Discussion: more formal (?) guidelines when submitting data to ExperimentHub

Stephanie Hicks

October 17, 2019

Bioconductor Developer’s Forum

Two questions to that I want to discuss today:

Should we create more formal guidelines for developers

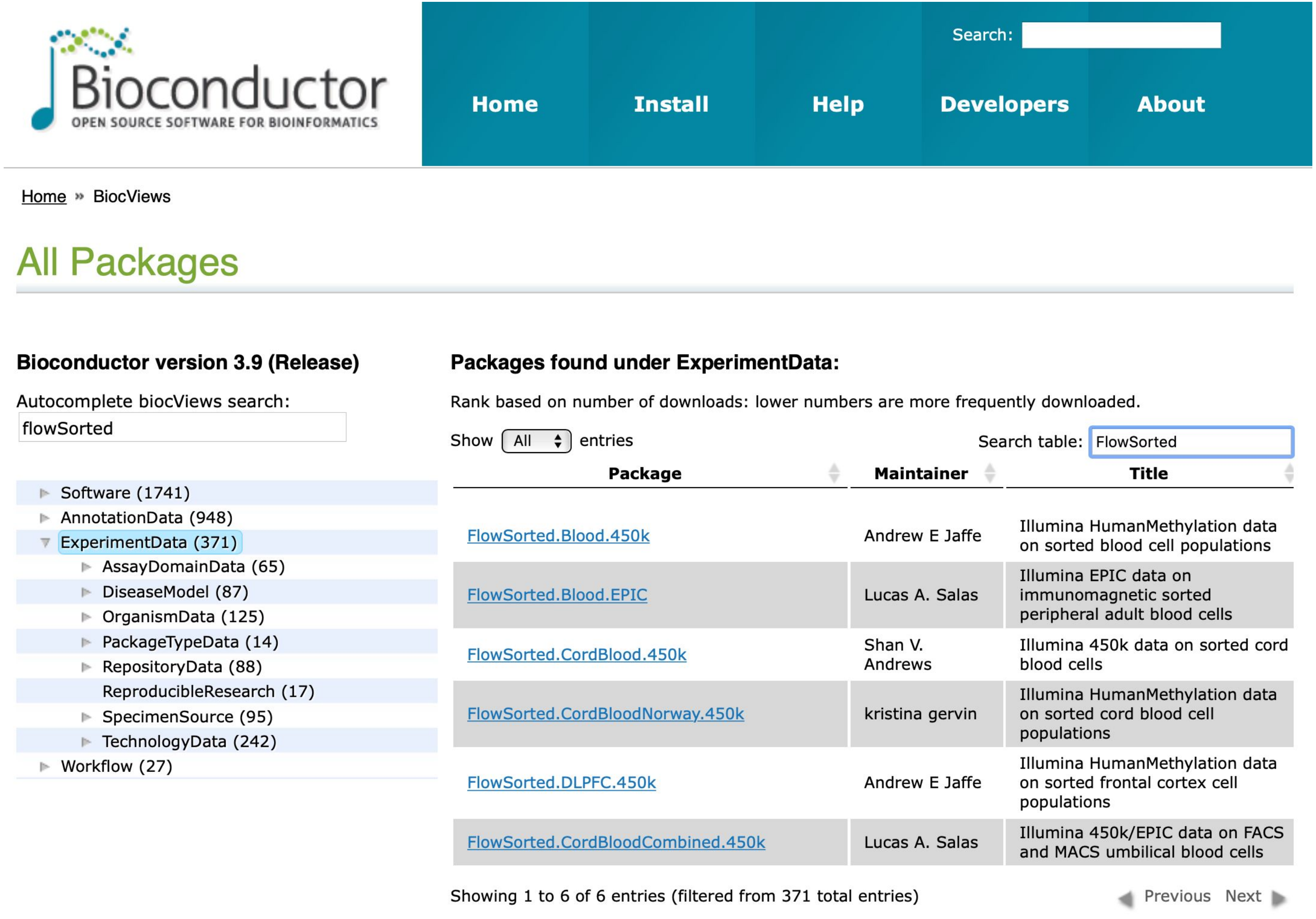
1. On how to **name** ExperimentHub data packages?

2. What format to **store** data in when submitting ExperimentHub data package?

A case study

Flow sorted purified cell types from various blood and brain samples

Currently, all FlowSorted data based on array platforms



Home » BiocViews

All Packages

Bioconductor version 3.9 (Release)

Autocomplete biocViews search: flowSorted

- Software (1741)
- AnnotationData (948)
- ExperimentData (371)
  - AssayDomainData (65)
  - DiseaseModel (87)
  - OrganismData (125)
  - PackageTypeData (14)
  - RepositoryData (88)
  - ReproducibleResearch (17)
  - SpecimenSource (95)
  - TechnologyData (242)
- Workflow (27)

Packages found under ExperimentData:

Rank based on number of downloads: lower numbers are more frequently downloaded.

Show All entries

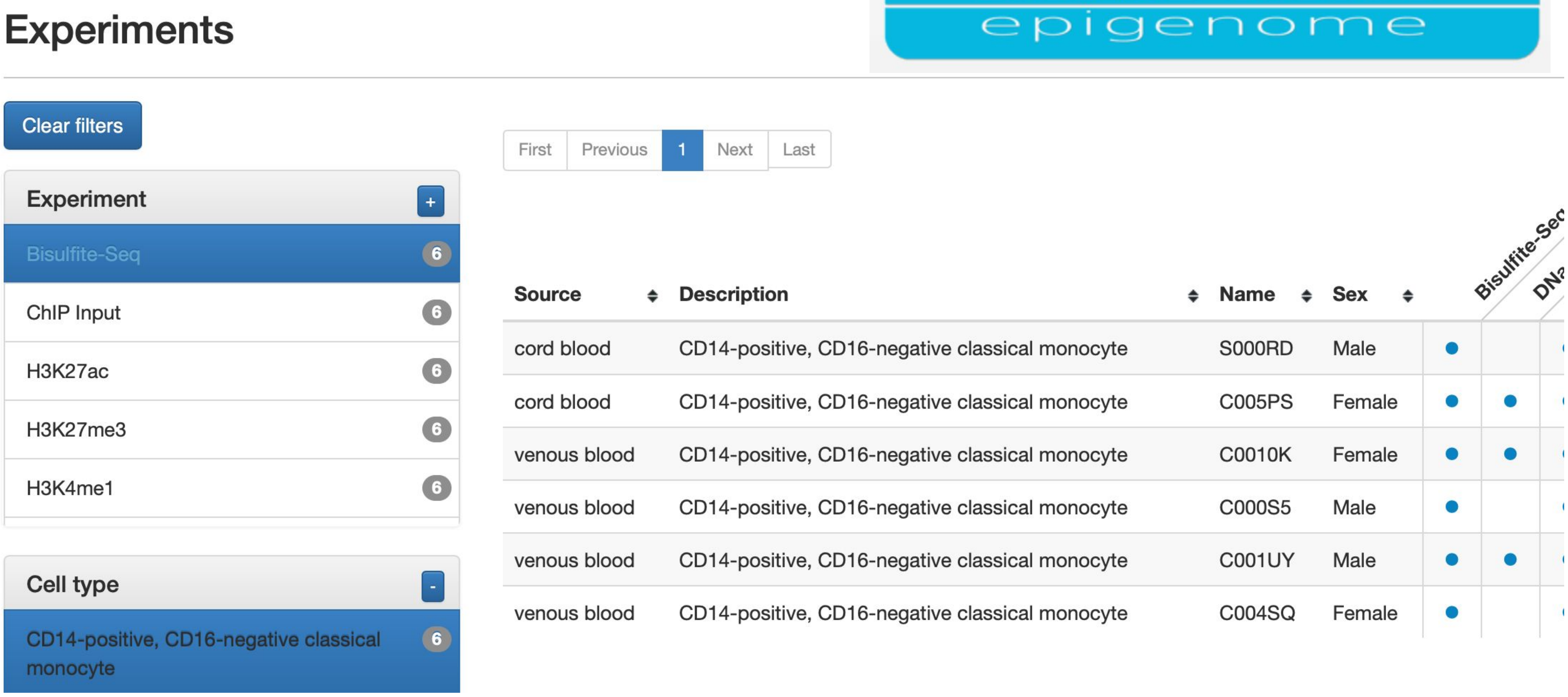
Search table: FlowSorted

Package	Maintainer	Title
FlowSorted.Blood.450k	Andrew E Jaffe	Illumina HumanMethylation data on sorted blood cell populations
FlowSorted.Blood.EPIC	Lucas A. Salas	Illumina EPIC data on immunomagnetic sorted peripheral adult blood cells
FlowSorted.CordBlood.450k	Shan V. Andrews	Illumina 450k data on sorted cord blood cells
FlowSorted.CordBloodNorway.450k	kristina gervin	Illumina HumanMethylation data on sorted cord blood cell populations
FlowSorted.DLPFC.450k	Andrew E Jaffe	Illumina HumanMethylation data on sorted frontal cortex cell populations
FlowSorted.CordBloodCombined.450k	Lucas A. Salas	Illumina 450k/EPIC data on FACS and MACS umbilical blood cells

Showing 1 to 6 of 6 entries (filtered from 371 total entries)

Previous Next

BLUEPRINT epigenome



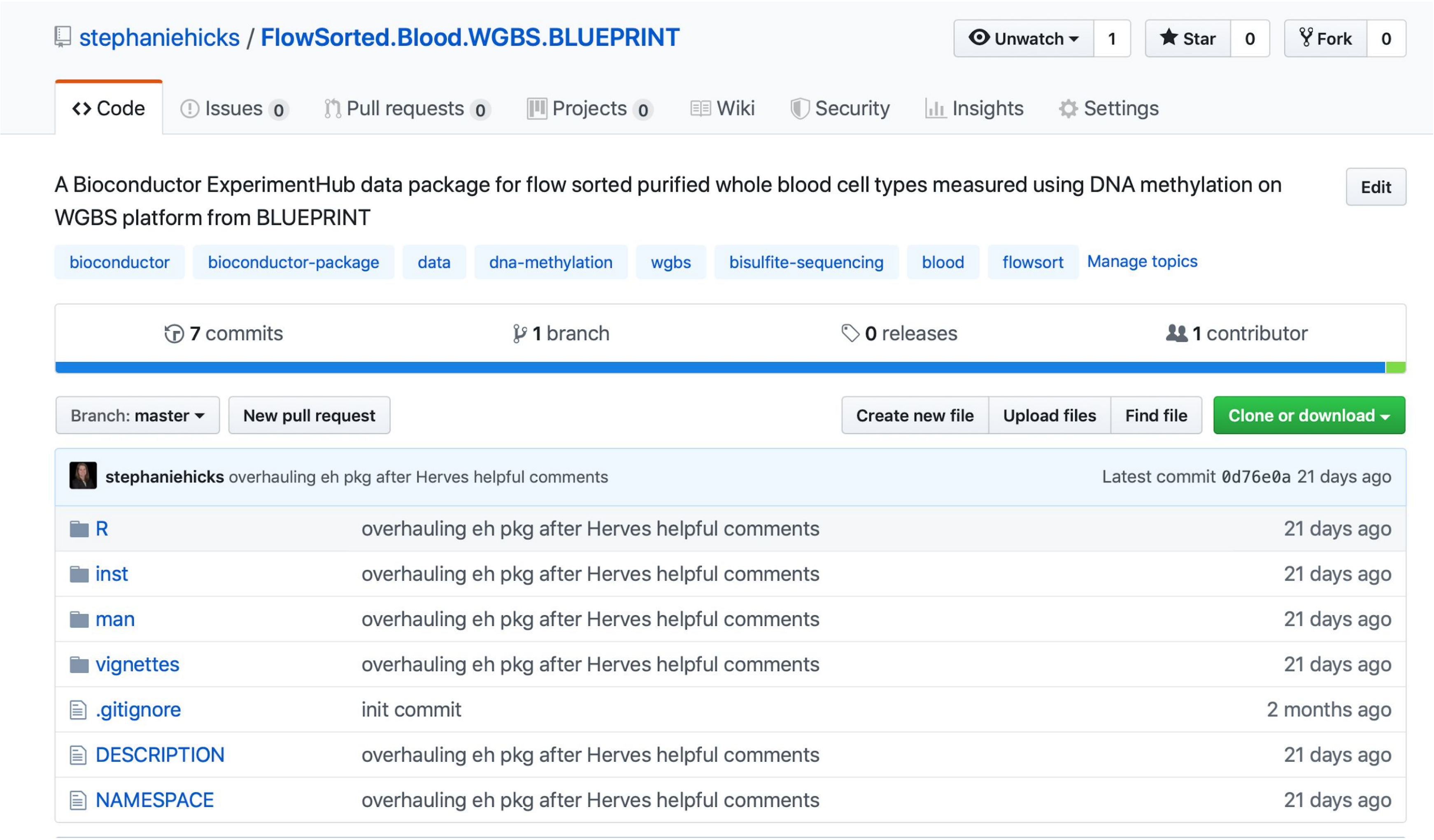
Experiments

Experiment +	
Bisulfite-Seq	6
ChIP Input	6
H3K27ac	6
H3K27me3	6
H3K4me1	6

Cell type -	
CD14-positive, CD16-negative classical monocyte	6

Source	Description	Name	Sex	Bisulfite-Seq	DNa	
cord blood	CD14-positive, CD16-negative classical monocyte	S000RD	Male	●		●
cord blood	CD14-positive, CD16-negative classical monocyte	C005PS	Female	●	●	●
venous blood	CD14-positive, CD16-negative classical monocyte	C0010K	Female	●	●	●
venous blood	CD14-positive, CD16-negative classical monocyte	C000S5	Male	●		●
venous blood	CD14-positive, CD16-negative classical monocyte	C001UY	Male	●	●	●
venous blood	CD14-positive, CD16-negative classical monocyte	C004SQ	Female	●		●

http://dcc.blueprint-epigenome.eu/#/experiments



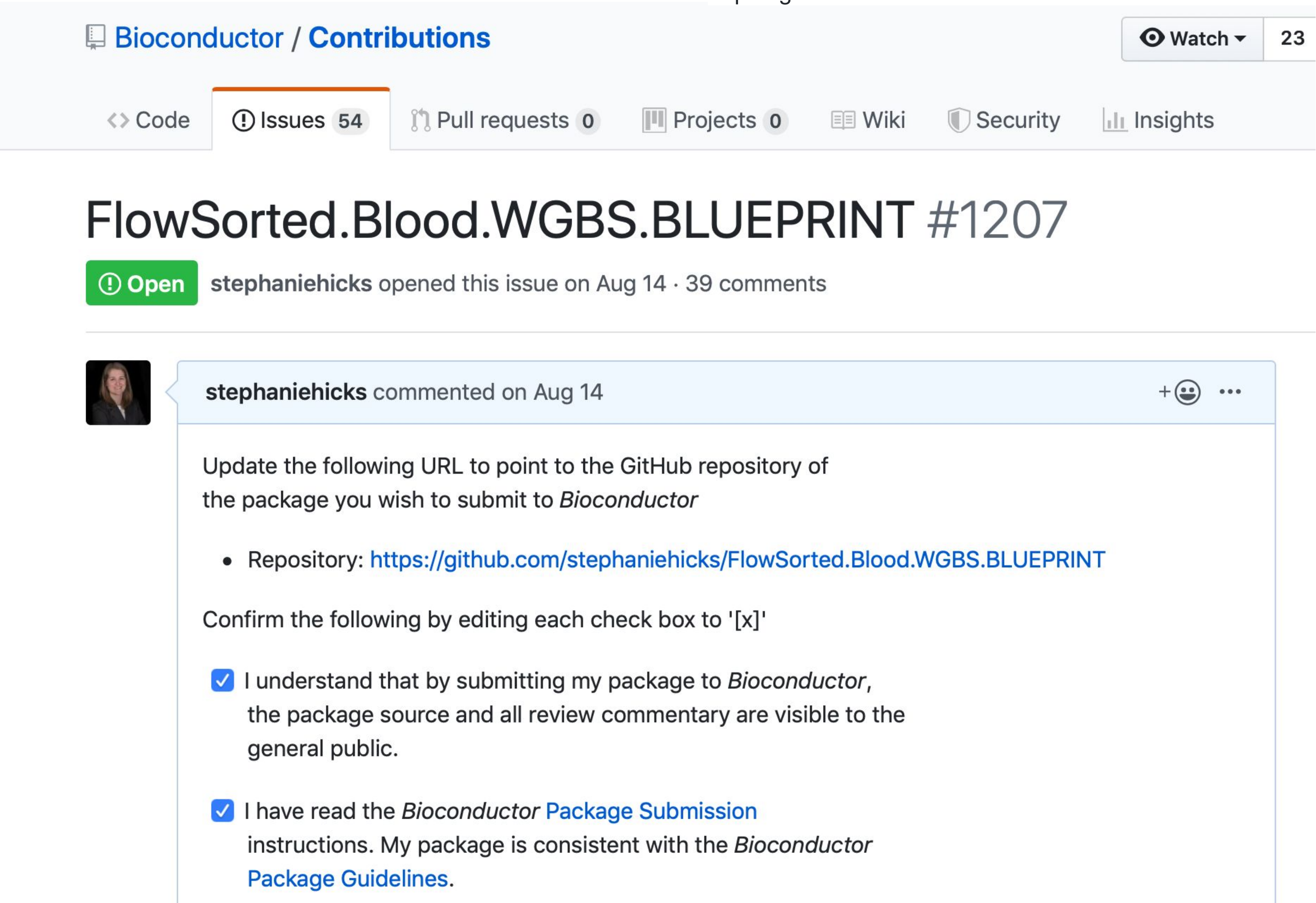
A Bioconductor ExperimentHub data package for flow sorted purified whole blood cell types measured using DNA methylation on WGBS platform from BLUEPRINT

bioconductor · bioconductor-package · data · dna-methylation · wgbs · bisulfite-sequencing · blood · flowsort

stephaniehicks overhauling eh pkg after Herves helpful comments — Latest commit `0d76e0a` 21 days ago

R	overhauling eh pkg after Herves helpful comments	21 days ago
inst	overhauling eh pkg after Herves helpful comments	21 days ago
man	overhauling eh pkg after Herves helpful comments	21 days ago
vignettes	overhauling eh pkg after Herves helpful comments	21 days ago
.gitignore	init commit	2 months ago
DESCRIPTION	overhauling eh pkg after Herves helpful comments	21 days ago
NAMESPACE	overhauling eh pkg after Herves helpful comments	21 days ago

https://github.com/Bioconductor/Contributions/issues/1207#issue-480969976



FlowSorted.Blood.WGBS.BLUEPRINT #1207

stephaniehicks opened this issue on Aug 14 · 39 comments

stephaniehicks commented on Aug 14

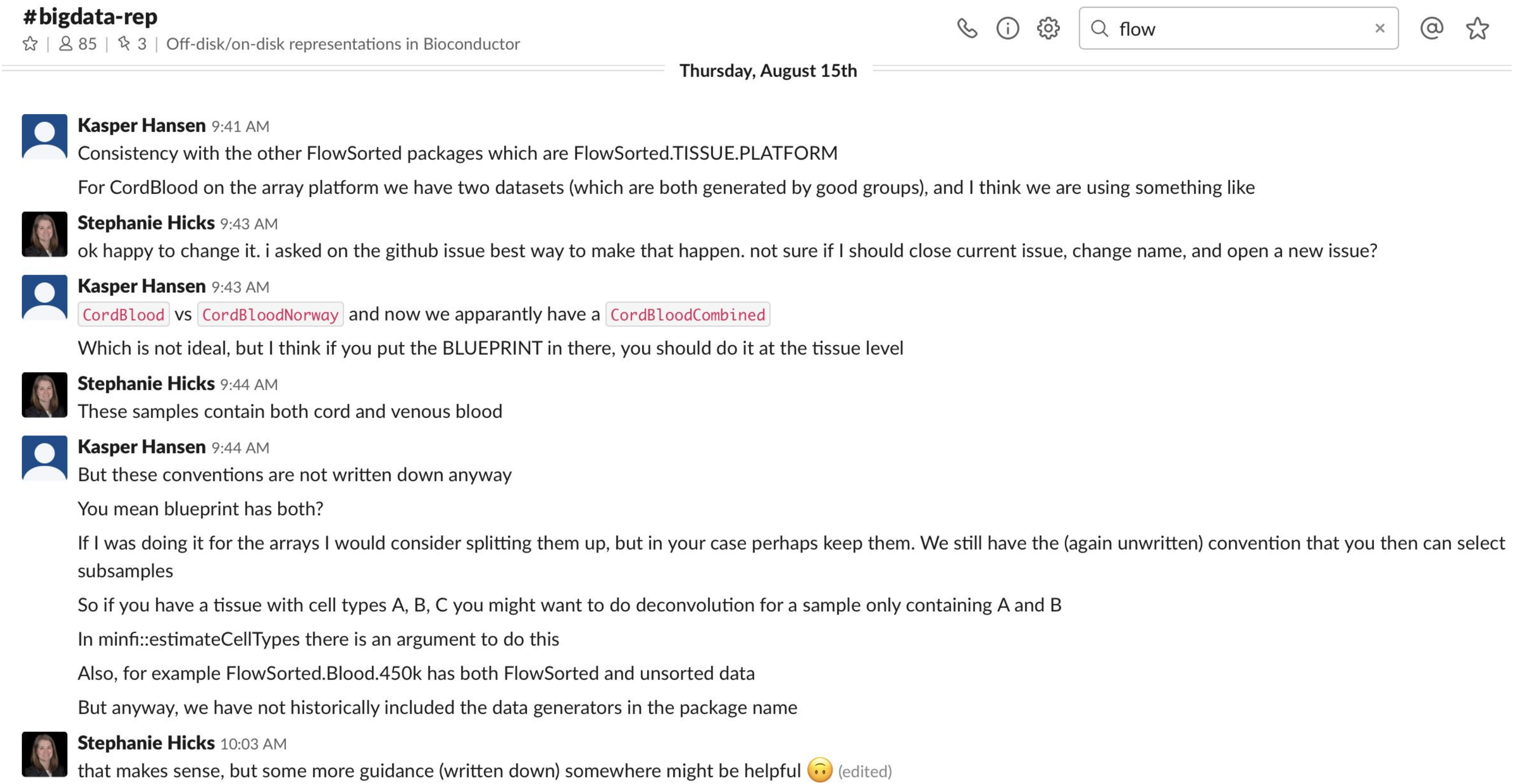
Update the following URL to point to the GitHub repository of the package you wish to submit to *Bioconductor*

- Repository: https://github.com/stephaniehicks/FlowSorted.Blood.WGBS.BLUEPRINT

Confirm the following by editing each check box to '[x]'

☑ I understand that by submitting my package to *Bioconductor*, the package source and all review commentary are visible to the general public.

☑ I have read the *Bioconductor* Package Submission instructions. My package is consistent with the *Bioconductor* Package Guidelines.



#bigdata-rep

☆ | 85 | 3 | Off-disk/on-disk representations in Bioconductor

Thursday, August 15th

Kasper Hansen 9:41 AM

Consistency with the other FlowSorted packages which are FlowSorted.TISSUE.PLATFORM

For CordBlood on the array platform we have two datasets (which are both generated by good groups), and I think we are using something like

Stephanie Hicks 9:43 AM

ok happy to change it. i asked on the github issue best way to make that happen. not sure if I should close current issue, change name, and open a new issue?

Kasper Hansen 9:43 AM

`CordBlood` vs `CordBloodNorway` and now we apparantly have a `CordBloodCombined`

Which is not ideal, but I think if you put the BLUEPRINT in there, you should do it at the tissue level

Stephanie Hicks 9:44 AM

These samples contain both cord and venous blood

Kasper Hansen 9:44 AM

But these conventions are not written down anyway

You mean blueprint has both?

If I was doing it for the arrays I would consider splitting them up, but in your case perhaps keep them. We still have the (again unwritten) convention that you then can select subsamples

So if you have a tissue with cell types A, B, C you might want to do deconvolution for a sample only containing A and B

In minfi::estimateCellTypes there is an argument to do this

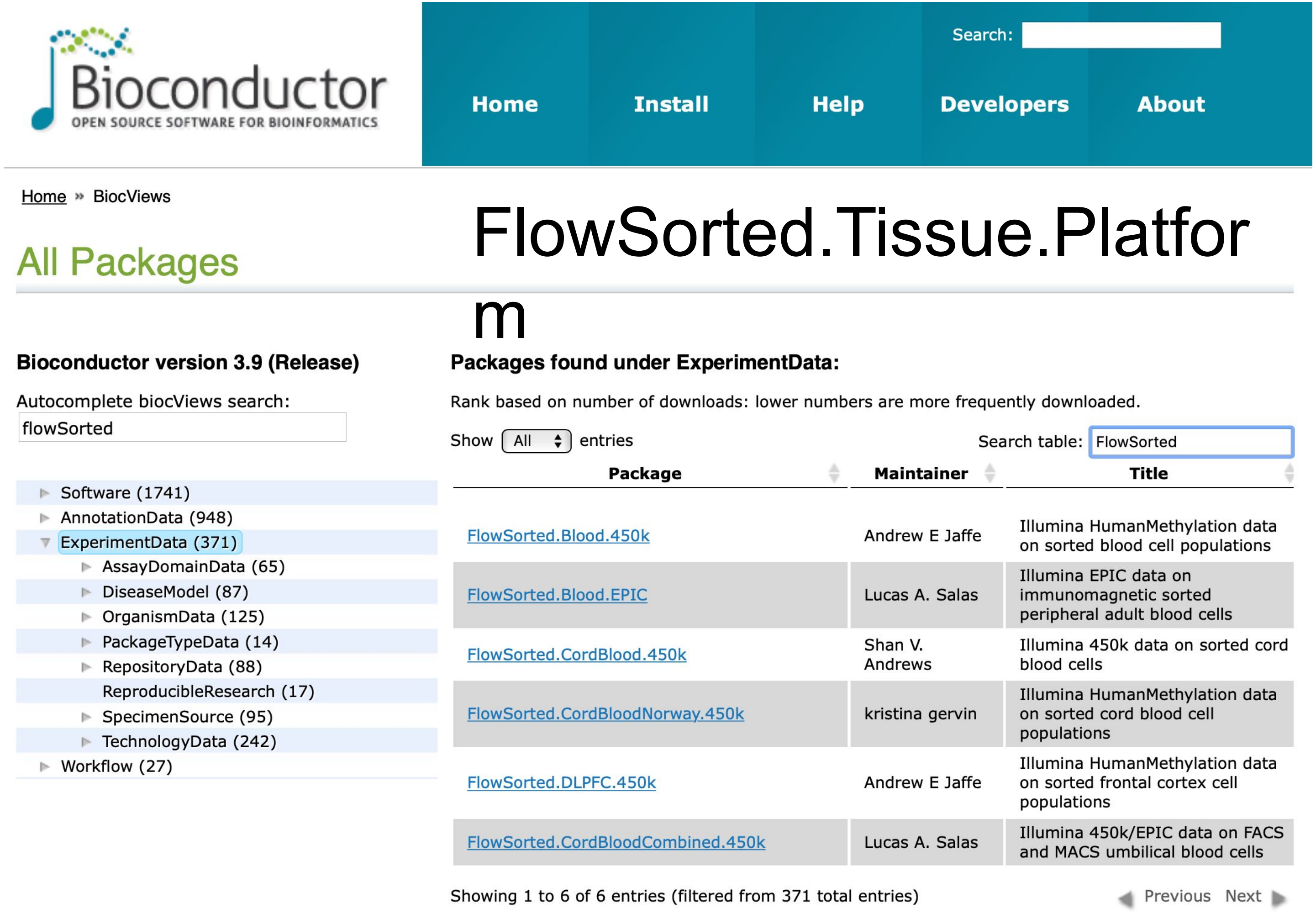
Also, for example FlowSorted.Blood.450k has both FlowSorted and unsorted data

But anyway, we have not historically included the data generators in the package name

Stephanie Hicks 10:03 AM

that makes sense, but some more guidance (written down) somewhere might be helpful 🙃 (edited)

FlowSorted.Tissue.Platform



All Packages

Bioconductor version 3.9 (Release)

Autocomplete biocViews search: flowSorted

- Software (1741)
- AnnotationData (948)
- ExperimentData (371)
  - AssayDomainData (65)
  - DiseaseModel (87)
  - OrganismData (125)
  - PackageTypeData (14)
  - RepositoryData (88)
  - ReproducibleResearch (17)
  - SpecimenSource (95)
  - TechnologyData (242)
- Workflow (27)

Packages found under ExperimentData:

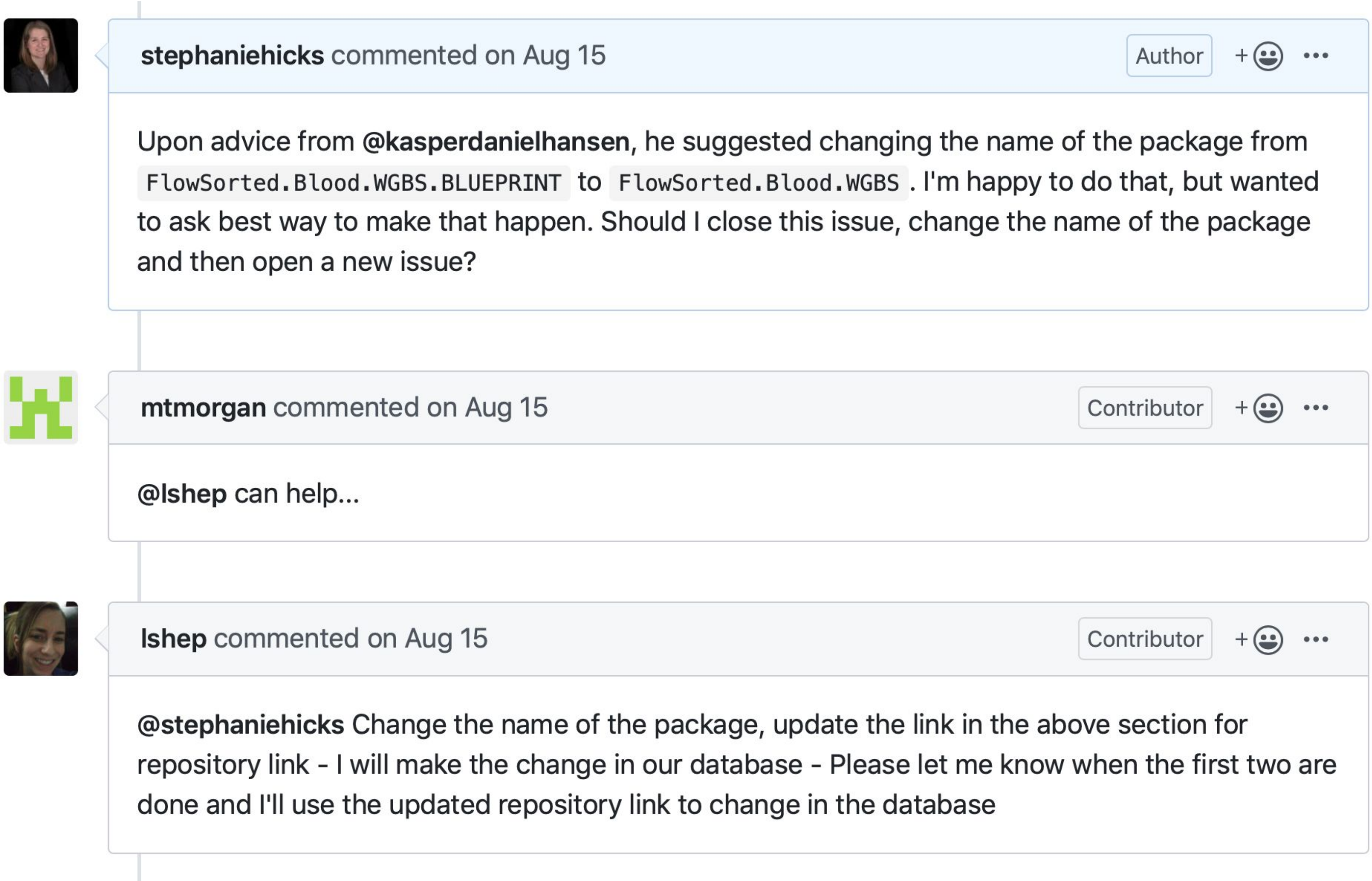
Rank based on number of downloads: lower numbers are more frequently downloaded.

Show All entries

Search table: FlowSorted

Package	Maintainer	Title
FlowSorted.Blood.450k	Andrew E Jaffe	Illumina HumanMethylation data on sorted blood cell populations
FlowSorted.Blood.EPIC	Lucas A. Salas	Illumina EPIC data on immunomagnetic sorted peripheral adult blood cells
FlowSorted.CordBlood.450k	Shan V. Andrews	Illumina 450k data on sorted cord blood cells
FlowSorted.CordBloodNorway.450k	kristina gervin	Illumina HumanMethylation data on sorted cord blood cell populations
FlowSorted.DLPFC.450k	Andrew E Jaffe	Illumina HumanMethylation data on sorted frontal cortex cell populations
FlowSorted.CordBloodCombined.450k	Lucas A. Salas	Illumina 450k/EPIC data on FACS and MACS umbilical blood cells

Showing 1 to 6 of 6 entries (filtered from 371 total entries)



stephaniehicks commented on Aug 15

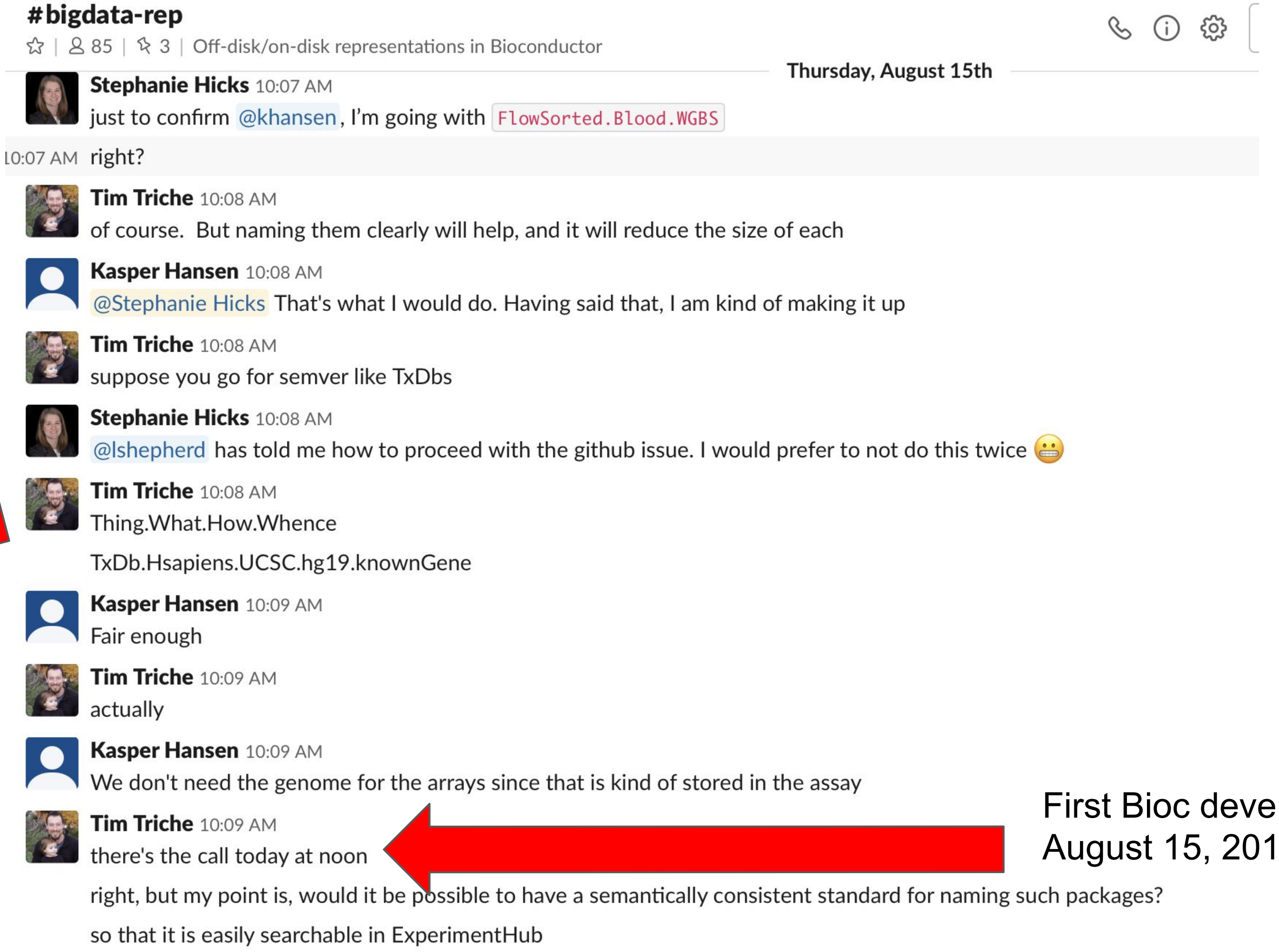
Upon advice from **@kasperdanielhansen**, he suggested changing the name of the package from `FlowSorted.Blood.WGBS.BLUEPRINT` to `FlowSorted.Blood.WGBS`. I'm happy to do that, but wanted to ask best way to make that happen. Should I close this issue, change the name of the package and then open a new issue?

mtmorgan commented on Aug 15

@lshep can help...

lshep commented on Aug 15

@stephaniehicks Change the name of the package, update the link in the above section for repository link - I will make the change in our database - Please let me know when the first two are done and I'll use the updated repository link to change in the database



#bigdata-rep

☆ | 85 | 3 | Off-disk/on-disk representations in Bioconductor

Thursday, August 15th

Stephanie Hicks 10:07 AM
just to confirm @khansen, I'm going with `FlowSorted.Blood.WGBS`

10:07 AM right?

Tim Triche 10:08 AM
of course. But naming them clearly will help, and it will reduce the size of each

Kasper Hansen 10:08 AM
@Stephanie Hicks That's what I would do. Having said that, I am kind of making it up

Tim Triche 10:08 AM
suppose you go for semver like TxDbs

Stephanie Hicks 10:08 AM
@lshepherd has told me how to proceed with the github issue. I would prefer to not do this twice 😬

Tim Triche 10:08 AM
Thing.What.How.Whence

TxDb.Hsapiens.UCSC.hg19.knownGene

Kasper Hansen 10:09 AM
Fair enough

Tim Triche 10:09 AM
actually

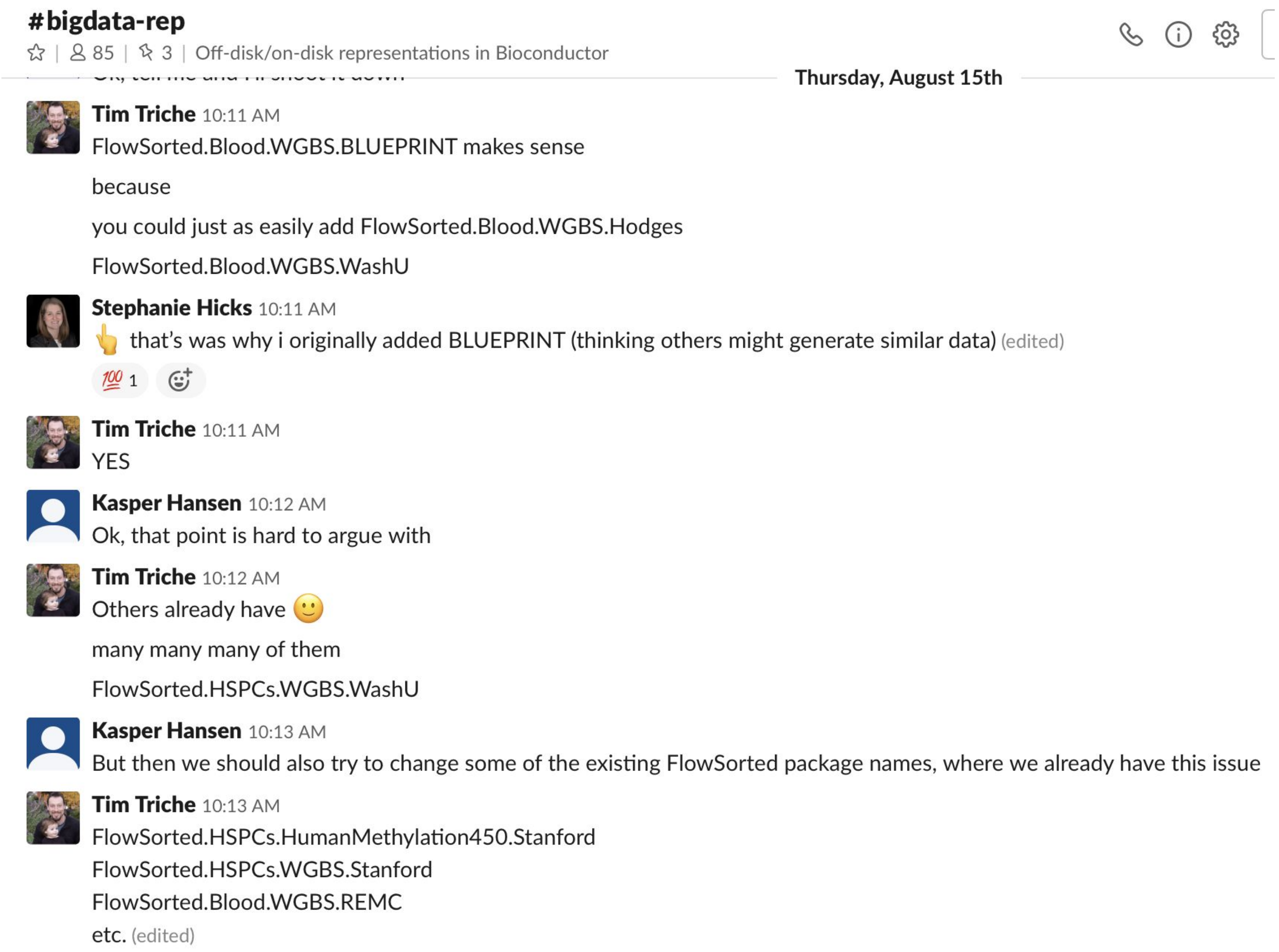
Kasper Hansen 10:09 AM
We don't need the genome for the arrays since that is kind of stored in the assay

Tim Triche 10:09 AM
there's the call today at noon

right, but my point is, would it be possible to have a semantically consistent standard for naming such packages?

so that it is easily searchable in ExperimentHub

First Bioc developer's forum in August 15, 2019



#bigdata-rep

☆ | 85 | 3 | Off-disk/on-disk representations in Bioconductor

Ok, tell me and I'll shoot it down

Thursday, August 15th

Tim Triche 10:11 AM

FlowSorted.Blood.WGBS.BLUEPRINT makes sense

because

you could just as easily add FlowSorted.Blood.WGBS.Hodges

FlowSorted.Blood.WGBS.WashU

Stephanie Hicks 10:11 AM

👆 that's was why i originally added BLUEPRINT (thinking others might generate similar data) (edited)

💯 1

Tim Triche 10:11 AM

YES

Kasper Hansen 10:12 AM

Ok, that point is hard to argue with

Tim Triche 10:12 AM

Others already have 🙂

many many many of them

FlowSorted.HSPCs.WGBS.WashU

Kasper Hansen 10:13 AM

But then we should also try to change some of the existing FlowSorted package names, where we already have this issue

Tim Triche 10:13 AM

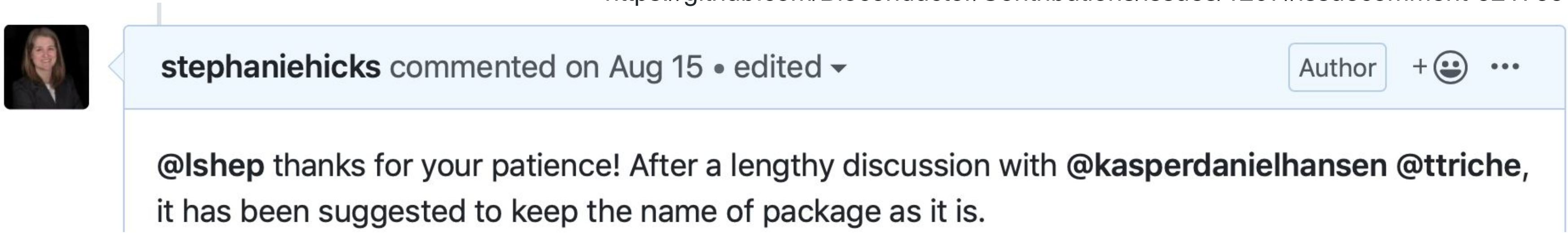
FlowSorted.HSPCs.HumanMethylation450.Stanford
FlowSorted.HSPCs.WGBS.Stanford
FlowSorted.Blood.WGBS.REMC
etc. (edited)

Proposed naming convention:

Thing.TISSUE.PLATFORM.SUPPLIER

If so, then do we need to change already existing ExperimentHub packages? (maybe save discussion of this idea to end?)

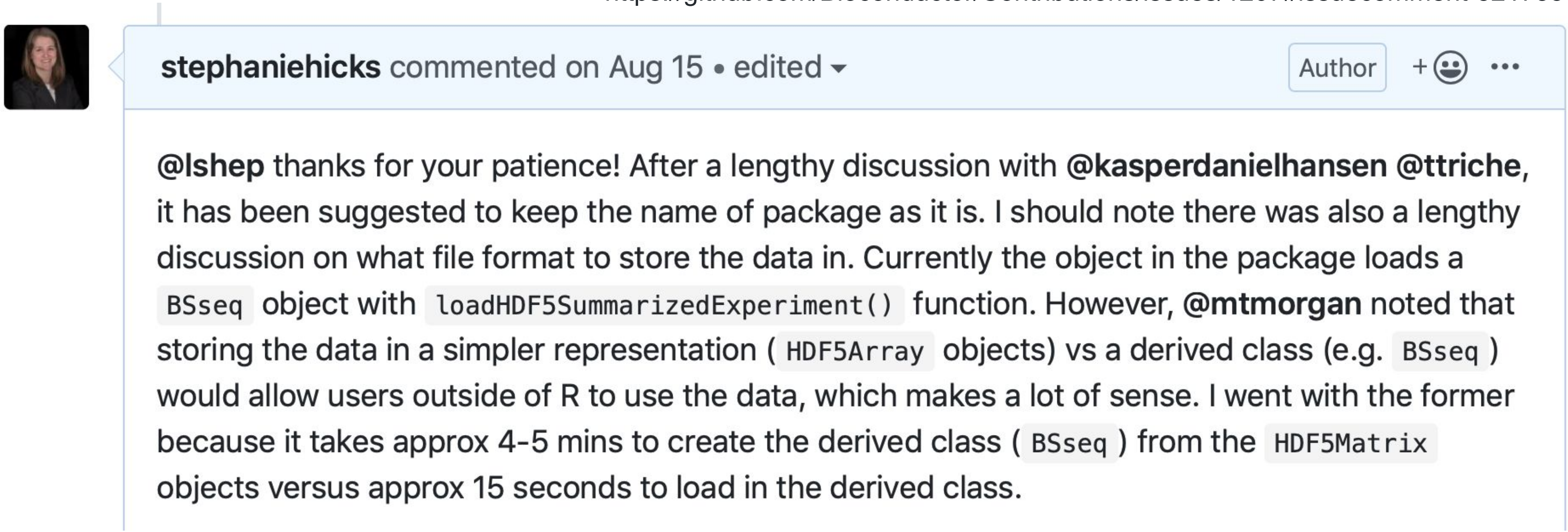
https://github.com/Bioconductor/Contributions/issues/1207#issuecomment-521735531



stephaniehicks commented on Aug 15 • edited

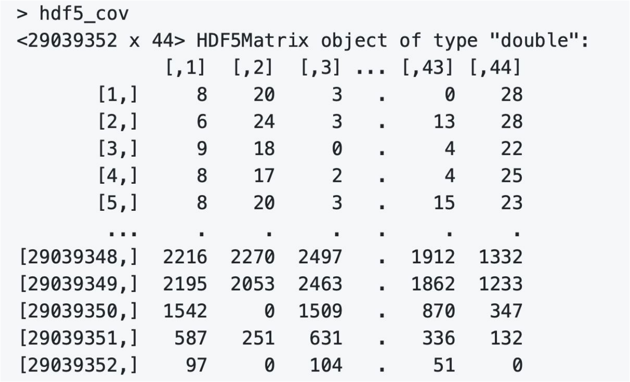
@lshep thanks for your patience! After a lengthy discussion with **@kasperdanielhansen** **@ttriche**, it has been suggested to keep the name of package as it is.

https://github.com/Bioconductor/Contributions/issues/1207#issuecomment-521735531

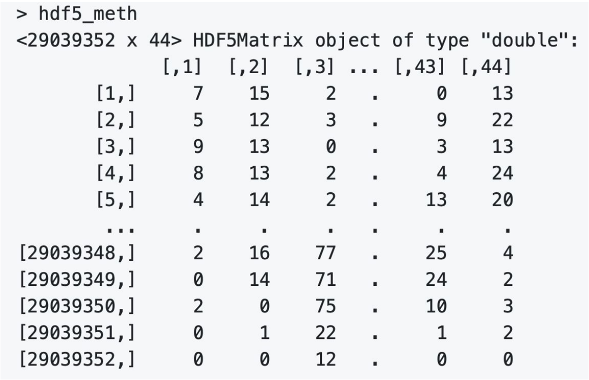


stephaniehicks commented on Aug 15 • edited ▾

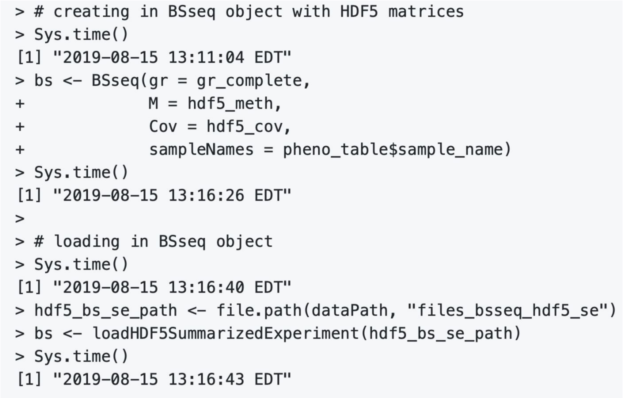
@lshep thanks for your patience! After a lengthy discussion with **@kasperdanielhansen** **@ttriche**, it has been suggested to keep the name of package as it is. I should note there was also a lengthy discussion on what file format to store the data in. Currently the object in the package loads a `BSseq` object with `loadHDF5SummarizedExperiment()` function. However, **@mtmorgan** noted that storing the data in a simpler representation (`HDF5Array` objects) vs a derived class (e.g. `BSseq`) would allow users outside of R to use the data, which makes a lot of sense. I went with the former because it takes approx 4-5 mins to create the derived class (`BSseq`) from the `HDF5Matrix` objects versus approx 15 seconds to load in the derived class.



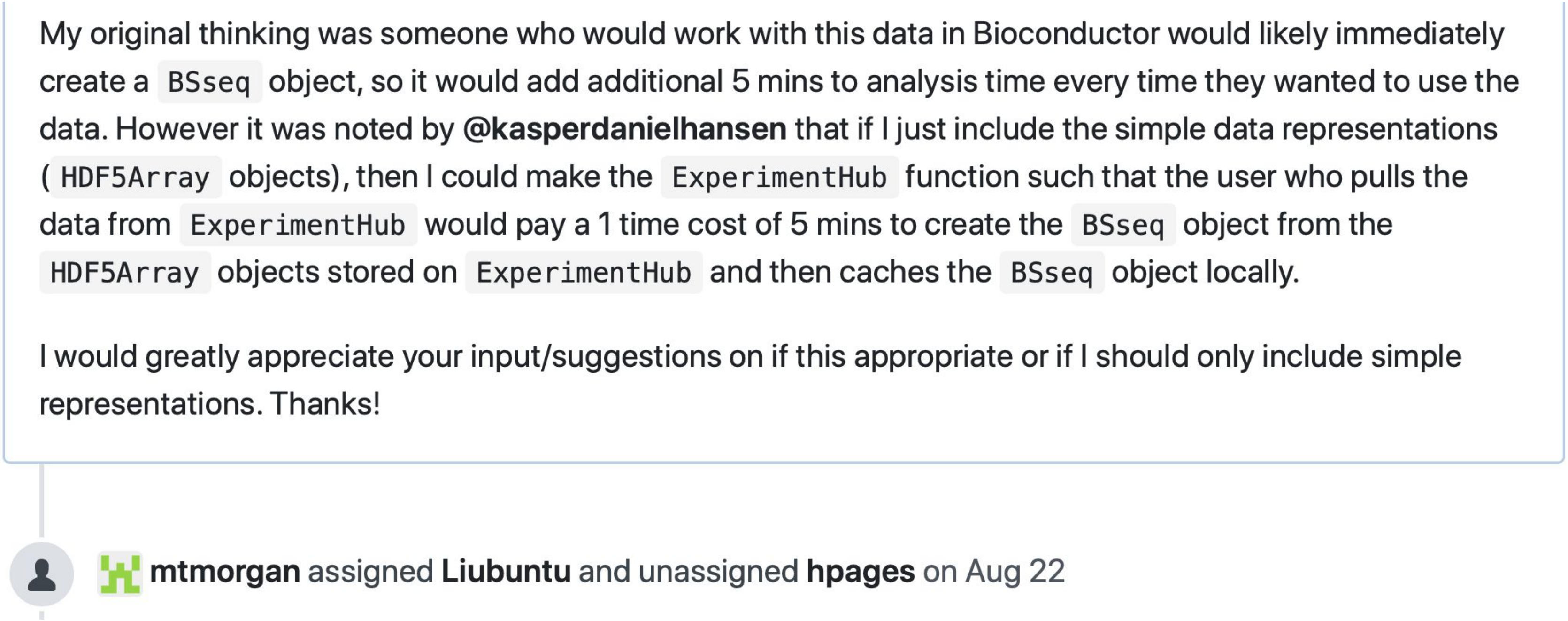
```
> hdf5_cov
<29039352 x 44> HDF5Matrix object of type "double":
              [,1]  [,2]  [,3] ... [,43] [,44]
       [1,]      8    20     3   .     0    28
       [2,]      6    24     3   .    13    28
       [3,]      9    18     0   .     4    22
       [4,]      8    17     2   .     4    25
       [5,]      8    20     3   .    15    23
        ...      .     .     .   .     .     .
[29039348,]   2216  2270  2497   .  1912  1332
[29039349,]   2195  2053  2463   .  1862  1233
[29039350,]   1542     0  1509   .   870   347
[29039351,]    587   251   631   .   336   132
[29039352,]     97     0   104   .    51     0
```



```
> hdf5_meth
<29039352 x 44> HDF5Matrix object of type "double":
              [,1]  [,2]  [,3] ... [,43] [,44]
       [1,]      7    15     2   .     0    13
       [2,]      5    12     3   .     9    22
       [3,]      9    13     0   .     3    13
       [4,]      8    13     2   .     4    24
       [5,]      4    14     2   .    13    20
        ...      .     .     .   .     .     .
[29039348,]      2    16    77   .    25     4
[29039349,]      0    14    71   .    24     2
[29039350,]      2     0    75   .    10     3
[29039351,]      0     1    22   .     1     2
[29039352,]      0     0    12   .     0     0
```



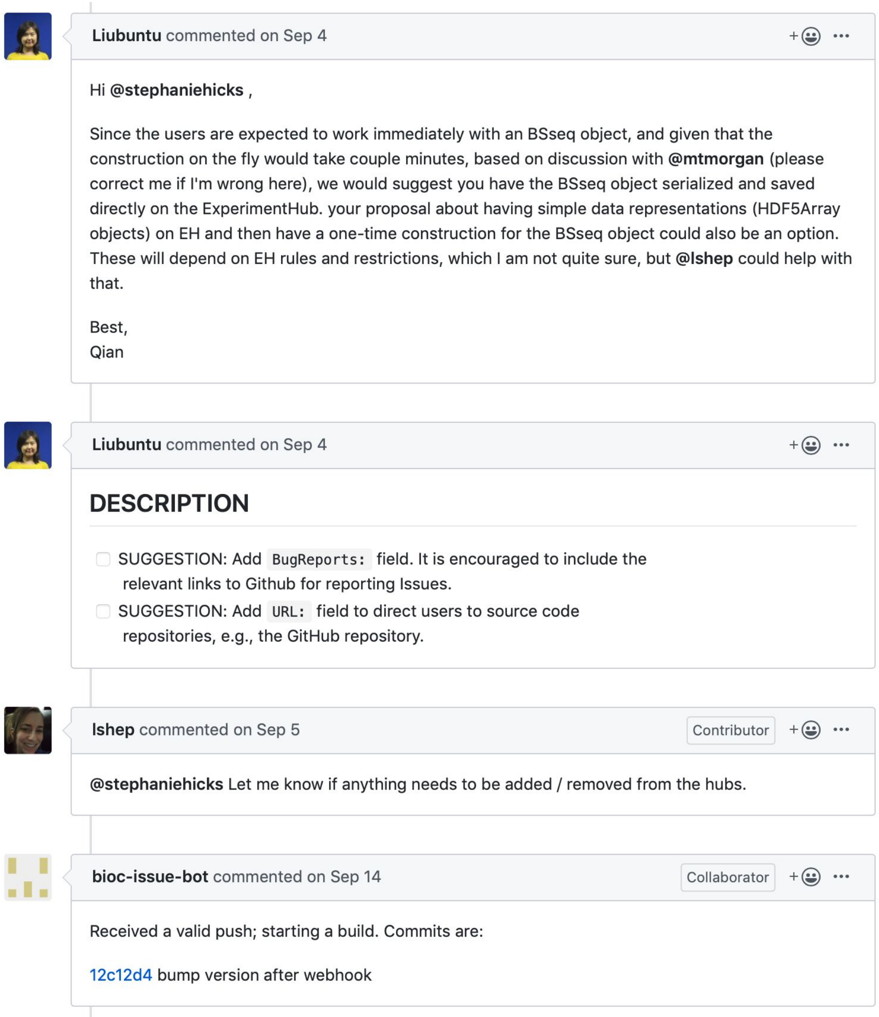
```
> # creating in BSseq object with HDF5 matrices
> Sys.time()
[1] "2019-08-15 13:11:04 EDT"
> bs <- BSseq(gr = gr_complete,
+             M = hdf5_meth,
+             Cov = hdf5_cov,
+             sampleNames = pheno_table$sample_name)
> Sys.time()
[1] "2019-08-15 13:16:26 EDT"
>
> # loading in BSseq object
> Sys.time()
[1] "2019-08-15 13:16:40 EDT"
> hdf5_bs_se_path <- file.path(dataPath, "files_bsseq_hdf5_se")
> bs <- loadHDF5SummarizedExperiment(hdf5_bs_se_path)
> Sys.time()
[1] "2019-08-15 13:16:43 EDT"
```



My original thinking was someone who would work with this data in Bioconductor would likely immediately create a `BSseq` object, so it would add additional 5 mins to analysis time every time they wanted to use the data. However it was noted by **@kasperdanielhansen** that if I just include the simple data representations (`HDF5Array` objects), then I could make the `ExperimentHub` function such that the user who pulls the data from `ExperimentHub` would pay a 1 time cost of 5 mins to create the `BSseq` object from the `HDF5Array` objects stored on `ExperimentHub` and then caches the `BSseq` object locally.

I would greatly appreciate your input/suggestions on if this appropriate or if I should only include simple representations. Thanks!

mtmorgan assigned **Liubuntu** and unassigned **hpages** on Aug 22



Liubuntu commented on Sep 4

Hi **@stephaniehicks** ,

Since the users are expected to work immediately with an BSseq object, and given that the construction on the fly would take couple minutes, based on discussion with **@mtmorgan** (please correct me if I'm wrong here), we would suggest you have the BSseq object serialized and saved directly on the ExperimentHub. your proposal about having simple data representations (HDF5Array objects) on EH and then have a one-time construction for the BSseq object could also be an option. These will depend on EH rules and restrictions, which I am not quite sure, but **@lshep** could help with that.

Best,
Qian

Liubuntu commented on Sep 4

DESCRIPTION

- □ SUGGESTION: Add `BugReports:` field. It is encouraged to include the relevant links to Github for reporting Issues.
- □ SUGGESTION: Add `URL:` field to direct users to source code repositories, e.g., the GitHub repository.

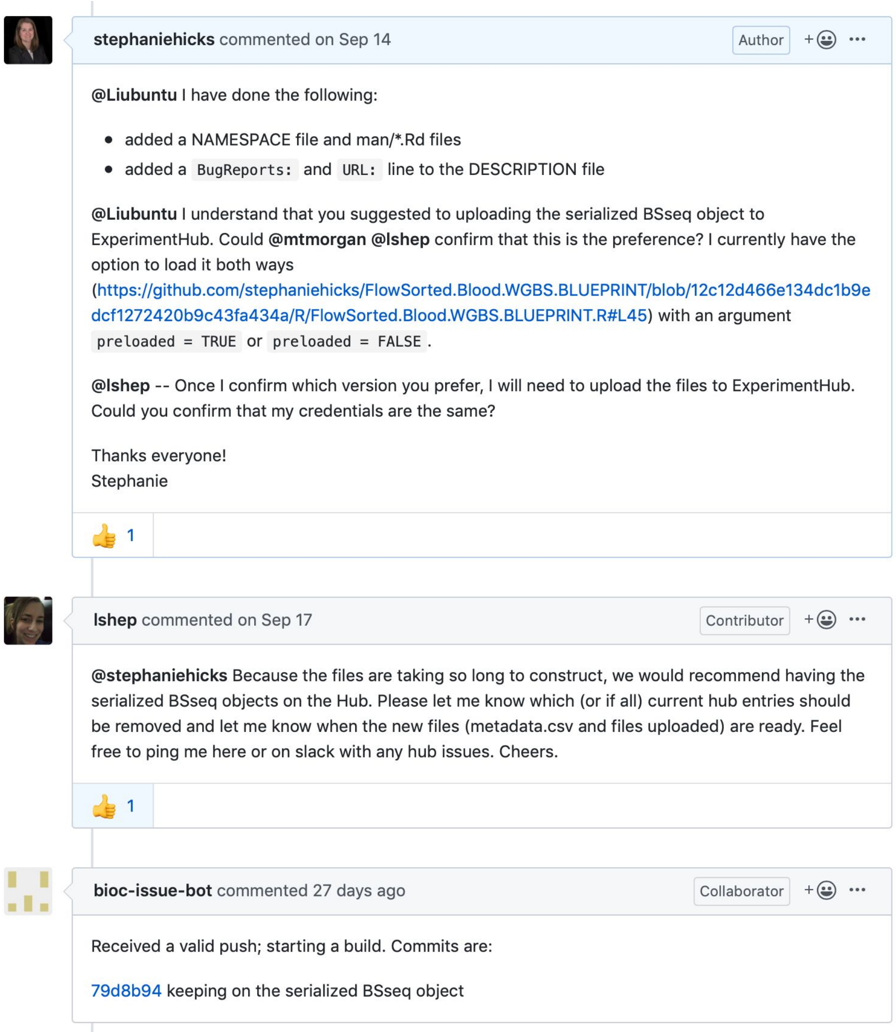
lshep commented on Sep 5 Contributor

@stephaniehicks Let me know if anything needs to be added / removed from the hubs.

bioc-issue-bot commented on Sep 14 Collaborator

Received a valid push; starting a build. Commits are:

12c12d4 bump version after webhook



stephaniehicks commented on Sep 14 Author

@Liubuntu I have done the following:

- added a NAMESPACE file and man/*.Rd files
- added a `BugReports:` and `URL:` line to the DESCRIPTION file

@Liubuntu I understand that you suggested to uploading the serialized BSseq object to ExperimentHub. Could **@mtmorgan** **@lshep** confirm that this is the preference? I currently have the option to load it both ways (https://github.com/stephaniehicks/FlowSorted.Blood.WGBS.BLUEPRINT/blob/12c12d466e134dc1b9edcf1272420b9c43fa434a/R/FlowSorted.Blood.WGBS.BLUEPRINT.R#L45) with an argument `preloaded = TRUE` or `preloaded = FALSE` .

@lshep -- Once I confirm which version you prefer, I will need to upload the files to ExperimentHub. Could you confirm that my credentials are the same?

Thanks everyone!
Stephanie

👍 1

lshep commented on Sep 17 Contributor

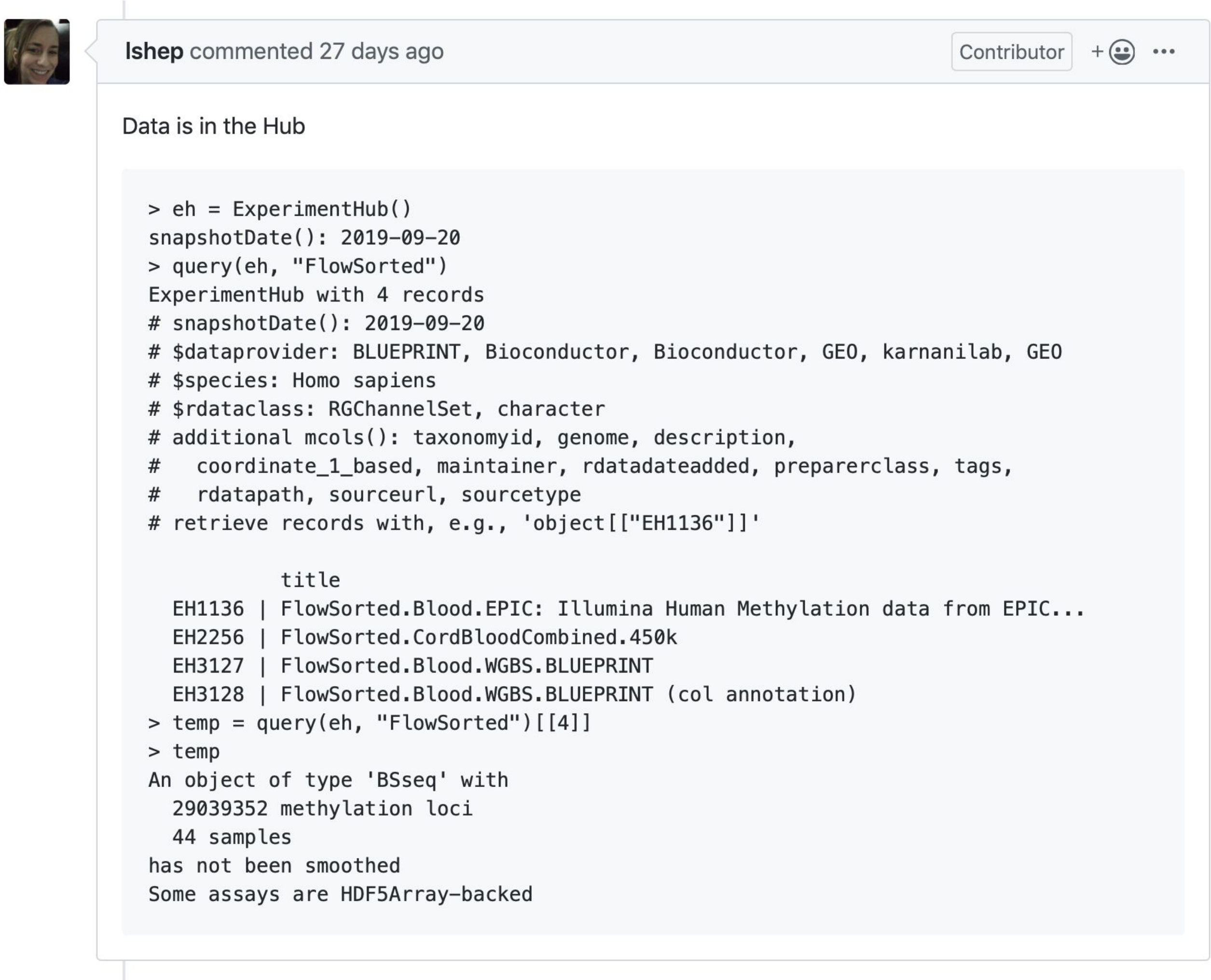
@stephaniehicks Because the files are taking so long to construct, we would recommend having the serialized BSseq objects on the Hub. Please let me know which (or if all) current hub entries should be removed and let me know when the new files (metadata.csv and files uploaded) are ready. Feel free to ping me here or on slack with any hub issues. Cheers.

👍 1

bioc-issue-bot commented 27 days ago Collaborator

Received a valid push; starting a build. Commits are:

79d8b94 keeping on the serialized BSseq object

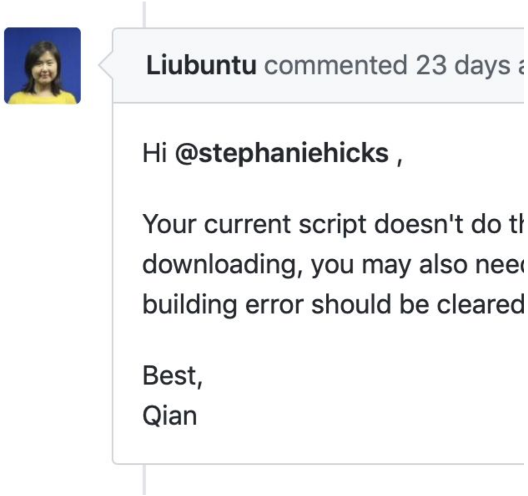


lshep commented 27 days ago

Data is in the Hub

```
> eh = ExperimentHub()
snapshotDate(): 2019-09-20
> query(eh, "FlowSorted")
ExperimentHub with 4 records
# snapshotDate(): 2019-09-20
# $dataprovider: BLUEPRINT, Bioconductor, Bioconductor, GEO, karnanilab, GEO
# $species: Homo sapiens
# $rdataclass: RGChannelSet, character
# additional mcols(): taxonomyid, genome, description,
#   coordinate_1_based, maintainer, rdatadateadded, preparerclass, tags,
#   rdatapath, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["EH1136"]]'

           title
  EH1136 | FlowSorted.Blood.EPIC: Illumina Human Methylation data from EPIC...
  EH2256 | FlowSorted.CordBloodCombined.450k
  EH3127 | FlowSorted.Blood.WGBS.BLUEPRINT
  EH3128 | FlowSorted.Blood.WGBS.BLUEPRINT (col annotation)
> temp = query(eh, "FlowSorted")[[4]]
> temp
An object of type 'BSseq' with
  29039352 methylation loci
  44 samples
has not been smoothed
Some assays are HDF5Array-backed
```

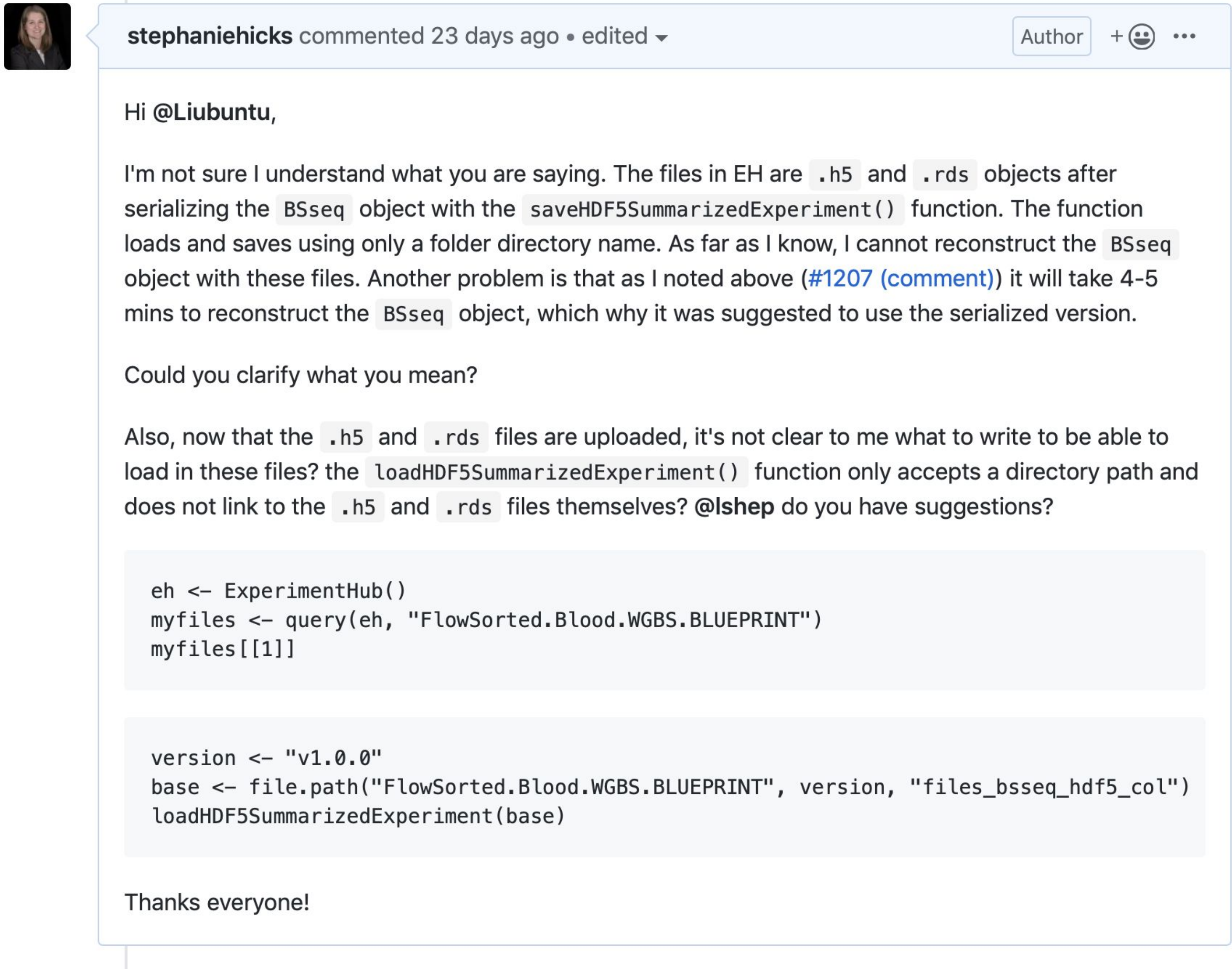


Liubuntu commented 23 days a

Hi **@stephaniehicks** ,

Your current script doesn't do tl
downloading, you may also neec
building error should be cleared

Best,
Qian



stephaniehicks commented 23 days ago • edited

Hi **@Liubuntu**,

I'm not sure I understand what you are saying. The files in EH are `.h5` and `.rds` objects after serializing the `BSseq` object with the `saveHDF5SummarizedExperiment()` function. The function loads and saves using only a folder directory name. As far as I know, I cannot reconstruct the `BSseq` object with these files. Another problem is that as I noted above (#1207 (comment)) it will take 4-5 mins to reconstruct the `BSseq` object, which why it was suggested to use the serialized version.

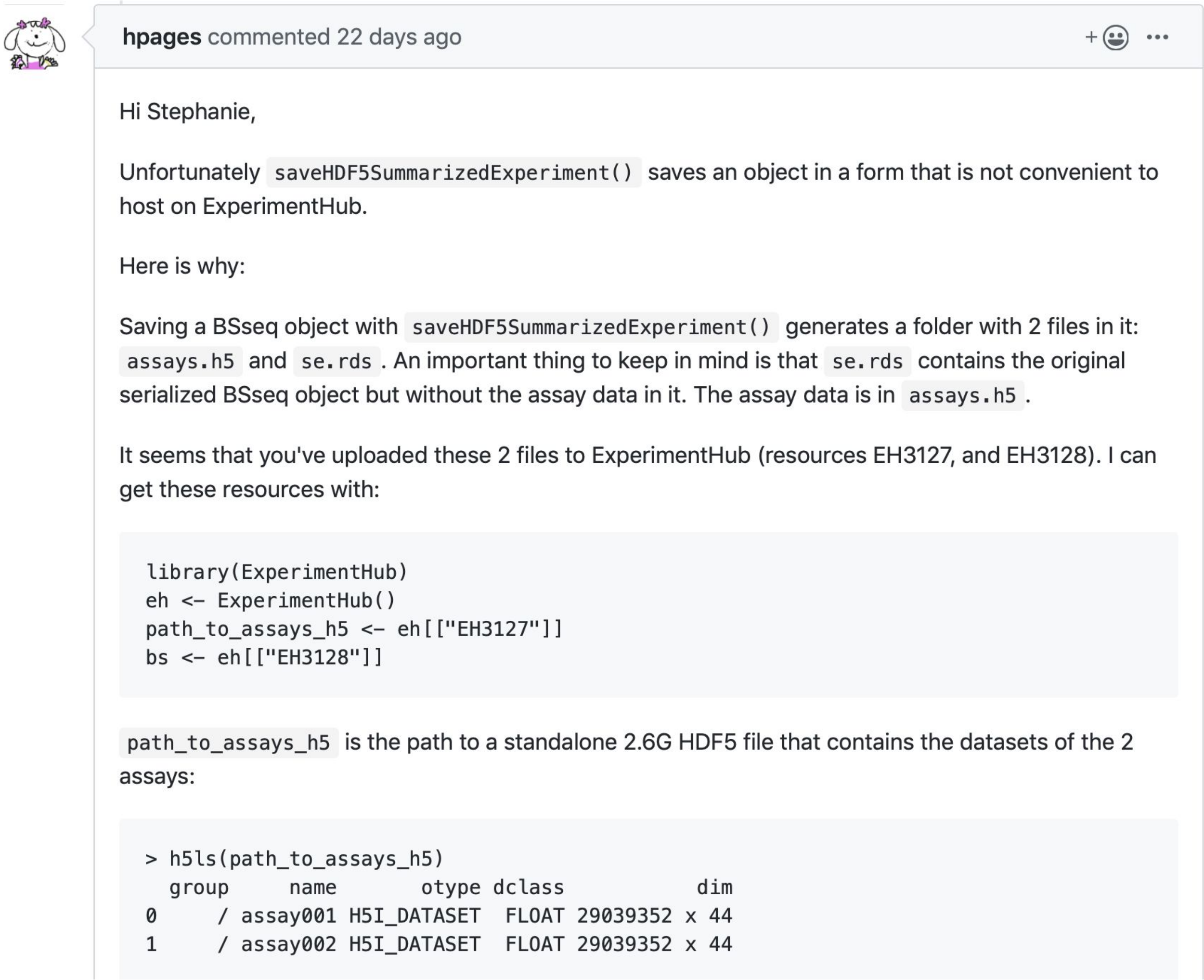
Could you clarify what you mean?

Also, now that the `.h5` and `.rds` files are uploaded, it's not clear to me what to write to be able to load in these files? the `loadHDF5SummarizedExperiment()` function only accepts a directory path and does not link to the `.h5` and `.rds` files themselves? **@lshep** do you have suggestions?

```
eh <- ExperimentHub()
myfiles <- query(eh, "FlowSorted.Blood.WGBS.BLUEPRINT")
myfiles[[1]]
```

```
version <- "v1.0.0"
base <- file.path("FlowSorted.Blood.WGBS.BLUEPRINT", version, "files_bsseq_hdf5_col")
loadHDF5SummarizedExperiment(base)
```

Thanks everyone!



hpages commented 22 days ago

Hi Stephanie,

Unfortunately `saveHDF5SummarizedExperiment()` saves an object in a form that is not convenient to host on ExperimentHub.

Here is why:

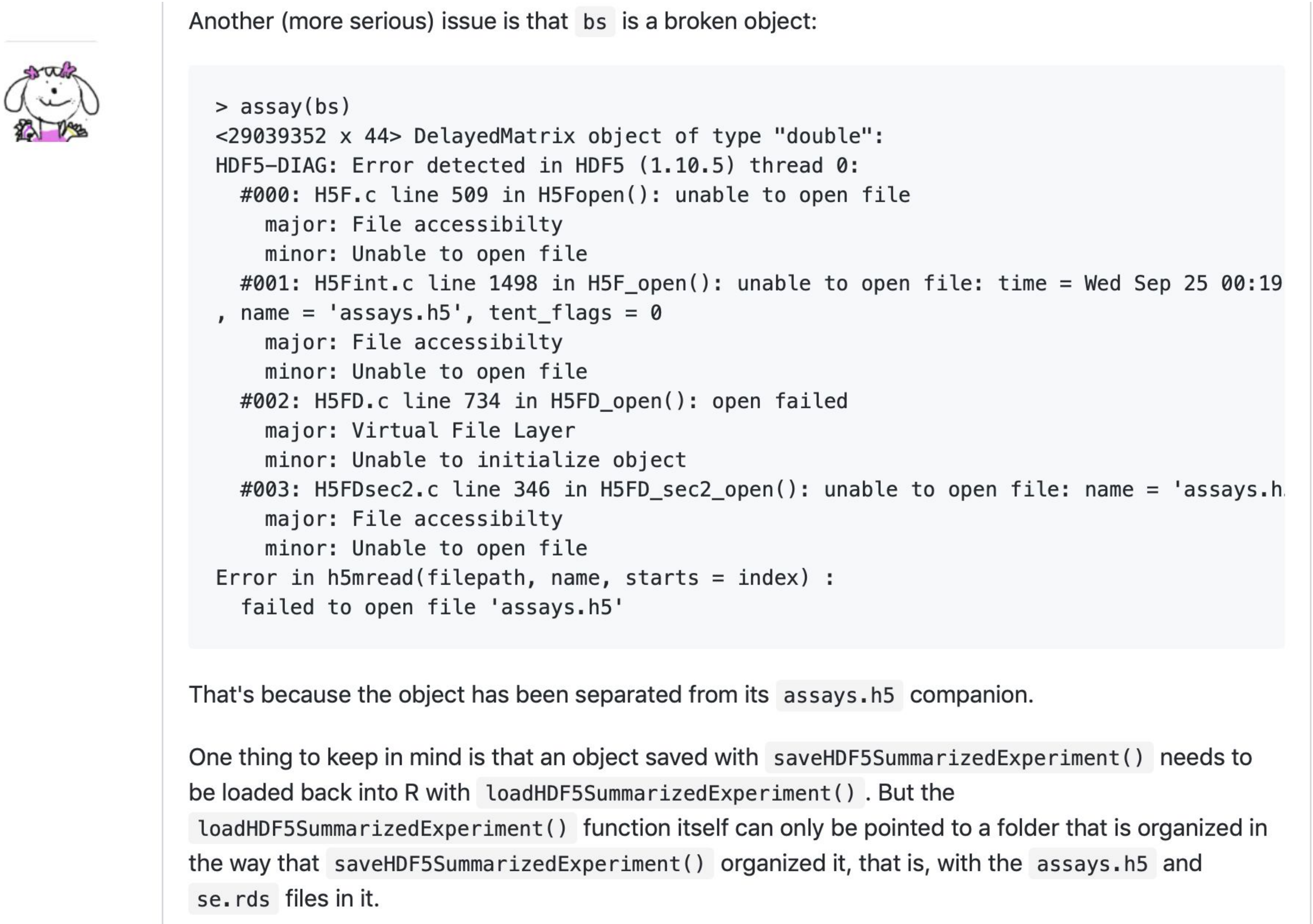
Saving a BSseq object with `saveHDF5SummarizedExperiment()` generates a folder with 2 files in it: `assays.h5` and `se.rds`. An important thing to keep in mind is that `se.rds` contains the original serialized BSseq object but without the assay data in it. The assay data is in `assays.h5`.

It seems that you've uploaded these 2 files to ExperimentHub (resources EH3127, and EH3128). I can get these resources with:

```
library(ExperimentHub)
eh <- ExperimentHub()
path_to_assays_h5 <- eh[["EH3127"]]
bs <- eh[["EH3128"]]
```

`path_to_assays_h5` is the path to a standalone 2.6G HDF5 file that contains the datasets of the 2 assays:

```
> h5ls(path_to_assays_h5)
  group     name       otype dclass            dim
0     / assay001 H5I_DATASET  FLOAT 29039352 x 44
1     / assay002 H5I_DATASET  FLOAT 29039352 x 44
```

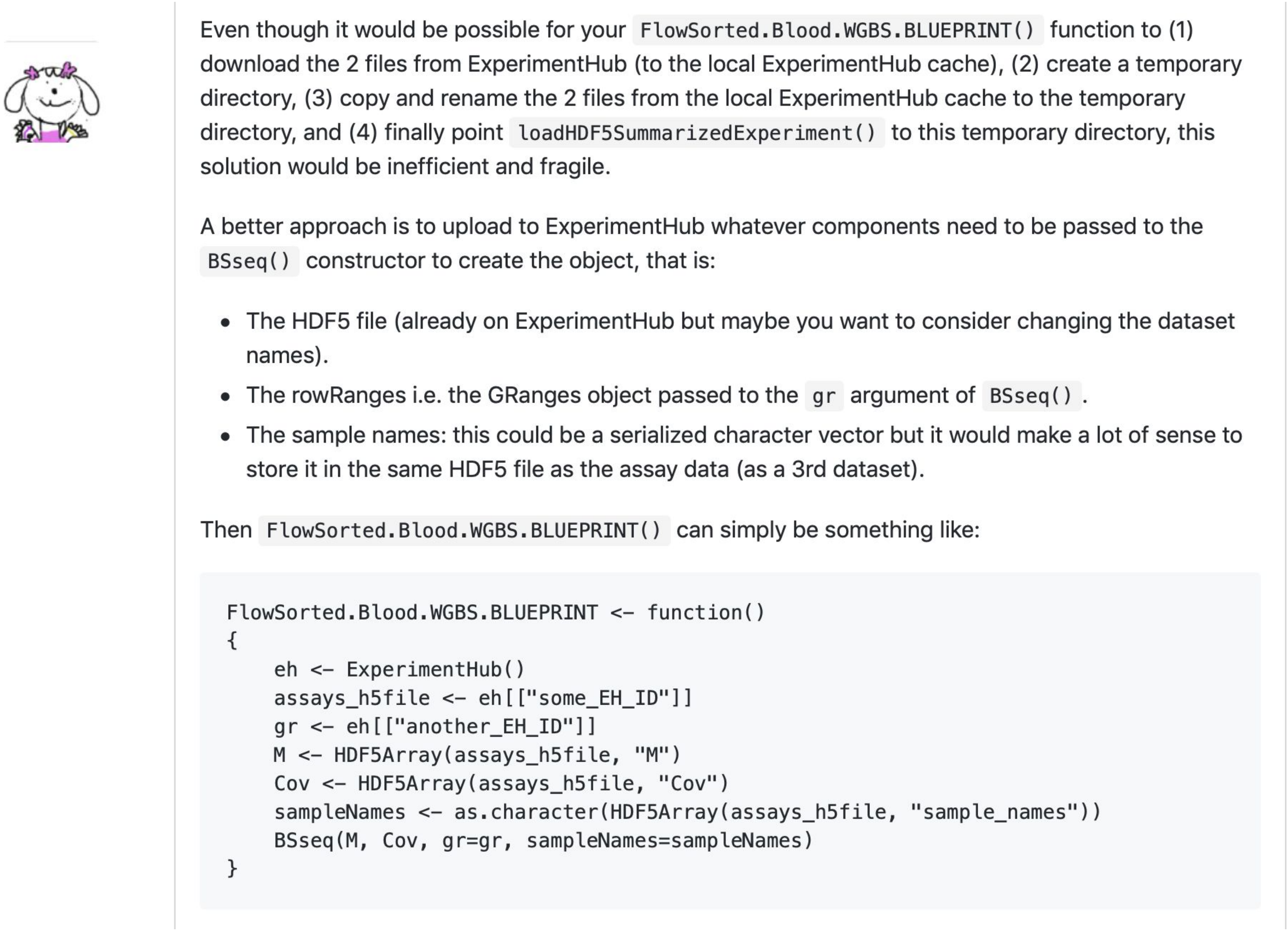


Another (more serious) issue is that `bs` is a broken object:

```
> assay(bs)
<29039352 x 44> DelayedMatrix object of type "double":
HDF5-DIAG: Error detected in HDF5 (1.10.5) thread 0:
  #000: H5F.c line 509 in H5Fopen(): unable to open file
    major: File accessibilty
    minor: Unable to open file
  #001: H5Fint.c line 1498 in H5F_open(): unable to open file: time = Wed Sep 25 00:19
, name = 'assays.h5', tent_flags = 0
    major: File accessibilty
    minor: Unable to open file
  #002: H5FD.c line 734 in H5FD_open(): open failed
    major: Virtual File Layer
    minor: Unable to initialize object
  #003: H5FDsec2.c line 346 in H5FD_sec2_open(): unable to open file: name = 'assays.h
    major: File accessibilty
    minor: Unable to open file
Error in h5mread(filepath, name, starts = index) :
  failed to open file 'assays.h5'
```

That's because the object has been separated from its `assays.h5` companion.

One thing to keep in mind is that an object saved with `saveHDF5SummarizedExperiment()` needs to be loaded back into R with `loadHDF5SummarizedExperiment()`. But the `loadHDF5SummarizedExperiment()` function itself can only be pointed to a folder that is organized in the way that `saveHDF5SummarizedExperiment()` organized it, that is, with the `assays.h5` and `se.rds` files in it.



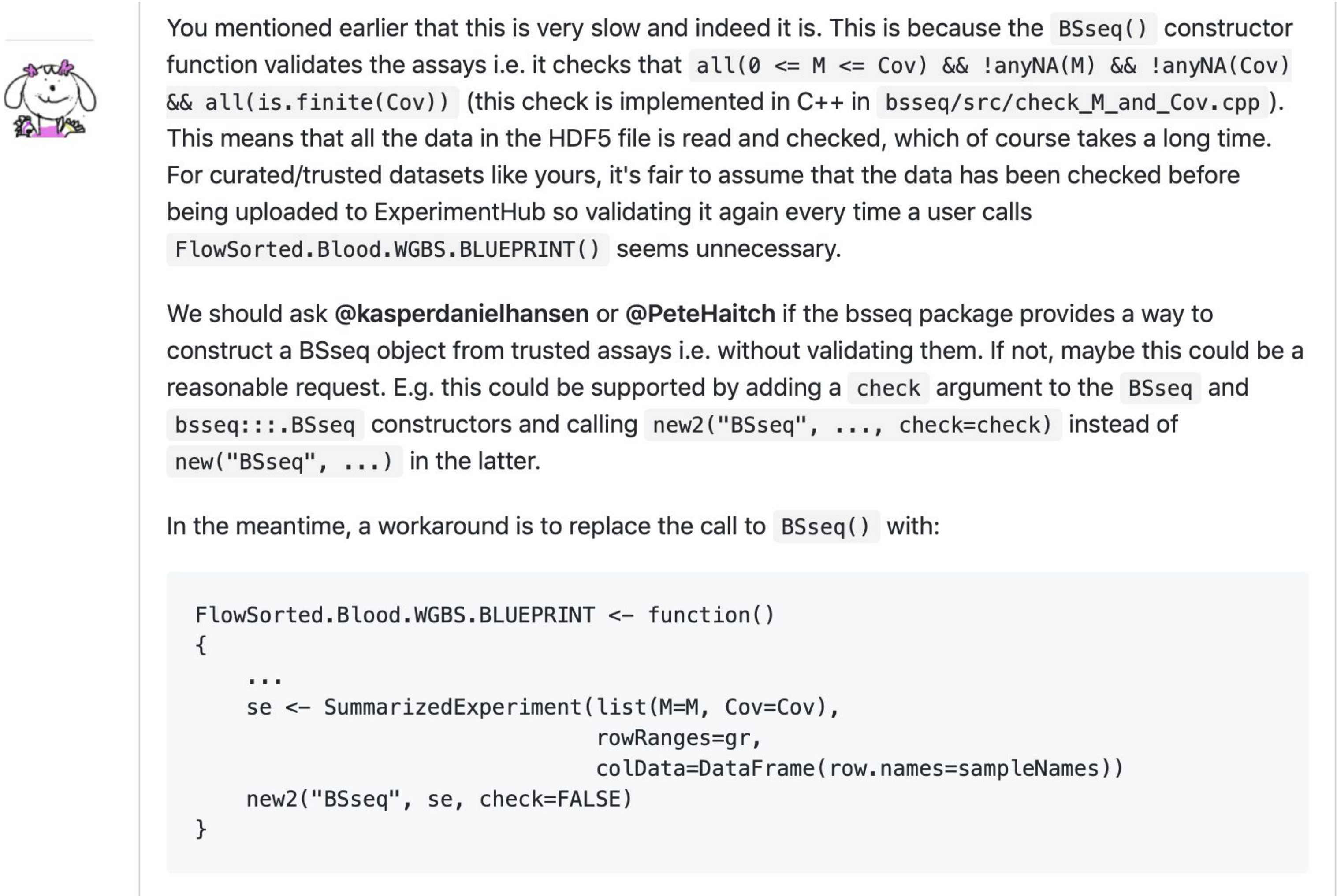
Even though it would be possible for your `FlowSorted.Blood.WGBS.BLUEPRINT()` function to (1) download the 2 files from ExperimentHub (to the local ExperimentHub cache), (2) create a temporary directory, (3) copy and rename the 2 files from the local ExperimentHub cache to the temporary directory, and (4) finally point `loadHDF5SummarizedExperiment()` to this temporary directory, this solution would be inefficient and fragile.

A better approach is to upload to ExperimentHub whatever components need to be passed to the `BSseq()` constructor to create the object, that is:

- The HDF5 file (already on ExperimentHub but maybe you want to consider changing the dataset names).
- The rowRanges i.e. the GRanges object passed to the `gr` argument of `BSseq()`.
- The sample names: this could be a serialized character vector but it would make a lot of sense to store it in the same HDF5 file as the assay data (as a 3rd dataset).

Then `FlowSorted.Blood.WGBS.BLUEPRINT()` can simply be something like:

```
FlowSorted.Blood.WGBS.BLUEPRINT <- function()
{
    eh <- ExperimentHub()
    assays_h5file <- eh[["some_EH_ID"]]
    gr <- eh[["another_EH_ID"]]
    M <- HDF5Array(assays_h5file, "M")
    Cov <- HDF5Array(assays_h5file, "Cov")
    sampleNames <- as.character(HDF5Array(assays_h5file, "sample_names"))
    BSseq(M, Cov, gr=gr, sampleNames=sampleNames)
}
```



You mentioned earlier that this is very slow and indeed it is. This is because the `BSseq()` constructor function validates the assays i.e. it checks that `all(0 <= M <= Cov) && !anyNA(M) && !anyNA(Cov) && all(is.finite(Cov))` (this check is implemented in C++ in `bsseq/src/check_M_and_Cov.cpp`). This means that all the data in the HDF5 file is read and checked, which of course takes a long time. For curated/trusted datasets like yours, it's fair to assume that the data has been checked before being uploaded to ExperimentHub so validating it again every time a user calls `FlowSorted.Blood.WGBS.BLUEPRINT()` seems unnecessary.

We should ask **@kasperdanielhansen** or **@PeteHaitch** if the bsseq package provides a way to construct a BSseq object from trusted assays i.e. without validating them. If not, maybe this could be a reasonable request. E.g. this could be supported by adding a `check` argument to the `BSseq` and `bsseq:::.BSseq` constructors and calling `new2("BSseq", ..., check=check)` instead of `new("BSseq", ...)` in the latter.

In the meantime, a workaround is to replace the call to `BSseq()` with:

```
FlowSorted.Blood.WGBS.BLUEPRINT <- function()
{
    ...
    se <- SummarizedExperiment(list(M=M, Cov=Cov),
                               rowRanges=gr,
                               colData=DataFrame(row.names=sampleNames))
    new2("BSseq", se, check=FALSE)
}
```

Discussion on data storage format convention:

Simplest data representation (e.g. HDF5) vs derived classes (serialized objects)

Two questions to that I want to discuss today:

1. Should we create guidelines for developers on **naming** ExperimentHub data packages?

2. Should we create guidelines for developers on what format the data are **stored** in when submitting an ExperimentHub data package?