possible would be a great addition to the document. A great example of a conceptual workflow that could be automated (with the help of packages like bcddata!) is Presented in Figure 24 - "Conceptualization of combining spatial data (GIS) on hazards and consequences to visualize wildfire risk on a single 2-dimensional wildfire "risk map". Adapted from Alwathaf and Mansouri (2011)."

Open Source - Iterative Reporting

There is a key question we have been asking ourselves lately and that is

"What is the most useful way to present information?"

A Few Thoughts

From our perspective - the most useful way to present information is in a version controlled format that allows it to be interactive, reproducible, read online, served out as a static document, updated, copied, used as a template, etc. Iterative reporting with a permissive license is a great way to do all those things. Although there is a steep learning curve to produce this type of product and it can be quite time consuming - it then allows leveraging of all the components of the work to future projects for everyone. Licenseing is also key and should be considered. Here is an example of a license that promotes collaboration and sharing of work [here](#).

From our perspective - the playbook is the type of document that is well suited to an iterative reporting format such as can be demonstrated within this [code repository](#) which builds and serves this [online report](#). We would like to suggest that in future iterations, it be considered that the playbook is converted to a dynamic document because:

Tables and Figures

There are an incredible number of tables within the document that may have taken a very long time to format. We suggest the tables be converted to simple csvs or pages of an excel workbook and pulled into the project using scripts. This would allow others to access the data and use it in their own projects. The tables could also then be easily updated with new data as it becomes available and the formatting would be handled by the scripting. An example of this can be seen here in Table [1](#) with the actual file served [here](#):

Table 1: Table 2:

Resource	URL	Details
bcfishpass	https://github.com/smnorris/bcfishpass	Build and maintain an aquatic connectivity / fish passage database for British Columbia to track known barriers to fish passage, model potential barriers to fish passage, model passability/accessibility of streams based on species swimming ability, model streams with potential for spawning and rearing activity, etc
fpr	https://github.com/NewGraphEnvironment/fpr	Fish Passage Reporting functions for British Columbia. R package used for cleaning, transforming and extracting data to build interactive reports related to fish passage planning including fish passage assessments and habitat confirmation assessments at road-stream crossings. In
dff-2022	https://github.com/NewGraphEnvironment/dff-2022	Building digital field forms for field work in British Columbia, mobile QGIS projects and processing data collected using standardized provincial data layer and data collection templates. Utilizes open source software including R, python, bash and QGIS and others.
rpf	https://github.com/NewGraphEnvironment/rpf	Reproducible Field Projects for British Columbia. R package to standardize and track the creation of collaborative GIS projects for aquatic restoration/conservation planning and field work. Under development incorporating many dff-2022 scripts into R package functions.
Restoration Framework	https://github.com/NewGraphEnvironment/restoration_framework	Public repository for development of and linking to restoration / conservation planning tools and concepts.
Restoring Fish Passage in the Peace Region 2023	https://newgraphenvironment.github.io/fish_passage_peace_2023_reporting/	REPORT NOT YET LIVE (20240220). Interactive reporting for 2023 fish passage restoration planning activities in the Parsnip, Carp and Crooked Watershed Groups north of Prince George in the Peace Region. Designs completed for two crossings in the Table River watershed (Parship River watershed group) and materials purchased for remediation. Phase 1 and Phase 2 assessments in Carp and Crooked watershed groups. Fish sampling with PIT tagging conducted in Parsnip River watershed group. Aerial imagery collected by drone. Repository located at https://github.com/NewGraphEnvironment/fish_passage_peace_2023_reporting
Restoring Fish Passage in the Peace Region 2022	https://newgraphenvironment.github.io/fish_passage_peace_2022_reporting/	Interactive reporting for 2022 fish passage restoration planning activities in the Parsnip, Carp and Crooked Watershed Groups north of Prince George in the Peace Region. Includes interactive planning dashboard. Crossing in Missinka watershed replaced and design done for site on Chuchinka-Arctic FSR. Fish sampling with PIT tagging conducted. Aerial imagery collected by drone.
Restoring Fish Passage in the Peace Region 2021	https://newgraphenvironment.github.io/fish_passage_parnip_2021_reporting/	Interactive reporting for 2021 fish passage restoration planning activities in the Parsnip Watershed Group north of Prince George in the Peace Region. Includes interactive planning dashboard and

Any figures used in the reporting could be saved in the project and used by anyone else in their work to tell their own story and help their teams achieve their goals. An example of this can be seen here Table 1 with the actual file served [here](#):



shutterstock.com · 715797241

A Few Thoughts

- The references in this document are incredible. There are so many and they are extremely valuable. Thank you. To maximize the value of these references, they could be saved in a .bib file that can be imported into a reference manager such as Zotero or Mendeley. This would allow the references to be transferred all at once to others libraries so they can be used in other projects and updated as needed. An example of this can be seen [here](#).
- The document is quite long and from some perspectives may be considered a touch difficult to navigate. It would be helpful to have the report served online with a table of contents on the left hand side that allows the reader to jump to the section. An example of this can be seen [here](#). In this particular situation this is done using the open source package bookdown and the document is served by [github pages](#).
- Although it requires a bit of gymnastics to get it set up the document can also be served in a way that allows the reader to download updated pdf versions of the report that are formatted for printing and suitable for upload to a central database such as Ecocat. An example of this can be seen [here](#) where a download button at the top left of the document allows the user to download a pdf version of the report from [this url](#).

Conclusion

It can't be fun putting out documents for review. It's great that you are doing that. Smart. You have done an excellent job and we are very grateful for the opportunity to review the playbook. We hope that our feedback does not come across as too critical as overall we are very impressed with the work that has been done and look forward to mining the document for resources and ideas for our own projects.

Thank you!!!

References

EDI Environmental Dynamics Inc. 2023. "Playbook to Guide Landscape Recovery Strategies & Priorities for Salmon Habitat Following Major Wildfires."

Session Info

– Session info

Session Info

setting	value
version	R version 4.3.2 (2023-10-31)
os	<NA>
system	aarch64, darwin20
ui	RStudio
language	(EN)
collate	en_US.UTF-8
ctype	en_US.UTF-8
tz	America/Vancouver
date	2024-03-11
rstudio	2023.12.1+402 Ocean Storm (desktop)
pandoc	3.1.1 @ /Applications/RStudio.app/Contents/Resources/app/quarto/bin/tools/ (via rmarkdown)

– Packages

package	* version	date (UTC)	lib	source
bcddata	0.4.1	2023-03-18	[1]	CRAN (R 4.3.0)
bit	4.0.5	2022-11-15	[1]	CRAN (R 4.3.0)
bit64	4.0.5	2020-08-30	[1]	CRAN (R 4.3.0)
bookdown	* 0.38	2024-03-04	[1]	CRAN (R 4.3.1)
bslib	0.6.1	2023-11-28	[1]	CRAN (R 4.3.2)
cachem	1.0.8	2023-05-01	[1]	CRAN (R 4.3.0)
cellranger	1.1.0	2016-07-27	[1]	CRAN (R 4.3.0)
chk	0.9.1.9000	2024-01-08	[1]	Github
(poissonconsulting/chk@7ebd4b3)				
class	7.3-22	2023-05-03	[2]	CRAN (R 4.3.2)
classInt	0.4-10	2023-09-05	[1]	CRAN (R 4.3.0)
cli	3.6.2	2023-12-11	[1]	CRAN (R 4.3.1)
colorspace	2.1-0	2023-01-23	[1]	CRAN (R 4.3.0)
conflicted	* 1.2.0.9000	2024-01-11	[1]	Github (r-
lib/conflicted@321d77c)				
crayon	1.5.2	2022-09-29	[1]	CRAN (R 4.3.0)
DBI	1.2.1	2024-01-12	[1]	CRAN (R 4.3.2)
dbplyr	2.4.0	2023-10-26	[1]	CRAN (R 4.3.1)
devtools	* 2.4.5	2022-10-11	[1]	CRAN (R 4.3.0)
digest	0.6.34	2024-01-11	[1]	CRAN (R 4.3.1)
dplyr	* 1.1.4	2023-11-17	[1]	CRAN (R 4.3.1)
e1071	1.7-14	2023-12-06	[1]	CRAN (R 4.3.1)
ellipsis	0.3.2	2021-04-29	[1]	CRAN (R 4.3.0)
evaluate	0.23	2023-11-01	[1]	CRAN (R 4.3.2)
fansi	1.0.6	2023-12-08	[1]	CRAN (R 4.3.1)
fastmap	1.1.1	2023-02-24	[1]	CRAN (R 4.3.0)
forcats	* 1.0.0	2023-01-29	[1]	CRAN (R 4.3.0)
fpr	* 1.1.0	2024-03-06	[1]	Github
(newgraphenvironment/fpr@f3c021e)				
fs	1.6.3	2023-07-20	[1]	CRAN (R 4.3.0)
generics	0.1.3	2022-07-05	[1]	CRAN (R 4.3.0)
ggplot2	* 3.4.4	2023-10-12	[1]	CRAN (R 4.3.1)
glue	1.7.0	2024-01-09	[1]	CRAN (R 4.3.1)
gtable	0.3.4	2023-08-21	[1]	CRAN (R 4.3.0)
highr	0.10	2022-12-22	[1]	CRAN (R 4.3.2)
hms	1.1.3	2023-03-21	[1]	CRAN (R 4.3.0)
htmltools	0.5.7	2023-11-03	[1]	CRAN (R 4.3.1)
htmlwidgets	1.6.4	2023-12-06	[1]	CRAN (R 4.3.1)
httpuv	1.6.14	2024-01-26	[1]	CRAN (R 4.3.1)
janitor	2.2.0	2023-02-02	[1]	CRAN (R 4.3.0)
jquerylib	0.1.4	2021-04-26	[1]	CRAN (R 4.3.2)
jsonlite	1.8.8	2023-12-04	[1]	CRAN (R 4.3.1)
kableExtra	1.4.0	2024-01-24	[1]	CRAN (R 4.3.1)
KernSmooth	2.23-22	2023-07-10	[2]	CRAN (R 4.3.2)

Session Info

```

later      1.3.2      2023-12-06 [1] CRAN (R 4.3.1)
lifecycle  1.0.4      2023-11-07 [1] CRAN (R 4.3.2)
lubridate  * 1.9.3      2023-09-27 [1] CRAN (R 4.3.1)
magrittr   2.0.3      2022-03-30 [1] CRAN (R 4.3.0)
memoise    2.0.1      2021-11-26 [1] CRAN (R 4.3.2)
mime       0.12       2021-09-28 [1] CRAN (R 4.3.0)
miniUI     0.1.1.1     2018-05-18 [1] CRAN (R 4.3.0)
munsell    0.5.0      2018-06-12 [1] CRAN (R 4.3.0)
pagedown   * 0.20       2022-12-13 [1] CRAN (R 4.3.0)
pak        0.7.1      2023-12-11 [1] CRAN (R 4.3.1)
pillar     1.9.0      2023-03-22 [1] CRAN (R 4.3.0)
pkgbuild   1.4.3      2023-12-10 [1] CRAN (R 4.3.1)
pkgconfig  2.0.3      2019-09-22 [1] CRAN (R 4.3.0)
pkgload    1.3.4      2024-01-16 [1] CRAN (R 4.3.1)
poisutils  0.0.0.9010 2024-01-08 [1] Github
(poissonconsulting/poisutils@8310dc4)
processx   3.8.3      2023-12-10 [1] CRAN (R 4.3.1)
profvis    0.3.8      2023-05-02 [1] CRAN (R 4.3.0)
promises   1.2.1      2023-08-10 [1] CRAN (R 4.3.0)
proxy      0.4-27     2022-06-09 [1] CRAN (R 4.3.0)
ps         1.7.6      2024-01-18 [1] CRAN (R 4.3.1)
purrr      * 1.0.2      2023-08-10 [1] CRAN (R 4.3.0)
R6         2.5.1      2021-08-19 [1] CRAN (R 4.3.2)
rappdirs   0.3.3      2021-01-31 [1] CRAN (R 4.3.0)
Rcpp       1.0.12     2024-01-09 [1] CRAN (R 4.3.1)
readr      * 2.1.5      2024-01-10 [1] CRAN (R 4.3.1)
readxl     1.4.3      2023-07-06 [1] CRAN (R 4.3.0)
remotes    2.4.2.1    2023-07-18 [1] CRAN (R 4.3.0)
rfp        * 0.0.0.9000 2024-02-22 [1] local
rlang      1.1.3      2024-01-10 [1] CRAN (R 4.3.1)
rmarkdown  * 2.26       2024-03-05 [1] CRAN (R 4.3.1)
roxygen2    7.3.1      2024-01-22 [1] CRAN (R 4.3.1)
rstudioapi 0.15.0     2023-07-07 [1] CRAN (R 4.3.0)
sass       0.4.8      2023-12-06 [1] CRAN (R 4.3.1)
scales     1.3.0      2023-11-28 [1] CRAN (R 4.3.1)
servr      0.28       2023-12-19 [1] CRAN (R 4.3.1)
sessioninfo 1.2.2     2021-12-06 [1] CRAN (R 4.3.0)
sf          * 1.0-15     2023-12-18 [1] CRAN (R 4.3.2)
shiny      1.8.0      2023-11-17 [1] CRAN (R 4.3.1)
shrtcts    0.1.2      2024-01-25 [1] Github
(gadenbuie/shrtcts@41051cf)
snakecase  0.11.1     2023-08-27 [1] CRAN (R 4.3.0)
stringi    1.8.3      2023-12-11 [1] CRAN (R 4.3.1)
stringr    * 1.5.1      2023-11-14 [1] CRAN (R 4.3.2)
svglite    2.1.3      2023-12-08 [1] CRAN (R 4.3.1)
systemfonts 1.0.5     2023-10-09 [1] CRAN (R 4.3.1)

```

tidyr	* 1.3.0	2023-01-24	[1]	CRAN (R 4.3.0)
tidyselect	1.2.0	2022-10-10	[1]	CRAN (R 4.3.0)
tidyverse	* 2.0.0	2023-02-22	[1]	CRAN (R 4.3.0)
timechange	0.3.0	2024-01-18	[1]	CRAN (R 4.3.1)
tzdb	0.4.0	2023-05-12	[1]	CRAN (R 4.3.0)
units	0.8-5	2023-11-28	[1]	CRAN (R 4.3.1)
urlchecker	1.0.1	2021-11-30	[1]	CRAN (R 4.3.0)
usethis	* 2.2.2	2023-07-06	[1]	CRAN (R 4.3.0)
utf8	1.2.4	2023-10-22	[1]	CRAN (R 4.3.1)
vctrs	0.6.5	2023-12-01	[1]	CRAN (R 4.3.1)
viridisLite	0.4.2	2023-05-02	[1]	CRAN (R 4.3.0)
vroom	1.6.5	2023-12-05	[1]	CRAN (R 4.3.1)
websocket	1.4.1	2021-08-18	[1]	CRAN (R 4.3.0)
withr	3.0.0	2024-01-16	[1]	CRAN (R 4.3.1)
xfun	0.41	2023-11-01	[1]	CRAN (R 4.3.1)
xml2	1.3.6	2023-12-04	[1]	CRAN (R 4.3.1)
xtable	1.8-4	2019-04-21	[1]	CRAN (R 4.3.0)
yaml	2.3.8	2023-12-11	[1]	CRAN (R 4.3.1)
yesno	0.1.2.9000	2024-01-11	[1]	Github

(poissonconsulting/yesno@2318599)

[1] /Users/airvine/Library/R/arm64/4.3/library
 [2] /Library/Frameworks/R.framework/Versions/4.3-
 arm64/Resources/library
