Global Alliance
for Genomics & Health

GA4GH refget API

Minutes and Actions 2021

These are the minutes for the refget API meetings, a subgroup of the GA4GH Large Scale Genomics Work Stream. For further information, please visit the GA4GH Work Stream page.

Let's make the world slightly more interoperable and end the chr1 vs 1 debacle of bioinformatics!

Meeting Protocols

Agendas, Minutes and Actions

- 2021-12-15: Refget discussion
- 2021-12-07: Reverse Lookup discussion
- 2021-12-01: Refget discussion
- 2021-11-17: Refget discussion
- 2021-11-09: Reverse Lookup discussion
- 2021-11-03: Refget discussion
- 2021-10-20: Refget discussion
- 2021-10-19: Reverse Lookup discussion
- 2021-10-06: : Refget discussion
- 2021-09-22: Refget discussion
- 2021-09-14: Refget, Reverse Lookup
- 2021-09-08: Refget discussion
- 2021-08-25: Refget discussion
- 2021-08-17: Refget, Reverse Lookup
- 2021-08-11: Refget
- 2021-07-14: Refget
- 2021-06-30: Refget
- 2021-06-22: Refget, Reverse Lookup
- 2021-06-16: Refget
- 2021-06-02: Refget
- 2021-05-25: Refget, Reverse Lookup
- 2021-05-19: Template
- 2021-05-05: Template
- 2021-04-21: Template
- 2021-04-07: Continued string-to-digest discussion
- 2021-03-24: Continued string-to-digest discussion
- 2021-03-10: Connect Debrief
- 2021-02-24: Continued string-to-digest discussion

2021-02-03: Continued string-to-digest discussion
2021-01-20: Continued string-to-digest discussion
2021-01-06: First Sequence Collection 2021
2021-MM-DD: Template

Meeting Protocols

- Please note that by participating in meetings, attendees agree to adhere to the GA4GH Standards of Professional Conduct.
- Meetings may be recorded for note-taking purposes. Recordings will be deleted within three months of the meeting taking place.
- Dates should be specified in the international format yyyy-mm-dd
- 2020 Previous minutes available here
- 2019 Previous minutes available here
- 2018 Previous minutes available here
- https://w3id.org/ga4gh/minutes/refget should point to latest minutes

Agendas, Minutes and Actions

2021-12-15: Refget discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Sveinung Gundersen, Yash Puligundla, Robert Davies, Shakuntala Baichoo

Meeting Recording (to be linked)

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			
3			
4			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	

2.	Previous Actions Review - /service-info discussion: https://github.com/ga4gh/seqcol-spec/pull/24#pullrequestreview-832574915 - Endpoint naming decision: http://seqcolapi.databio.org/docs - ADR for endpoint names, and POST body format: https://github.com/ga4gh/seqcol-spec/pull/24	
3.	Comparison function demo: http://cloud.databio.org/links.html	Nathan
4.	Terminology streamlining: https://github.com/ga4gh/seqcol-spec/issues/25	Nathan
5.	Other issues raised	
6.	AOB	

Minutes:

NS: This is what I settled on and implemented: http://seqcolapi.databio.org/docs

NS: You would give it two digests or pass one digest via the url and the second one would be a post. If you click on those you can see examples. In addition there's also the ADR for end point names which I think people have approved. People had raised a few points. The only one question still is the service info question that Andy raised. Andy, you're one of the ones who contributed to the service info spec so the way we're interpreting it is this is how we should do it. The refget one puts service info under /sequence. But that doesn't make sense to me, might make more sense to have service info live at the top level of a particular API spec.
AY: The way the spec was written was how you originally said. Refget namespaced it under sequence and this is partly due to involvement of John Marshall. Aim was to ensure that the espec provided service info so it was already namespaced away from clashin service infos on the same server. What refget did is not necessarily what service info si set at the moment. Most of the service info endpoints are hardcoded. Assume that people have urls that they are giving which you can start to cue of service info from. This is a point where the specs would diverge ultimately you may end up with inconsistent naming of sequence collection API. Not strongly for or against. If we haven't decided to cut it off under a url hierarchy should we be pickier about the names we use to enforce the hierarchy.
NS: Feel like GA4GH should come out with recommendation saying if you provide a bunch of endpoints you should provide /service info. And then maybe people could provide suggested namespace. We can't control under what top level paths people use for this particular spec. The scope of our spec is what are our set of things. I don't think we should think too much about what servers are going to be doing. Feel like that should be determined at the G4GH evel. Any

collection of endpoints that fit GA4GH should be /serviceinfo. Doesn't make sense to namespace refget.
AY: Ok with the ADR as it stands then.
TC: The only other thing is in your example where you do have sequence collection service info and sequence implemented, then it is a little bit confusing to have service info at the top. Service info is inside the sequence. In those cases it's going to be a bit confusing.
NS: If I did want to implement two services and not namespace them, which may be possible if there are no clashes, then it seems like the service info ought to be able to define multiple services that are defined at this top level end point
TC: Don't know enough about service info to say if it's possible. They say you should use the service registry to specify service info. Feels like it's not what's recommended.
NS: You can think of my sequence end point as sequence collection.
TC: Nice example of how it could be misunderstood because they don't actually clash.

Comparison demo: http://cloud.databio.org/links.html
Before we finalized the return value of the comparison end point we wanted to look at some actual examples. So this is a static html page. I made a little matrix that gives you a link to the comparison for these genome. Took the four genome that Andy had used and added the two genomes that we're using in refgenie. Here we can tell that A is a subset of B completely for all three of the arrays. That's useful info to know. Primary one is a subset of the top level one.

Another instance is that there are some lengths shared but the name and sequence dont match at all. But all of the lengths in A are in B but not in the same order.
TC: Seems odd that it doesn't match in order for length even though it does for sequence. Because length is a subset of sequence. In the sense that the same sequence should have the same length.
NS: That's true. Will have to look into that. Not a lot of congruence between these ones. Some of the sequences are shared but it's not a…
AY: will be the case for most if not all of the providers because I don't think a single provider uses the same names. Various different groups have different sequence contents. At least here you could say if you have elements shared between sequences you have some compatibility between the two providers.
NS: Yeah but also highlights lack of compatibility.
AY: In some of them it will be ambiguity symbols changing to ends. In other cases it will be interpretation or representation of various haplotypes of certain hashes. Length can also be influenced heavily because different groups may have created different references based on the nature of their assembledness. But really they shouldn't be changing lengths. These gnomes are compatible with each other but they're not the same thing
NS: We often see this relationship that the lengths are subset relationships. E.g. UCSC compared to ensemble primary assembly, all B and A lengths. Lengths is often subset relationships but they have different names and different sequences.
SG: What is lacking is the possibility of showing different subsets. Whether the differences are in the alternative sequences or…

AY: Top level means the highest level of representation of every piece of sequence we have (single chromosomes, contigs and super contigs). That file also contains the patch regions. They are enpadded to match the lengths of the original chromosomes. This is bad. Ensemble shouldn't do this and they'll be changing this. The primary assembly is what you would consider traditional chromosomes. Primary assembly is part of top level.
SG: In a sense this level of representation is a bit overrepresenting the differences
AY: There will be more compatibility in the more important parts for sure. There will still be difference between providers.
NS: UCSC has a different set of sequences but NCBI and Ensemble are using the same.
AY: I thought UCSC had more compatibility with Ensemble than NCBI. Might have done it incorrectly.
NS: What I figured out by doing this is the refgenie 38 we're using uses the ensemble sequences but the ** uses the UCSC sequences.
AY: That is going through another layer of modification as well. The refgenie38 doesn't agree with names so the names are being projected.
NS: It's using the UCSC names but the ensemble sequences. Is this the return value that we want. The point of the comparison function is to highlight the differences. You're trying to say how close are these. You're looking for things that are a good match.
SG: Yeah but I eman if for instance everything you're interested in is the primary sequences then this is a bit useless. Many of the changes might not be there. I would strongly support having subsets so we can provide analysis at subset level also. For example primary sequences would be a subset of sequences.
NS: You meana sequence collection would annotate some as primary and some as secondary? Yeah ro any other subset
TC: Isn't that where the second comparison function comes up to play.
SG: Yes but if the sequence collection already had that in the data structure…
NS: One of the arrays we suggested ewas priorities. So people could put a priorities array.
SG: Yeah but hat wouldn't say anything in this context right?
NS: Yes you're right
TC: I understand where you're coming from but then we need to define what is a subset and it's going to be hard to define how the subset will be provided. It's easier to say when you have a subset provide it.
SG: But I do believe most of these reference genomes are already organized in subsets.
NS: The problem with that is I see a sequence collection as a FASTA files. If you're talking about adding in subset information then you're going beyond a FASTA file.
TC: I'm not sure I'm convinced that it's a part of sequence collection but it can be implemented by the service itself.
NS: Yeah so that simplifies things because you use what we've already done and run you rocparisons against the subsets. It's just not built in. I think I'd advocate for that because it reuses what we already have.
TC: It's something you could implement on top of the current infrastructure without changing the current infrastructure. Defining the subset within sequence collections seems complicated

SG: I did provide suggestions for how to do that which will also solve the order question. It would complicate the comparison function. In any case it would be an optional thing. If a sequence collection does have this subset…
AY: I'm confused about what's coming back in those sets as well. Wondering why these things not matching each other?
NS: What I intended it to be was for the set that was in both, are they in the same order? So are the shared elements in the same order. Does make sense that if sequences are there, lengths should also be there. So there's probably some kind of bug in the order match.
RD: Think it's getting a match from the depths..
NS: that's probably what's happening. If you have a whole bunch of chormosomes just by chance you'll match length. But if you match even just one then it'll show up on your match. That's not a bug but it's confusing.
TC: Do we have enough information? In the case where things match, I think we do. But when things mismatch I wonder if we have use cases that are unfulfilled. When we have tings that match then it's fine, but we'll be puzzled when it doesn't match. Wondering if it's the job of sequence collection to solve that.
NS: That's been my vision of this. The comparison function tells you where things are a hit. I need all A and B for names and lengths and sequences . If that's not the case then we're not compatible. Otherwise there are so many possibilities.
TC: Probably a communication thing to address. If you have the sequences where the UPAC codes have been converted to ends they're incompatible but they're perfectly fine to use. You can use one for the other and the elements would be fine.
NS: In that case, youw ould get that you're lengths and names matched and were in the same order
TC: Yeah but your requirements would be on sequences .
NS: Depends on what you're doing
TC: Not trying to say it's not good enough, but there will be a disconnect between what we say is compatible and what other people are using that word for.
Doesn't answer all questions of compatibility but does answer some. E.g. do I have the same lengths and names but just with different sequences. Maybe we have to think about what compatibility questions we want to answer.
SG: Allow return of the number of matching and mismatching elements. Another thing I was thinking about is the possibility of doing this kind of analysis. As a user you could say you want to compare…For me I'm assuming that the differences we're seeing are not so central. I feel that the current compatibility function is a bias because even a tiny differences would show up with a whole warning sign. Limiting that effect. One way to limit the effect is to say I just want to look at these sequences or these lengths. Those you don't take into account here.
NS: Can't you do that by only posting sequences you're interested in.
SG: Are these NCBI and UCSC differ in the premise…
NS: to answer that question we need sequence collections of only the primary sequences and you just ask it of those sequences.
AY: What you're asking is not what's different about the fundamental data type in the set. You're asking about the data that's behind it. Sequences is one of the few fields we have where the

data in the array is insufficient to answer that. Not sure of a good way to answer that without leaving it up to the user. Would have to do an actual analysis. Will probably affect everyone who goes to this and asks let's see what my genomes looks like. Could engage with GEL and Australian genomics. We could ask them to do a test with their human pan genome reference.. To lessen the warning, we could select a subset but it will still take you to the same thing. That chromosome 1 from all three providers were different. If you want to figure out more you have to do it yourselves. Know we said that paper is out of scope but maybe we need to consider it. It will be scary for a lot of people. They will think their analysis is incompatible.
TC: To me, at least in the current state that your example is in, I think we have the answer to the question we wanted to ask in the beginning. We should provide examples of what everything means.
NS: It's a fundamentally confusing question. Is this level of abstraction useful? I think what we have answers a lot of the use cases in something that is relatively simple. We're not going to have a simple function to answer all the other questions that are coming up. Maybe going to the numbers is valuable enough to elevate it. For most of those questions, if it's not all A and B and not all B and A then who cares.it won't be comptable. It's only if there's a subset relationship or a match in lengths and names that I can go on computationally without needing a human to come in.
AY: I would also say that you can answer some fundamental questions from ingest.

2021-12-07: Reverse Lookup discussion

Chair: Tim Cezard
Attendees "Name (Affiliation)": Rob Davies, Shakun Baichoo, Yash Puligundla, Andy Yates

Meeting Recording (to be linked)

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Specification of id based reverse lookup: see #604	Tim

	• Within refget • New endpoint	
4.	Next meeting 4th January ?	
5.	A.O.B	

2021-12-01: Refget discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”:

Meeting Recording (to be linked)

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Implement new comparison function return value with some real examples (see: http://seqcolapi.databio.org/docs/)	Nathan	Done
2	Finalize compare endpoint API, just needs a decision	Nathan	Done
3	ADR for endpoint names: - /comparison? - /collection	Nathan	Done
4	ADR for POST format: /comparison/digestA POST content: complete sequence collection for digest B	Nathan	Done

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Comparison function API specification: 1. Endpoint names for comparison function, finalize GET vs POST options: https://github.com/ga4gh/seqcol-spec/issues/23 2. Data structure for user POST value	
4.	Order debate	

	https://github.com/ga4gh/seqcol-spec/pull/19	
4.	A.O.B.	

2021-11-17: Refget discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Yash Puligundla (GA4GH), Shakuntala Baichoo (University of Mauritius), Robert Davies, John Marshall, Anders Leung, Sveinung Gundersen

Meeting Recording

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Comparison function API specification https://github.com/ga4gh/seqcol-spec/issues/21 - Flipped data structure for return - Endpoint names for comparison function, finalize GET vs POST options - Data structure for user POST value	
4.	Order debate https://github.com/ga4gh/seqcol-spec/pull/19	
5.		
6.	A.O.B	

2021-11-09: Reverse Lookup discussion

Chair: Tim Cezard

Global Alliance
for Genomics & Health

Attendees “Name (Affiliation)”: Shakuntala Baichoo, Anders Leung, Robert Davies (Sanger), Sveinung Gundersen, Yash Puligundla, John Marshall

Meeting Recording

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Versioning of Refget: ● Duplicate files (for major version only?) ● Tag repository with spec specific tags (see #194 for information) ● Pages in the Github repository can maintain old versions (keep old version by linking to specific commit)	
4.	Allow sequence endpoint to accept identifiers #604	
5.	A.O.B.	

Minutes

TC: Would like to talk about what people think is the best way of versioning. Last time, we talked about this, there were only really two ways that would work (i.e., easy and not destructive).

1. Duplicate the markdown file and the open API file and put them into a directory* (easiest option, but not sure if it’s sustainable)

TC: We’ve opened a different repo for reverse lookup. The reason I’m interested in versioning refget spec, is because there are two PRs that are bumping the verison. During Connect, there was discussion about having a canonical version that people could refer to for their spec. It would be good to point towards a specific version of refget, not just the repo. This way people can control what version of refget they want to use. At the moment, that’s not easy to do. There’s a few things that reverse lookup would like to do on the refget spec themselves.

JM: taking things in the repo has come up before (issue #194). Hts specs has been a repo with simple infrastructure. Makes available ;latest version of these documents. For file format, that’s appropriate. Htsget is in same boat as refget, but they also just made available version of the spec. They’ve recently added an appendix to the spec which gives a listing of the changes that

have occurred over time. Feel that this is more useful than just looking at an old version of the spec to figure out the differences. What we will probably move to do is what JB suggested (see issue #585), we've got version numbers which will turn into links to the markdown files to the repo that best represent those versions. Another approach you could take (htsget is doing this) is have one document to the repo and have the latest version of it and that's what appears on the website. Version history is linked back to pages within github repo that gets displayed as markdown history on github.
TC: Is it a manual step to create those html pages?
JM: No, it's manual to create the pdf documents, but github automatically updates the html pages.
TC: It is essentially tagging, but you don't give a name to the tag.
JM: That's true. For tagging, because the repo is shared between a number of specs, distinctions between spec's spaces need to be made clear.
TC: At the end of the previous meeting we were leaning towards duplicating the files, but I don't think that's sustainable. Was leaning towards doing tagging. What you're proposing is tagging without giving it a name. Reserve the right to change the actual commit that represents 1.1.,1.2 etc.
SG: Make use of the integration between github and zenodo. When you create a release in github it uploads to zeonod. They provide version identifiers in a way that will always point to the latest version.
RD: Problem there is if you release in hts specs, you release in everything.
SG; Agreed. Could set up specific integration that pushes the pdf file into zenodo.
RD: The idea of having links back to actual commits is pretty simple as a solution.
TC: Yes, it's simple and flexible.
RD: Maybe we should be better at revising document version level whenever we make a change. For our links back we can do it so that when the feature came in and the latest *** level of that documentation. Don't worry about the intermediates.
TC: Advantage of link back is that it's up to the maintainer how they want to link back. Doesn't have to be decided. Would be better if it's consistent. Will open a PR or make a modification to Andy's PR.
SB: How many versions do we expect to keep track of ? Should we keep track of all versions?
TC: Assuming that refget gets used across GA4GH, we won't necessarily know what version is being used in other tools. As soon as we start tracking versions and say that we are bumping, we need to keep track of this. But the choice of when we want to bump the version is up to the maintainer.
SB: Also we would release a new version of the protocol when we have had some major changes, isn't it?
TC: Yes, that's the idea. The protocol can change even without a major verison *. E.g. allowing refget to receive identifiers and not just digest. That's not going to be a breaking change, it's adding functionality .But it will still change the protocol.
SG: Make use of semantic versioning (3 numbers, 1st major number changes when there's breaking change, second number changes when there's feature changes that's backward

compatible and the third number changes when there are bug fixes that are also backward compatible)
TC: In all of the specs maintained by hts spec, we are not assigning patches. No real reason for that other than the fact that it's cumbersome. Means we are not keeping track of small changes apart from commits.
JM: The pdf specs of SAM and VCF have on their title page commit hash..., last modified.. So you can tell how new the doc is.
TC: Do you think it's worth adding that to the pages?
JM: Never looked into it, but might be.
TC: Want to talk about sequence end point. Will be discussing with Rasko and Andy, to have this sequence end point accept other types of strings. Have suggested a rewording.

…

JM: Concern I have is that refget is independent of the server that you talk to. So if you ask for a digest then the server says I don't know or here it is. And it doesn't matter what server you're talking to. So if we query by identifiers, how do you maintain that proversty. If you require it to be a registered identifier that answers the question probably.
T : Not up to the refget spec to specify what identifier you want to support. You don't even have to support identifiers. It's just another way of naming. So as long as your identifier uniquely identifies your sequence then that's fine.
JM: Isn't that vague. Chr1 could be ambiguous. Then you get back to it matters what server you talk to.
TC: Not sure if that's a problem.
RD: Would have to be choosy about which identifier to allow.
TC: In the example I have on this issue, if you don't give the version number you won't get the issue. The one without the version number doesn't resolve to a sequence.
JM: This is the same question that reverse lookup has had the whole time.
TC: Originally that ID comes from Rasko, who implemented that for ENA.
...
TC: What's the solution tha DRS implemented?
RD: Look up things in identifier.org. But it's not working.

2021-11-03: Refget discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees "Name (Affiliation)": Yash Puligundla (GA4GH), Shakuntala Baichoo (University of Mauritius), Robert Davies, Sveinung Gundersen

Meeting Recording

	Actions Arising (label SC/RL/general)	Assigned To	Deadline

1	Create a flipped data structure for example return value	Nathan	Done
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Comparison function API specification https://github.com/ga4gh/seqcol-spec/issues/21	
4.	A.O.B	

NS: Decisions to be made: End point name, structure of post-content (post or get?), structure of value returned by server.

Structure of value returned by server

SG: Question about whether the ADR should provide more detailed answers. Which element is shared, how many are shared?
RB: Shelve it under future enhancements.
NS: We could approach it as the minimum you have to implement to satisfy the implementation. Doesn't mean you can't provide more info.
SG: Could make it an option.
RD: People can request that if they want to. Or else they may get a lot more data than they expect.
SG: What about the sizes of this set? The number of A sequences in B and B sequences in A.
NS: Yeah that's a possibility. The reason I made it simple like this, is if you're looking for if they're the same or subsets, and if they're not then that answers your question.
SG: If you changed from any element to, "elements shared" there could be three types of answers to all of this. Yes or No, True or False, the number or actual sequences. You could provide that as a parameter which results at the level at which you want to answer. Would logically be the same.
AY: Keeping payload simpler is probably better because it will help drive other implementations as there will be less work to do.
SG: Agree as a requirement. Idea was to have an optional spec given you want to provide more information.
AY: If we want to specify way to provide extra info, we could suggest a way to do the return as an extra piece of supplementary info (i.e., best practice)

NS: Proposal → We make the way it's currently specified required. Then option is an underscore enumerated*, and in there you have by attribute arrays with additional attributes that could be shared elements, only A and only B.
AY: Like the idea. Do we need the underscoring?
NS: Name space is such that those can be any value since those can be any attribute.
AY: Underscore attributes could be reserved items.
NS: This isn't an attribute of your array. The other things are all attributes of your arrays. If you're including info that is not an attribute of an array, you should prefix it with an underscore. This is a convention (AY +1)
SG: Could be specified the other way around so that the 2nd level is 1st and the 1st level is 2nd. Might make it easier if we want to add a new type of output.
NS: Flipping the data structure could solve the problem but seems a bit less logical.
SG: True, but also not meant for human consumption.
NS: Of the opinion that it should stay this way and I like the underscore idea at the attribute level. One of the benefits is that the payload would be smaller. Can create that data structure for next time in the other direction. So we can say that it will be requested that it will return that information, but just that info in some form. Then we'll have a suggested structure for how to return additional info.

End Point

NS: Should the end point serve only post requests, only get requests, or both?
SG: Only get is not actually an option.
NS: The proposal is that we'll have an end point which will provide this comparison return value. That endpoint needs to get info about what the two sequence collections to compare are. If you give me two identifiers, I could compare them. That could easily be set up as a get request. There's an argument that that's a convenient way to run a comparison assuming that the database has both identifiers. There's the argument that I have a sequence that your server doesn't know about. It's harder to run...Do we ignore the get, because the post one can handle the simpler case...The point of the server is not just to provide computer power, it's to provide access to data. Do we get rid of the get end point and only implement post, or do we also implement get because that's convenient?
TC: We are not defining the implementation, we are defining the spec. We are defining something that is minimal. Don't want to have too many options, because it's not healthy to have so many branches. Specifying the get function won't have a massive impact, but as a principle I want to stick to simplicity.
AY: Only reason to implement get is because there's a requirement to specify a more simplistic server. Only argument for get is for consistency with older spec which is get based.
SG: People will be using this as well and get makes it simple for them. Since the impact isn't large, don't see why we shouldn't.

NS: What about making the post optional and get required?
AY: Like the idea of get being default, and post being additional. Would require extensions to**
TC: Not necessarily that I prefer post vs get. Point is that I would prefer to stick to simple solutions. So it's valid to say that get would be default, and post could be an additional one for more complex comparison. Also helps with points about whether or not we allow comparisons between other levels. Doesn't require a database. If you only have get you don't have the option.
SG: We could provide different parameters for different levels. Specify which ones you allow. Then you assume that if it's level one, it's level one.
TC: I would be okay with get by defaults and optional post.
RD: +1 -- Do the minimal possible and see how much use it gets. If people want more functionality, add it later.

NS: For later discussion → What's the structure of the post given by the user?

2021-10-20: Refget discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees "Name (Affiliation)": Yash Puligundla(GA4GH), Rob Davies(Sanger, Anders Leung (GA4GH), Shakuntala Baichoo (Uni. of Mauritius/H3aBioNet), Sveinung Gundersen - 10min late Susan Fairley (GA4GH)

Meeting Recording

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	

3.	Starter kit for refget sequence collections	
4.	Connect Session Debrief	
5.	Compatibility Tests	
6.	A.O.B	

2021-10-19: Reverse Lookup discussion

Chair: Tim Cezard, Andy Yates
Attendees “Name (Affiliation)”: ,Yash Puligundla (GA4GH), Robert Davies (Sanger), Shakuntala Baichoo (Uni. of Mauritius/H3ABioNet), Susan Fairley, Anders Leung, Rasko

Meeting Recording

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3	Refget versioning: Is there a stable official URL for each version? ● Could use tag with prefixes ● Migrating to its own repo is not preferred ● Multiple file for each file Standard text to describe the normalisation Also need a standard text on how to cite	
4	Reverse lookup for unique identifier ● No discovery interface at the moment but that should be manageable at the moment because there isn’t many implementations	

	Reverse lookup using filters https://github.com/ga4gh/reverse-lookup-spec/issues/1	
5.	A.O.B	

Need support for newer service info
Support for ga4gh identifier and deprecation of trunc512
Discuss the compression of the sequence in a future meeting

2021-10-06: : Refget discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Rob Davies (Sanger), Yash Puligundla (GA4GH), Sveinung Gundersen (Uni. of Oslo), Shakuntala Baichoo (Uni. of Mauritius)

Meeting Recording (to be linked)

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Connect Session	
4.	Compatibility tests - Implementation example by Nathan	

Minutes

NS: Wrote up an implementation. We are trying to finalize what the compatibility function is. A function that, when given two sequence collections, tells you something about the relationship between them. If tht function is useful, then that solves some of the questions we’ve been dealing with with whether or not we include order and arrays. Last time, we talked about a few different levels of compatibility. E.g. one response that says true or false. A second level could return more detailed info. Discussed the possibility that the simple layer is required and the

more complicated layer is optional. There's a compatibility function and search function. That's a different query. Given one sequence collection, find all the other seq. Coll in the database that are somehow compatible. Is that search function part of the spec or is it just an implementation that people can choose to implement? It's a harder question to ask. Hadn't thought about the search function in the new array based approach. Thought of a way to accommodate it taking into account the arrays.
RD:...
TC:...
NS: Are you suggesting that the search would only be required for a single seq collection…
TC: Search through multiple arrays is an interesting suggestion. My thought was to provide a mask that would provide a compatibility type that you are searching for.

2021-09-22: Refget discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees "Name (Affiliation)": Rasko (EBI) , Yash Puligundla (GA4GH), Rob Davies (Sanger)

Meeting Recording
Meeting Transcript (to be linked)

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Come up with comparison examples for the compatibility function based on some common human references	Tim	
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review - See https://github.com/ga4gh/seqcol-spec/pull/19	
3.	Discussion on compatibility tests: - Related discussion: https://github.com/ga4gh/seqcol-spec/issues/7 -	
4.	A.O.B.	

Minutes

NS: The compatibility function takes the complete representation with the element, It requires two of those. Would return the compatibility level for each array. Would build on top of that a

convenience API where all you have to give it is two digests as long as they both exist. You could also give it one digest. There's two components: python function and server component. The server component would nede a few different API endpoints. Would all be based on that underlying function.
TC: Agree in what it's supposed to be providing but to define the spec we need to talk about the end point.
NS: And we need to talk about what it's returning. TC had this comment proposing what it should return. For any one array there's 4 things we're looking for.
TC: Do we have anybody who is against the idea of the compatibility function or sees issue with it, before we go into the interface that it will be provided through?
NS: We need to describe the API. What are the inputs and what info must it return? Sveinung had a suggestion that it could return more information. Instead of all A and B and all B and A it could return all the elements in both. It could give a lot more info then just this. Initially envisioned that comparison between two collections could be represented by two binary flags. If you 'and' it to 1 and 2 then you know the sequence content is identical. Makes for fast comparisons, but not as info rich as what Sveinung was proposing.
TC: Would have to provide sets. Would need to have multiple objects for each of the arrays. Need to have all of the ones that are in A and not in B and all of the ones that are in B and not in A...More informative and useful to have the detail but not sure what use case it responds to. What you want to know is what's compatible. If they are incompatible, all of the use cases we came up with so far are finding things that are compatible. Can't think of use cases where you would want to know if they're incompatible.
NS: You could always figure that out. Just query the APi get the two sts and run some function on them. That's why I'm thinking the compatibility function would return something very simple. Just tell you if it's compatible or not. More functionality sounds great but also it makes it harder to implement the spec.
TC: Could be a bit more expensive to calculate. To find out if All A are in B just need to find one element in A that is not in B. At the spec level if we enforce that they need to provide all details of incompatibility then it's more expensive for implementation. Not sure what the benefit would be. Want to clarify that this is not a search function - cannot find something that you already know. Is this a problem?
NS: Similar comment here. I do think that's useful, when we had discussion about it last year there wasn't a lot of enthusiasm for that so we put it as a secondary thing. Fine with either way. Think I would use that but can also see building that on top of a simpler sequence collection framework that only has compatibility function.
TC: Don't need to spend too much time talking about this find function, but should look into the possibility of adding this.
NS: Wouldn't be hard to implement. Clear that those kinds of functions are useful. Question is, is it a part of sequence collections or a separate project that someone would do on top of a sequence collections server.
TC: Not a question we need to answer now but it's something we could add on.
NS: So there's a compatibility function for which inputs are two sequence collections and outputs are compatible. And there's a second concept of search function. Input is one sequence

collection and compatibility criterion and return is a bunch of sequence collections. Search function is out of scope for sequence collections 1.0 but acknowledge that it's useful and someone could implement it or could be added in version 2.0.
R: Need time to think about use cases for search functions.
RD: Think a search function could be useful for reducing proliferations from the pov of someone loading sequence collections.
NS: Could also be from the pov of someone using sequence collections. Say you had a user that named chr 2 as chr 1 and you were doing your compatibility test. How would you return this? Starts to get complicated.
RD:More likely they have completely named it differently or they have all the chr named the same but are in a different order. Or they've done both different names and order.
TC: Don't understand the use case for wanting the details for why things are not compatible.
RD: Another possibility is for a BED file which has one digest on the top of it and VCF file with another digest and you want to know that the one on the BED file is the same as VCF but it only contains names and lengths but not sequences. That's a useful comparison to make.
NS: But that you can get with just base level stuff. Wouldn't have to go down to the specific diferenes. Is it enough to say match or not match, or do we have to go the step further to say this is what doesn't match.
RD: Could also punt this down to end users. Nothing here needs to be run on the server.
NS: I'm pro providing compatibility function but you're right that it's not a c ore part of the spec.
TC: Could consider it the same way as we are considering the search function, It's an add on.
NS: One of the reasons we wanted to have this compatibility function is because it would make it easier to answer the question: are these the same but just in different orders? That's a common question.
RD: Need a definition of what compatible means.
NS: Proposal is those 4 questions. Given two arrays **1. Are all elements of A in B? 2. Are all elements of B in A? 3. Are they sharing an order (i.e., are they identical?) 4. Are there any overlaps?** With that information you could make a good estimation of compatibility.
TC: Does same order mean they need the same content (e.g., if one is missing stuff at the end but they are in the same order).
NS: Let's say B is a subset of A that is missing one so you have all B in A (true) and you'd also have the order of B matches. So that's basically saying that the order of B is the same as the order of A. If those are both true then you know that B is a subset of A and is in the same order as A. You can imagine all the combinations.
TC: Do you mean there would be two order subsets? A as the same order as B and B as the same order as A?
NS: Yes.
YP: When you say are all elements in A present in B are you referring to name, elements or digests itself.
NS: Now because we're considering each array as its own independent thing, we would ask questions for each array independently. Each array would get these values.
YP: So it would be contents of the sequence array

NS: Yes, it would be contents of sequence array ,contents of length, so instead of separate things you would get flags for each.
TC: Need some flag for when an array is missing.
One way of doing this is to make an array of the value in it's position in the array so you get two lists. You sort them both on the item then you do a merge and resort...all matching ones get zero.
NS: You take the union of A and B (union array) essentially the OR and then you have an array for A which is the index of that element. What's the index in array A and what's the index in array B.
RD: Would probably do it in a hash table.
NS: If one of them is just missing one then all your positions are +1 and don't match.
RD: You return a flag to say this isn't in this array.
NS: That's fine for the one that is missing but if the rest are in the correct order. Would need to do more work to ask if they are in the same order.
RD: If some were missing would you describe them as in the same order?
NS: Yes. ABC for AC they are in the order or ABC but for ABC, I would say no because it doesn't contain everything. It depends on which way you're asking the question. Which is why I initially had two order variables, One for each array.
RD: If there was a hole in the middle I'd say they weren't in the same order.
NS: Yeah but you get that info from looking at the content questions.
AY: Basic yes and no makes sense. As someone that would want to use that info to then figure out why that's not the same thing would want more info coming back.
NS: Do you think every time you find out a false you would want to dig into it?
AY: If you were writing a program that wanted to answer that question, then you would just want true and false. If I was trying to submit to EBA and I was told that the submission is wrong then there should be some way to know why.
NS: The user would be able to look it up. How streamline does it need to be?
TC: Maybe we could just make sure to use negative in the key word.
NS: So you'd say something like A and not B, and then if that's empty then you know B is a subset of A. if it's not empty then you know those elements were in A but not B. Then you can say they're true or provide the elements, or provide the number of elements.
RD: Wondering if we need to make this more concrete.
TC: Can write a few examples for next time.
NS: Need to think about how a compatibility function works on array based approaches. Also there seems to be two levels of compatibility functions - simple and advanced level. Could all be in the same endpoint and different ways of returning info or all in different endpoints.

2021-09-14: Refget, Reverse Lookup

Chair: Andy Yates / Tim Cezard
Attendees "Name (Affiliation)": Yash Puligundla (GA4GH)

Meeting recording

Global Alliance
for Genomics & Health

Meeting transcript

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review • See issue #1 in github	
3.	A.O.B	

2021-09-08: Refget discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Rob Davies (Sanger), Jeremy Adams (GA4GH) , Yash Puligundla (GA4GH) , Rasko Leinonen, Sveinung Gundersen, Anders Leung

Meeting recording (to be linked)

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
	Write an ADR for choosing option A (draft)	Nathan	

	Agenda Item	Person/Time
1.	Welcome, Introductions	
2.	Previous Actions Review	
3.	Discussion	

- Is it safe to assume that names are unique?
- Tim: most other issues are behind this one.
- Nathan: Three proposals for order:
    - https://github.com/ga4gh/seqcol-spec/pull/17
    - 1st Option: Digest includes the order of the sequences
        - Lot of use cases we don't care about order
        - Hard to compare things that do not care about order
    - 2nd Option: Digest maintain order + order array
        - We need a canonical order for the order array (e.g. length, name)
        - Challenge because how does this order array relate to other arrays
        - Order by name -> uniqueness requirement for names if ordering is done by name
    - 3rd Option: Each array ordered individually
        - No need for canonical order
- Tim: 3rd option increases the space (doubles space required)
- Nathan: 3rd option is hard to understand and explain
- Sveinung: proponent of 3rd option
    - https://github.com/ga4gh/seqcol-spec/issues/5
- Rasko:
    - ENA sequence canonical order for sequence collection checksum digest computation: sorted sequence checksums
    - Arrays could be sorted in any order and the canonical sequence collection checksum is unrelated to the array order
- Rob:
    - Unordered sequence collection is only useful for searching
- Tim:
    - Provide an identifier for the sequence collection
    - If the identifier is different for different servers then this defeats the purpose of identifiers
- Nathan:
    - Possibly we are conflating digest and ordering
    - Do we need an order array at all?
- Tim:
    - Makes sense
- Sveinung:
    - What is a sequence collection
    - Should have an identifier (digest)
    - Might be without actual sequences
    - Might be ordered or not
- Nathan:

- Sequence collection must be ordered -> some use cases require order
- Coordinates systems have an inherent order as well

- Tim:
  - Makes sense
- Andy
  - No comment at this point
- Yash
  - No comment at this point
- Nathan
  - Digest algorithm should just use 1st option
  - Can be extended with other information
- Tim
  - Makes sense
  - One of the reason why things were mandatory (or not) was because or ordering
  - With 1st option almost nothing needs to be mandatory
- Nathan:
  - **Thinking that lengths at least should be mandatory**
  - **Also names or sequences possibly**
  - **Or perhaps just one array of whatever**
- Rob:
  - Need names and lengths for a BAM file
  - Names can be made up (e.g. ordinals)
  - CRAM file needs also sequences
- Rasko
  - Could be extended by e.g. ENA style sequence checksum based digest
- Sveinung
  - Confusing two things possibly: (reference) sequence collections and subsets of them
  - Use case: query subsets
- Nathan
  - We have tried to do this by saying that you can include just the arrays you want
  - Should be possible to do subset queries
- Sveinung
  - Option 3 solves the subseq query part (not without all permutation subquery digests)
  - Option 3 could be an optional feature but still in the standard
  - One could precompute all permutation subquery digests
    - This could be an optional feature

	- Nathan
	  - Implemented Option 1
	  - Could add extra features for subseq queries
	  - Sveinung use case is a different
	  - Give a digest and find stuff that is compatible with the digest (could be a subset digest)
	- Majority comments
	  - Option 1 is perhaps the best option	

4.	A.O.B.	

2021-08-25: Refget discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Rob Davie, Rasko,

Meeting Recording

	Actions Arising (label SC/RL/general)	Assigned To	Deadline

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review - Including order: https://github.com/ga4gh/seqcol-spec/pull/17 - Mandatory arrays: https://github.com/ga4gh/seqcol-spec/pull/18	

3.	Discussion on how to accommodate order in digest.
	See: https://github.com/ga4gh/seqcol-spec/issues/5
	● Original order digest
	● Canonical order digest + order order digest
	  ○ Order by length, names
	  ○ Order by checksum
	  ○ Order by names	

	○ Rob: make it as simple as possible ● Order each array individually and require an index into each array	
4.		
	A.O.B.	

2021-08-17: Refget, Reverse Lookup

Chair: Andy Yates / Tim Cezard
Attendees “Name (Affiliation)”: Susan Fairley, Rob Davies (Sanger), Rasko Leinonen, John Marshall

Meeting Recording

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Create proposal ● Tim to create an issue on the repo ● See issue #1 in github	Tim/Done	
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions	
2.	Previous Actions Review	
3.	Discussion	
4.	A.O.B.	

TC: Have been discussing whether there is appetite to continue developing this spec. Not clear whether there is. On my side, we are developing a tool that would do something akin to the reverse lookup of refget. The reverse lookup protocol would be useful for ensembl as well. But sounded a bit like something that would be very EBI centric.
SF: Did we manage to get contact with anyone in dbgap to see if they would also be interested.
TC: Haven’t
Rasko: What is the service that you are planning to create exactly?

TC: Contig alias tool where you can specify name of a contig, name of a sequence and it will give you back one or more results and means of filtering down to the one that is of interest. Filtering would be either by taxonomy or assembly id. Use case is straightforward. Receive all kinds of submissions with varying names from ENA, Ensembl, UCSC or other sometimes exotic sources. Getting assembly reports from GenBank and ENA. This is the main source for aliases. Planning to add MD4, SHA12 digests also as aliases and these could be used to add additional aliases.
Andy: challenges originally were like what is the chromosome 1 checksum without having to download the sequence and compute the checksum. Overlap here with sequence collections and this might be a better place to solve this.
Susan: discussions could be used as a means for underlying reference to know unambiguously what sequence is being referred to in VCF files.
Tim: sequence digest inside the VCF header would be a way to do this. Sequence collections would allow a digest to refer to a set, rather than one, sequence. Digests are only used in CRAM really at the moment.
Rasko: ENA's CRAM. ref. Registry to support also INSDC accessions. Guess this could be considered as a reverse lookup within a unique namespace. Would be interested to standardize this use case.
SF: Could send emails out to gather interested parties and present to SC. might be good to have some kind of one-pager to share with people to see if they are interested or not.
ACTION: Tim will initiate the process to come up with a proposal. Rasko will participate in the specification process.

2021-08-11: Refget

Chair: Nathan Sheffield / Tim Cezard
Attendees "Name (Affiliation)": Sveinung Gundersen, Rob Davies

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Create new PR for ADR for which fields are mandatory/allowable combinations	Nathan	Done
2	Canonical ordering proposal	Sveinung	
3	Create GitHub issue - ordering options	Nathan	
4			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	

	- Final review of array-based method: https://github.com/ga4gh/seqcol-spec/pull/14/files - Use case for missing sequences (added to https://github.com/ga4gh/seqcol-spec/pull/12) - Canonical order proposal by Sveinung	
3	Make a decision on the canonical ordering	
4.		

Meeting Recording
Meeting Transcript

2021-07-14: Refget

Chair: Andy Yates / Tim Cezard
Attendees “Name (Affiliation)”: Michael Love, Rob Davies, Rasko

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Rewrite how the canonical ordering should be done assuming we want to use an order array	Sveinung	
2	Need to get a decision on why we would allow sequence to be missing: what is the use case for this ?	Nathan	Done
3	Andy to give Tim appropriate permissions on GitHub	Andy	Done
4	Cancel next meeting	Neerjah	

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3	• Write array-based method into decision record, including the multi-tiered digest approach. (Nathan) https://github.com/ga4gh/seqcol-spec/pull/14/files • Finalizing list of allowable attributes (https://github.com/ga4gh/seqcol-spec/pull/12)	

3.	A.O.B. • Repository write permissions: Currently I cannot request/perform review or add labels on the Seqcol repo	Tim

Meeting Recording
Meeting Transcript

Minutes

Ordering
AY: Last time we spoke about the order that you digest the arrays in. 1) do not impose ordering - means that you end up with collections that fundamentally have different orders. Challenges came from how loose we are in what we allow to be an attribute. Not many key attributes that you can have available. Checksums are not mandatory, you could have a checksum but no name. Length was only a discriminator which is fine until you have two sequences that are the same length. Discussion about whether or not this is okay.
R: People have submitted lots of CRAMs to us, need some way to verify the reference sequence. Pulling checksums from reference header** Order them by checksums themselves, if you want to order them.
TC: Started off SA that you can have sequence collection without sequences
R: **
TC: The name and the length gives you a coordinate system, and that is compatible with..
R: Unique identifier*
AY: Is there one root of logic you can take? The discussion ended up with “There needs to be a priority. E.g., sort by lengths and if there is a tie and you have name defined then it’s by the name, but if name is not defined then it’s by the sequence digest.”. But it was still unclear - other elements allow you to sort to a unique point unless you have that reference sequence twice.
R:
AY: If we have multiple implementations, then we want to make sure they are following a standardized manner of calculating those checksums. There’s no problem with multiple encodings, but rather ensuring consistency.
R: Determine pros and cons**
TC: For the ordering itself, it is technically possible that we’d have implementation specific ordering. E.g., we say you need to have length, but for the implementation they might say that they want length and sequences every time
R: But at end of practice you want some kind of**
AY: Depends what gets used for name. Name is fundamental to how you refer to the sequence. For downstream analysis we would want to know the different names that people have used. Fundamentally changing the output files as well. Reinforces incompatibility between providers but incompatibility is ** to have.
R:
RD: If you change the name you fundamentally change the checksums.

R: Not unless you say that**before the checksums is completed have to translate name into checksum.
RD: A collection with different names has to have different checksums.
R: Have a scheme where you have a sequence*checksums* can pull set of sequences. Can use same ID but pull sequence name from different places. Theoretically possible is you convert the name to checksums.
ML: Want to know if the sequences are the same. Hope that we would be able to find out that two sequences are the same regardless of who is naming them without downloading them.
AY: You can still get to the checksum which is the digested version of the ordered sequences and you could do the comparison straight away .Still a two layered thing. Checksum on top level. So Mike, it is still supportable as long as the way that the checksum is created is **
TC: Important to recognize that the users will want names. Becomes important because the names are what will be used to identify. Almost all the time, the hashes disappear.
Tc: Possibility of defining sequence collection only caring about the coordination space of the annotation.
R: Same case, different naming scheme.
AY: Don't know if there's a case where there isn't a physical sequence somewhere. This particular use case kicks in downstream of the collection provider. BED file could have a sequence collection referred to and that could be a collection sequence checksums. To find sequence you care about, you would care about name and length.
AY: Need to get a decision recorded for the use case where there is no checksum.
R: Doesn't mean there can't be a proper naming scheme.
AY: Additional naming schemes could come as additional arrays. Could constitute part of the collection, but ultimately not portable between providers. Links to how compatible the providers are to start with.
TC: Question of whether we want to make sequence hash mandatory is a good one that we need to discuss further. If made mandatory, could then use it as a sorting index and would get rid of length. Couldn't find a user story from past documentation that focuses on coordinate space. Wonder if it was an actual use case driving the decision or a desire to model things that we know exist.
ML: Simplicity of algorithm will help adoption so if we can't come up with a compelling use case for mandatory hash, then keeping it simple is something to consider. (TC +1)

Array Based Structure and Multi-tiered Digest
TC: It's quite important for people to weigh in on these decisions.
AY: Rationale behind this decision - it records the switch from having a single encoding to this models here have independent arrays and it's a multi-tiered approach to generating the checksums.
TC: The idea is to be able to extend the amount of information tha tone provider might want to add to a sequence collection
R: Different information streams rather than being in a file format. Don't see a big file format logging all of this data.

AY: In Nathan's implementation it's possible to use this checksum in same endpoint to return subarray.
ML: Real use case: people modify the * file of the transcripts and they might share it around the lab. Extra info is stripped off * now different than a data provider but for certain cases it doesn't matter. Helps to know that the sequences are identical. Would want to know that the sequence collection is identical based on sequence alone. Here's the link to the UCSC / Bioc conflict on mm10 for everyone's entertainment/enlightenment → https://github.com/Bioconductor/GenomeInfoDb/issues/27#issuecomment-871282798
AY: The mechanism by which we do the digest. Depending on the mathematical operator you use, doesn't matter what ** you use**. But it is doable.

2021-06-30: Refget

Chair: Andy Yates / Tim Cezard / Nathan Sheffield
Attendees "Name (Affiliation)": Rob Davies, John Marshall, Daniel Cameron, Sveinung Gundersen

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Propose a canonical order.	Sveinung	
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda • EDI video	
2.	Previous Actions Review • Write array-based method into decision record, including the multi-tiered digest approach. (Nathan) https://github.com/ga4gh/seqcol-spec/pull/14/files • Finalizing list of allowable attributes (https://github.com/ga4gh/seqcol-spec/pull/12)	
3.	A.O.B.	

Minutes
AY: Just presented Sequence Collections to VRS/VCF meeting. So we have some actual consumers in the meeting today.
NS: Typed up array-based method into decision record: https://github.com/ga4gh/seqcol-spec/pull/14/files

Settled on this format for how we would organize the data that we're going to digest. This is my attempt to specify the decision. Please comment on/approve the pull request. **1) attributes and sequences would be units and collection of those would be the data OR 2) individual attributes are their own arrays
JM: As someone who has just reread the proposal and wasn't aware of this change, going to need to be explained. Seems a surprising decision. Spec will eventually need a good paragraph explaining why this is useful.
NS: Can probably add more to that. A few advantages to this approach is that it adds backwards compatibility. For example if you want to add another attribute (e.g. topology) or whether you want to add something in the future (e.g. whether it's a primary or secondary sequence) felt easier to do if they were separate arrays.* Another rationale is that it would allow us to more easily mix and match.
JM: Adding a new sequence to the collection becomes a harder operation. Claim that it's a less common thing to do.
NS: Either way it will be a new sequence collection with a new identifier.
AY: Becomes easier to add new elements and supporting third parties adding info into a collection**Becomes a question of serialization and how you do the digest for that. Definite advantages to this format over other ones.
JM: Certainly in favour of being able to query for something based on different subsets of attributes. Just think that it will need a certain amount of*Needs to canonical ordering which will probably be mentioned elsewhere.
NS: Talked about making that alphabetical. Ordering of arrays vs ordering of elements. Arrays would be alphabetical. Maybe today we should talk about ordering of elements. Order of elements in the array is preserved. Proposal is that now we are doing array-based, it opens up the possibility of order array. Would order components of the array**Wouldn't have to store as many digests for ordered stuff. Allows you to more easily identify if two sequence collections have the same sequence but in different orders.
AY: Rob brought up being able to calculate order independent checksums
RD: Could get away with exclusive ordering them together.
NS: If you had an order array that would not be necessary anymore. Order them by sequence length.
RD: Would need a tie breaker.
NS: That's how I think of genome assemblies being ordered. May not always have sequence digest, but will always have length.
RD: Do we know how many instances we've seen of permutations of the human genome?
DC: At least 6 commonly used ones for 19 and 38.
**
AY: Sequence report with what I'd call biological ordering.
https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/001/405/GCF_000001405.39_GRCh38.p13/GCF_000001405.39_GRCh38.p13_assembly_report.txt
Should probably assume that if there is a lack of any ordering**
NS: How are we going to order the digests? Length with a tie breaker makes sense.
RD: Have we made name compulsory?

JM: Another tie breaker could be alphabetical by hash. Assuming hash is compulsory.
NS: Get's into what is required. Only two possibilities is just lengths or lengths and orders. The sequence hashes are not required.
SG: Wondering if it's useful to have lengths without either names or sequences.
NS: If you only have lengths then the tie breaker doesn't matter.
RD: If you had names or tie break then you would ** two orderings.
NS: There needs to be a priority. E.g., sort by lengths and if there is a tie and you have name defined then it's by the name, but if name is not defined then it's by the sequence digest.
NS:Sequence digest is less important than the name
AY:**
JM: Wouldn't it be better for the canonical ordering t not be dependent on whether you call it ** or 1.
SG: Tie breaker gets broken by sequence digest. And given they are the same then name is required and will be the final tie breaker.
JM: Don't see it as needing it to be required.
SG: Given scenarios where you have two sequences twice you have no way to break them.
NS: If two things really are identical then the order is irrelevant.
There needs to be a priority of arrays. Do it systematically instead of imposing how we think of things. E.g. Alphabetical order of arrays. Then the arrays you have doesn't matter, it is algorithmic.
SG: Sequence as second tie breaker and * as third.
AY: Agree that we should impose lengths and if further discriminators are needed then Nathan's suggestion...trying to come up with the most logical way of sorting the arrays. Will mean you don't have to worry too much about order affecting checksum. This is how we expect collections to appear for the most part, but they may differentiate…
RD: Problem might be when you introduce another field and ordering may flip.
NS: As long as we enforce that names must be unique then you can stop there. If the names have to be unique then it doesn't matter what else you put in there. You can establish a guaranteed order by the names. Although you may not have the names.
SG: When sequences are the same, you can specify that you need the names.
AY: What names becomes is a unique identifier which is assigned by an external authority. That is the field that identifies the instance you are dealing with. Would be a fair constraint on names. Names can be unique, but it could make things more complicated. Would also mean that you couldn't have metagenomics experiments that**Probably good practice in sequence collections.
RD: Might have to experiment on this one and see where it works and where it doesn't.
AY: Can see something similar happening here that happened around GFF where semantics around a field are misinterpreted. As long as semantics around this are sorted then it should be okay.
NS: Can go back to earlier way of thinking so that order is not its own array.
RD: Think that might be simpler. Could enforce an order yourself.
SG: Once you have this first level, then that would be enough.
NS: Can we imagine a scenario where there are duplicate names?
RD: Would be good to ban duplicate names.

JM: True, we do already ban them in SAM, but it does occur occasionally.
NS: Duplicate names are not allowed or is it more complicated? E.g. we can accommodate duplicate names sometimes but not in others.
AY: If someone comes in with duplicate names then it speaks to a larger issue with incoming data.
SG: If they have duplicate names they don't have coordinates.
AY: might be easier to say name is a discriminator, name has to be unique, there is no API.
NS: Is there any use case where someone might want duplicate names? If so, we can decide we don't support that use case.
AY:**
SG: If you query at the second level you will be able to extract the coordinate system.**
AY: No reason why someone couldn't take the ideas that are here, format it into their own schema and eliminate any**
SG: Are duplicate names going to be disallowed? Are we going to have an order array? If we do what is the canonical order that we are indexing into? First by length then by sequence if it exists, then by names and because names must be unique that solves it. The only situation where it isn't solved is where you have length but not sequence digest or names. Sveinung to rewrite this.

2021-06-22: Refget, Reverse Lookup

Chair: Andy Yates / Tim Cezard
Attendees "Name (Affiliation)": Rob Davies, Neerjah Skantharajah (GA4GH)

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

Global Alliance
for Genomics & Health

	Agenda Item	Person/Time
1.	Welcome, Introductions	
2.	Discussion of User Stories • Working_doc_Reverse lookup User Stories	
3.	A.O.B.	

Meeting Recording

RD: Some data may want to stay private. Data security issues with reverse lookup,
TC: May want to add a flag to data that wants to stay private. Consideration for implementation (AY +1).
EBI is interested in reverse lookup because they are looking at something similar. Was hoping that some of the servers that contain refget could have**and not have to recalculate the MD5?/GA4GH digests. Already have a half finished implementation of reverse lookup
RD: If you are serving metadata then it should be relatively easy**
TC: Not quite clear what the different implementations of refget are currently serving. E.g., the CRAM registry is not serving much. Can continue to add specification to reverse lookup on EBI? side without having to involve GA4GH too much. Can come back and present final product to GA4GH community. Means that specification will be defined by us as opposed to the community.
TC: First use case is straightforward and general. You want to find something by attributes and provide as few or as many attributes as is currently supported by the server. The second one is the slightly more specific one where you have something that should uniquely identify the digest. There's a question of whether it's a different use case or not. If this was available, I would have all the use cases I need. Not sure if there are other use cases that are useful for anybody else.
AY: Agree. Beyond these two, I don't really have a use case.
TC: Was trying to see if the use cases for sequence collections would apply (e.g. compatibility), but it becomes complicated.On a refget server you are requesting a subset of sequence, but the other way wouldn't make sense.
YA: Question of alignment. Would be doing a sequence alignment search to find a candidate. Identification of unique candidates**
RD: Don't think we want to go there.
TC: As far as I'm concerned, these use cases should do it. Happy to leave the document as is. Can move on to potentially defining the list of attributes that it will support. We should do the same kind of thing we were doing in sequence collection where we list the ones we want to support. Not sure if we can/want this list to be extended. Could have a list of predefined names and say these are the ones you can implement for your reverse lookup. And some of them can be mandatory.
AY: This is where it starts getting complicated. There are fields which are specializations.***

TC: We could make all of them optional. If you have name and you say you want chromosome 1, the only thing the name provides is a definition of what name means. Doesn't necessarily say that name is a required, could just go by length. The fields don't necessarily need to be mandatory. Don't know if we can make any of them mandatory. Should a reverse lookup server support queries by length* Can assume that everything will have a name.
AY: Name could mean human readable*
TC: Do we want to allow multiple values for each of these?
AY: You could somehow coe to similar conclusions if you look at Reece's implementation of this. His** enabled the second implementation you described. It became easy to differentiate exactly what you said.
TC: If something is ambiguous the server should return all the results paginated.
RD: Up to some reasonable point.
TC: think it's useful if you have the possibility of saying give me chromosome 1 of homosapiens to potentially get 20 responses. It will highlight the complexity. The question of name is complicated because we have two choices 1) break down the name and be specific 2) allow name and need to provide context about where that name is coming from or you can provide whatever and the server needs to resolve what this means.
AY: https://github.com/biocommons/seqrepo-rest-service - Some examples of how this could work from Reece's work
TC: Could make name a specific attribute that we advise people to implement based on query. Will make the search easier.
AY: If we can support that query, that will make a lot of people happy. It doesn't help the situation where you know taxon, adn species and you still have to go in and search on that. If we can have situations where we can turn name into an identifier question**that's what Reece has done here. Turned name into ID issue.
TC: Does make an assumption that there is only one thing called**
aY: if your data is ambiguous it becomes difficult.
Forcing name to become query, it becomes difficult to say for example I want to have all the chromosome 1 from homosapiens. Also means that if you want to search for refseq succession you need to know that it's a refseq* For my personal use case, this [Reece's work] will be plenty. Length is a proxy for defining whether the genome is compatible, if you have exactly the same chromosome length it will likely be the same*
AY: Gets us back to competing GRCh38 in the wild.** If we progress down the avenue that Reece laid out, he made no difference between coming in with checksums and coming in with identifiers. Anything that came out of alias array, you could go back in to get that information. It's a strength, but like how refget says we are only using checksums**
TC: refget and reverse lookup server can be two different entities. Not necessarily easy, because on our side, the server we currently have for resolving** don't have digests. Unless we go download all the sequence computed digests, then we can't.
AY: Anything that we're getting is piggybacking off of what happening in**Are we missing anything from the CRAM spec?
RD: CRAM generally wants to go the other way. Usually have your reference or digests and you want your digest back.

TC: Will write up an issue around this issue of whether we should use queries for the name. Worth discussing with the wider community. Think we might want to keep the reverse lookup simple and keep a generic name. The implementer can decide what they want to put under the name. If we start forcing implementation to provide query then the existing refget servers are going to struggle to build that. If we say give me the name you already have, then that's a little easier. But it's worth discussing
AY: Also talking about the future of this group. Looking towards the amount of engagement. Maybe this is a use case that with our EBI and Sanger hats on, we recognize, but the rest of the community doesn't face it. Need to scope what we're going to deliver. Maybe it would be recommendations on how metadata from refget**should change.
TC: not attached to the idea that reverse lookup becomes its own product. Not sure it needs to and not sure that there is scope. Like the idea of extending the contigalias into reverse lookup so it can be easier to access refget server.
AY: Maybe the contig lookup can be a resource. It's compatible with refget and we use it to improve refget adoptability. Something for the next LSG leads call. In response to group takeup of this, we're open to questions.

2021-06-16: Refget

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees "Name (Affiliation)": Susan Fairley (GA4GH), Rob Davies (Sanger), Neerjah Skantharajah (GA4GH), Sveining Gundersen

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Write array-based method into decision record, including the multi-tiered digest approach.	Nathan	
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Finalize list of allowable attributes (https://github.com/ga4gh/seqcol-spec/pull/12)	
4.		
5.		

Minutes
NS: Finalize allowable attributes. Brief recap - come to the conclusion that lengths and order will be required. Started debating in depth the order array. People are on board with having order array on separately. Also talked about masks and types of masks. Assembly gaps is one type, and repeat are another. Haven't finalized yet. Tim created a pull request linked above. Lengths, sequence, names, typologies, priorities, order
TC: Talking about attribute without having finalized previous proposal of how we will be using the arrays (using list vs what is currently described in the spec - each entry as one record) and different levels of** a little more complicated to explain attributes without finalized order elements. Order was still being debated a little. Might be good to have special wording re: what order is really used for.
NS: Will write up a decision re: how we will be using the arrays
TC: Have we finalized whether we are going to digest entire array as one go or list of digests separately.
NS: Will summarize array base method and tier**
TC:: Couldn't remember how we specified the mask, but remembered it being complex. Didn't put it in because didn't think that simply putting masks would be descriptive enough. And wasn't sure about encoding of the mask.
NS: Other thing that was proposed that was missing is alphabet.
NS: Want o propose that we defer decision on maks. Unlike all other arrays, the mask are a new order of complexity above all other arrays. Every other array is just a simple array with one lament per sequence. A mask is a different entity entirely. It's a set of regions
AY: Yes, fits closer to being something like a BED format. Noticed that discussion talked about gaps within the sequences.
NS: The gaps is the same as masks, but it's a particular type. Gaps are regions, and whether that region is a repeat or an assembly gap is an annotation of that region. To me, there needs to be a way to specify the mask as an assembly, whether it's an assembly type mask, repeat, microsatellite etc. Given complexity, for both types of masks, we are building something that will accommodate that since it's backwards compatible. In the future we can add masks, but think we should move forward without that. COuld be a year long project itself. Envisioned having a reign for that digest set***
AY: Links to similar thinking re: gene sets.
SG: Agree with Nathan, that starting to talk about region set now...concept of assembly gap is a bit different than repeat masks. Very much connected to actual assembly of sequence. From a mathematical point of view assembly gap defined areas of the sequence. Closely connected to sequence and not**but agree that taking this on now [wouldn't be the best].
NS: Is alphabet an attribute of a sequence or sequence collection?
TC: technically, it is of a sequence. There is no reason why you can't have protein alongside DNA in practice, not sure if this would exist.
NS: All our current arrays are the same length but an alphabet array, does it have an entry for each sequence or is it one entry for a sequence collection?

SG: Feel it's' similar to the circular linear thing. Will get many of the same elements. Only difference is that**agree with Tim that in theory it is connected to the sequence.
AY: If we see a point where there will be a mixing, would you want to compose a sequence collection where the fundamental molecule is different? Should it have capacity to change?
TC: Do we want to forbid it?
SG: Alphabet means the possible characters that **can** be there. Thinks it should be there*
AY: Alphabet would point to what IUPAC* comes with certain amount of biological meaning. Think that alphabet combines two concepts together. It's biological meaning behind the sequence, and that in itself the potential**.
RD: Encoding maybe. If you restrict it to one type for everything in the collection you will regret it as soon as you want mixture.
AY: For a collection of DNA you could define it in a sparse array manner. Overall ability to say sparsely it is related to this element.
TC: Could probably be useful.
SG: Think that the name alphabet catches the meaning. For instance in Norwegian we have letters that aren't in English**
NS: How frequently will alphabet be used. If we're defining a bunch of region sets will most of them have alphabets or not.
TC: Guessing that more than the proprietary of sequence collection it's very likely that you will have implementations that contain all.. In theory, quite fairly**
NS: ACTG vs ACTG + ambiguity character → which is the alphabet that was considered
AY (chat): IUPAC brings an interesting issue … RNA and DNA are the same. So biological molecule isn't encoded in the strict IUPAC codes bar nucleotide & amino acid. See http://www.bioinformatics.org/sms/iupac.html
Got ACG and that was from a piece of DNA, mRNA or a peptide. The alphabet would allow you to distinguish between peptide and the other two but not the mRNA and DNA with alphabet alone.
TC: Doesn't tell you the source**alphabet in itself is useful, but doesn't answer all questions. If we introduce alphabet we need to stick to a definition of what it is.
SG: Model behind sequence. Typically you would know what to get out of it, most people should know that if not they should figure it out. What I'm thinking of is in line with what Andy mentioned.
RD: Molecule type (mRNA, DNA, peptide etc.) and then insert alphabet from that.
AY: We could look toward sequence ontology as nomenclature to use. Could strictly define within our spec, if you're encountering this type of molecule you must encode this using this type of molecule type. If another molecule type comes along that someone wants to encode in that same concept they should be able to. We should just say that your compatibility with out refget services will be hampered if you do that.
NS: Need to keep in mind that we are creating a flexible enough system that they can use it for whatever they want. Right now, we are walking this line between what do we make required and how standardized do we make it. There's some grey area. Probably useful for us to provide recommendations because those will increase compatibility.

TC: If someone wants to come up with a new attribute that should be fine. The more attributes we add the more restrictions we impose.
SG: In agreement that we should not extend ourselves too much. If you've written script and want to see if the sequence can be used**would like to know in advance that this sequence follows that. Still wouldn't be a summary of the actual sequence, but rather what could have happened. What can your machinery produce in a sense.
NS: See value in including what is in the actual sequence.
TC: Adding what is currently in the sequence is an interesting piece of info but don't think…
NS: It's exactly like length.
TC: Lengths is only useful when sequence is omitted.
NS: Second value of length is it might be useful if you have the sequence but don't want to look at it. Could derive value from an alphabet.
AY: Will be used in downstream formats to validate the integrity. In a BEd file would want to exclude sequences that extends over the region. There are other things downstreams that would be useful for *lengths*
NS: Molecule type and alphabet are different.
TC: Do you want to have a different digest?
AY: Personally, yes. They are not fundamentally the same. Because arrays are starting to bring biological meaning into it, if you had a collection and one was regarded circular and one was considered linear would we want those to come out**(NS +1)
TC: Should I add alphabet and molecule type. With reasonable definitions could make sense.
NS: What are the possible values for alphabet? One per sequence or one per collection?
TC: For the sake of simplicity, one per sequence. Values for alphabet could have controlled vocabularies. Could just call them nucleotides and that's it. Same thing for circular.
NS: Do we need them separate? Come snack to use cases. Do we need a separate molecule type and alphabet?
SG: Not against molecule type but it opens problems of controlled vocabularies and ontology.
TC: Nice to be able to tie back sequence collection to it's source. Seems like ontologies are going to come up and we need to address them. As we add more concepts, it's bound to happen
NS: Agree. Maybe the correct direction is the opposite. Define specification for how to create sequence collections and we're not defining what's upt in it. Because what's put in it need to connect to ontologies. What's our role? Maybe we should just provide the specification and the other things can work.
AY: Have to remember that the target here is an unambiguous description of at least the two human genomes. So there is a reason here to say that someone is going to have to resolve how that encodes. We are the right people to make the decision on how that encodes. We expect this to be used in VCF files. So there is utility in us doing this with what's coming down the road re: BED and VCF. Probably makes sense that we split the two concerns off from each other. On the ontology side, in practice many users of bioinformatic formats don't like ontologies. The simpler we encode the data the more likely it will be that people take it up.
https://www.wikidata.org/wiki/Q430258

SG: Agree. Topology represents going into the area of biological function and ontologies. But for me this is a representational/modeling issue. It's about modeling of DNA rather than biological content. Different to manage without certain letters. Still outside scope of biological function and ontology.
TC: Do we want to reserve the vocabulary? Prevent people from saying that molecule type means anything else? Do we want to encourage people to use it? It won't be mandatory. Can add it and define it clearly. Think it makes sense to have alphabet. Wouldn't mind having molecule type either.
RD: SAM is using DNA and doesn't say anything about alphabet.
NS: Proposal: specify an alphabet attribute that twould have an array of length the same as the Sequence. The values are uppercase sorted characters. Could be whatever you want them to be, they are basically the characters that could have been in the string. Wouldn't require ontologies or fixed terms. RNA vs DNA will have different digests now. From a practical point of view it's a simple solution that solves what we've been talking about.
SG: My suggestion for alphabet is just e.g. 'ACGNT', 'ACGT' , 'acgtACGT', etc.
NS: But wouldn't work because we are using refget digest.
NS: Other nice thing about this is that we're not imposing anything. E.g. you could say my string will be sentences.
AY: It's biological relevance that is important here. Want to make sure that biological designation is something we are retaining.
SF: In what circumstances are we expecting someone to look this up and say I want the protein version of** expect people to know species and what molecule. In terms of things we're expecting people to come to refget with, to what extent do we think [this will be useful].
AY: **

2021-06-02: Refget

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees "Name (Affiliation)": Sveinung Gundersen

	Actions Arising (label SC/RL/general)	Assigned To	Deadline

1	Create PR for defining the list of fields predefined and their definition: Mandatory fields could be only lengths and that would be the field we would order the collection on. Order might be made mandatory as well.	Tim	
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Compatibility Function https://github.com/ga4gh/seqcol-spec/issues/8#issuecomment-805843590 - Required arrays: lengths, and maybe order. Adding order array would: - Make it so that it doesn't require the compatibility function to answer order identity questions - Possibly save some space in the database, because we'd only need to store sequence collections in one order; but this is offset by the increased cost of storing an order array for every collection	
4.		
5.		

Minutes

NS: What arrays are allowed and what arrays are required?
NS: Allowed component is pretty open in the sense that if somebody wants to add stuff in, it's kind of they can
TC: There's still this question of control vocabulary. I think we should have controlled vocabulary, just to make sure that people know what to use. So that gets me to naming them.
SG: With naming also comes specification of contents.
NS: An issue about this:
https://github.com/ga4gh/seqcol-spec/issues/8#issuecomment-805843590
6 arrays: sequences, names, lengths, topologies, priorities and masks

SG: Example of masks that was useful for me was gaps (https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2438-1)
TC: would be tempted to try to keep that list relatively short at the moment with the idea that we could add more as required
NS: eight proposed arrays: sequences, names, lengths, topologies, priorities, order, alphabet and masks. Under masks, we've thought of two different specific types of mask, gap mask, or repeat mask. But really, maybe we see it more as like, the masks could be a list of arrays, or something.
TC: that's probably the reason why I would not go too far in definition of these things, just because you get into areas where I'm not sure that will be implemented. On this question of of mask, I would prefer if the name is the actual controlled vocabularies as descriptive as possible. I wouldn't want to use mask for more than one thing (gap and repeat).
TC: technical question, are these case sensitive? I think we should be case sensitive. It's not a bad idea to say that because JSONs like that most of the time.
NS: Would any of these be required?
TC: What I was thinking of doing is to try to nail down the list of fields. So we can at least agree on that and then we can, you can start talking about the the which one are required.
NS: sequences, names, links, and topologies. Those were like the original four, right? That seemed like they're all allowable. One thing that we need to figure out is how do you encode topologies? Binary string? That's zeros? Or ones? Or is it a string? Is it an array of zeros? And ones? Or is it an array of strings that are linear and circular? And are the only options linear and circular? Or are there other options?
SG: I think it makes sense to have them as strings, in order to to allow for other people this personal or craftiness?
NS: so we would say it's just an array of strings. And you can put whatever you want in, but then we're going to define that the term linear and circular, are controlled terms. But really, there's no reason you can't use it with whatever terms you want it and in the future, we could add, and actually, the graph is a good idea, because you could imagine if somebody created a graph genome, and then as long as you had a way of diet of creating a digest out of that, you could use sequence collections exactly how we're using it in the same way. And you could just change the topology to graph to say that digest refers to a graph genome instead of to a linear genome or to a circular genome.
TC: as graphs become more prevalent, it's going to be important to be able to know what the sequence we're talking aboutis a graph or not. I almost feel like you shouldn't actually overload topology for that. You should have a separate field.
TC: Topology should be an array, but not mandatory.
NS: ALmost think that nothing needs to be mandatory
SG: Suggestion to have length as mandatory and combine it with either name or sequences
TC: I feel like the order is shouldn't be a visible array. Another thought is that the order that you want to store things in is going to depend on what is actually mandatory. So if you say the length is mandatory, and needs to be in every single entry that you are sending, to or storing, then you could order based on the length, because it's always going to be there. If you do it on the

sequence, then sequence always needs to be there. So you do need to have something mandatory?
TC: Does feel like we need to have length mandatory. But it also means that you cannot do a comparison, or you cannot store a set vector of names by themselves.
TC: back on this order thing on the other thing I was it makes sense then to store the order separately. So enforce the order internally, and store the order as a separate array.
TC: I thought order array was required.
NS: I agree. I mean, I guess I think if order is an array, it probably makes sense for it to be required, because then you avoid this problem. I was just saying. So then the downside of this is just that we're kind of increasing the complexity, right?
TC: so if you have both, if you have those to digest, you're fine, I think, because you can, you can prepare at at the, at the lower level, I guess you have a problem. If you if you if you have a top level digest in your file, and you want to retrieve the equivalent, ordered in a different way. And your order has not been registered anywhere. But that's always going to be the case anyway.
SG: And I mean, the issue with the current situation are worse right? Because there's just one ordering that would be registered anywhere.
TC: we've been talking a lot about the compatibility function, I feel like there's a separate function that is, is starting to emerge, at least in my head, which is almost a search function, where you have your, your, you're finding all of the slightly compatible sequence collections. So the one that, for example, we have the same, exactly the same level one length array. Hard to say exactly the same. length and name array.
NS: Yeah, this is kind of what we talked about a lot, two weeks ago. It's like an API for identifying possible top level digests that are compatible with your given digest.
TC: Yeah. But yeah, it's not, it's not going to compatibility function that doesn't give you all of the different possibilities. It doesn't use a subset gives you all of the different, more complex. It's an identity level, but it can return more than one.
NS: From my perspective, it does actually increase the complexity of the implementation. Because now you require the ability to not just one way look up.
SG: It could be optional
NS: that might be a good point, because it does feel kind of like an optional. Like, it's a really cool use case. But it is a niche use case, right?
NS: I feel like we're leaning toward the second idea. But maybe it would be optional. I feel like we've kind of settled that length would be required. And then for the orders, it seems like we're positive on the idea of order being a separate array and would probably be required.

2021-05-25: Refget, Reverse Lookup

Chair: Andy Yates / Tim Cezard
Attendees "Name (Affiliation)": Jeremy Adams, Susan Fairley, Neerjah Skantharajah

	Actions Arising (label SC/RL/general)	Assigned To	Deadline

1	Start a use case document	Tim	
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions	
2.	Integration of Refget digest in file formats hts-specs #551	Tim / 10 min
3	Reverse lookup: review of previous work [slides] ● Goal: how to integrate refget standard into other file formats ● Determine concrete use cases for reverse lookup ○ Reach out to Michael Hofmannand Aaron Quinlan from BED ○ FASP - but might be too early ● Is reverse lookup the right solution? ● Previous discussions around differences between reverse lookup and data connect ● Left off on defining fields	Tim / 20 min
4.	Rescheduling meeting ● Reaching out to groups other than refget? (e.g., file formats) Collaborating with sequence collections group	

Meeting recording:
https://us02web.zoom.us/rec/share/Z9_US79OXVj6g07vjl2r_1Z66g5_48dTqHVhClnaiHUDAfRcTDawN58KSr8u3ThJ.aUvVBq0len2ND_2L?startTime=1621958114000

2021-05-19: Template

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Susan Fairley (GA4GH), Sveinung Gundersen (ELIXIR Noway), Michael Love, Rob Davies

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review • https://github.com/ga4gh/seqcol-spec/issues/10 •	
3.	Discussion on compatibility function (https://github.com/ga4gh/seqcol-spec/issues/7) and which comparisons are worth top-level digests.	
4.		
5.	A.O.B.	

Meeting Recording:
https://us02web.zoom.us/rec/share/od7OnVqwqSo6cHUTJtq5n_JXpFOfbpdNdED1i9Kv8cR7SXo55rv5Aag6szZQNtBA.-xrpuiOr1upirBO6

Access Passcode: refget!GA4GH

2021-05-05: Template

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Rob Davies (Sanger), Sveinung Gunderson (ELIXIR Norway), Jeremy Adams (GA4GH), , , , ,

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Bring issue 10 in as a PR	Nathan	4 weeks
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	

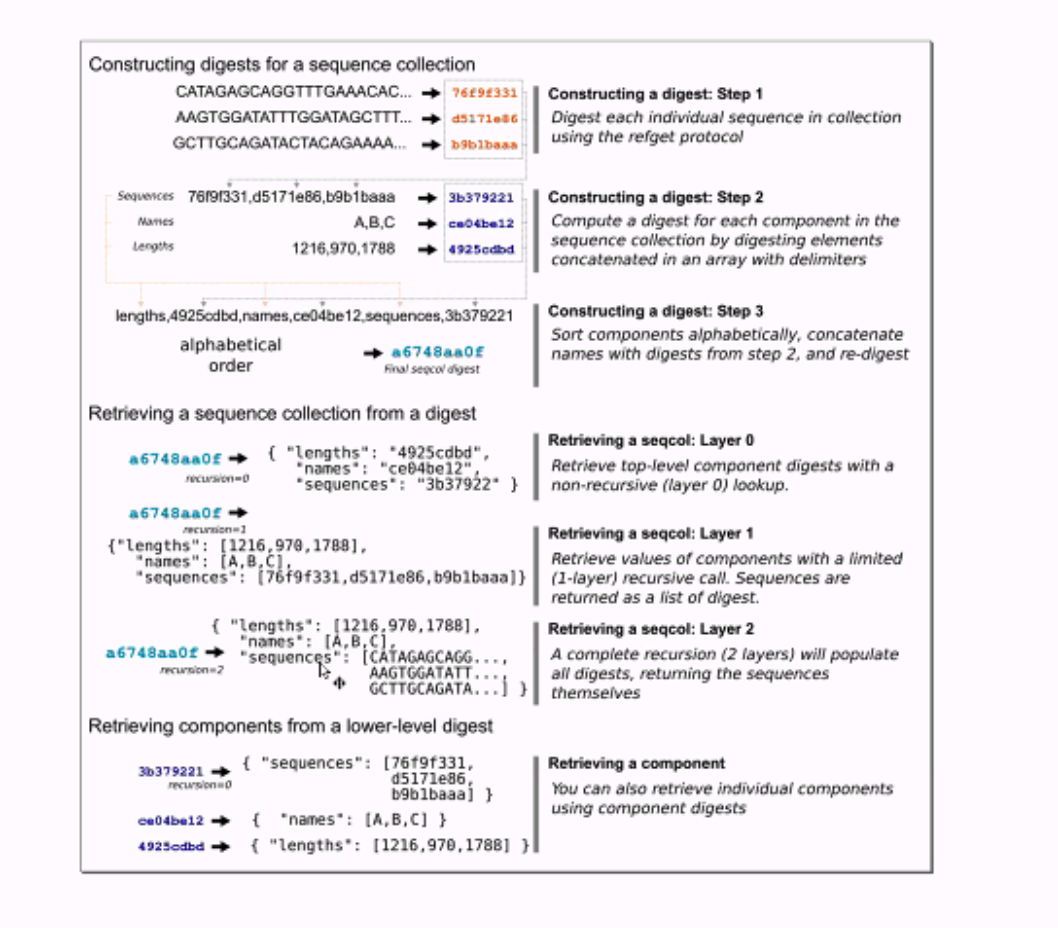
3.	Follow-up on digest calculation issue (https://github.com/ga4gh/seqcol-spec/issues/10)	Nathan / 3
4.	Determine what are the canonical sets/subsets of components.	All / 30
5	Revisit the compatibility function The comparison function takes as input: 2 “level 0” array sets, and returns a series of flags that compares them. As syntactic sugar, you could also pass either 1 or 2 top-level digests instead, and the server would simply look those up and then use the comparison function. But this is only possible if the database is aware of the digest. A “meta-server” could keep track of available servers, and a server could query this server, who could ask peer servers and remember which servers know about which digests. SG: What about encoding order as a separate array? This allows you to have digests that represent unordered sequences, and digests that represent ordered sequences. NS: This would simplify the different digests that need to be stored in the databases for the other arrays, but adds the complexity of having to store an order array NS: The big question is: what comparisons are worth their own identifiers? It’s obviously easier to just compare two digests than to run a compatibility function. But there are so many possible comparisons, that it’s not going to be possible to create a digest to enable each of these comparisons. How do we choose which comparisons rise to the level of importance that they’re worth the additional overhead (either in algorithm complexity or in database storage requirements) to make sure those digests exist for comparison? The alternative is to move away from the idea of relying on digest comparison, and instead move toward the compatibility function, which is more powerful and solves all comparison questions.	
	A.O.B. Refget reverse lookup call reschedule: 12 of May then every 4 weeks	Tim / 5

2021-04-21: Template

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Sveinung Gundersen (ELIXIR Norway), Susan Fairley (GA4GH/EMBL-EBI)

Global Alliance
for Genomics & Health

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Nathan to add digest method as issue, link to delimiter discussion https://github.com/ga4gh/seqcol-spec/issues/10	Nathan	2 weeks
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Nathan to share a concept figure for how to compute the digests  First part defines how the sequence collection digest is constructed and should be specified in the specification. The second part might contain components that are implementation specific. [SF] Is there a substantial implementation cost of making layer 0 mandatory?	5

	Considering that making it optional will lower the interoperability between servers. [NS] there is a cost associated with making the implementation mandatory but not a substantial one. In this construct we would only need one delimiter and we seem to crystallize around comma: https://github.com/ga4gh/seqcol-spec/issues/1 AI to add a proposal for comma in GH issue	
4.	Continue discussion on which information to include in the string-to-digest (issue 8). (what are the canonical sets of components?) [SG]: Should we mint digests for: 1) sub-components; 2) subsets of sequences?	
	A.O.B.	

2021-04-07: Continued string-to-digest discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Rob Davies (Sanger), Sveinung Gundersen, Rick Kim, , , , ,

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Continue discussion on which information to include in the string-to-digest (issue 8).	
4.		
	A.O.B.	

Global Alliance
for Genomics & Health

2021-03-24: Continued string-to-digest discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: , , , , , , ,

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Report on implementation of array-style digest with flexible entries allowed (Implementation at http://seqcolapi.databio.org/).	Nathan
4.	Continue discussion on which information to include in the string-to-digest (issue 8).	
	A.O.B.	

2021-03-10: Connect Debrief

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: , , , , , , ,

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	

2.	Previous Actions Review	
3.	Report on GA4GH Connect meeting.	
4.	Continue discussion on which information to include in the string-to-digest (issue 8).	All
	A.O.B.	

Connect Debrief
VRS/VCF/refget alignment - first step toward underlining the fact that everyone in GA4GH works off a reference sequence at some point, therefore everyone should be aware of it and refget team can work to improve things; came to a head with VCF. Need reference coordinate system
Action:

- Keep in touch with both of those groups (tim - VCF, andy - VRS)
- VCF to make checksums mandatory, not sure which yet, needs input from refget group

Questions:

- Alphabet behind sequence
- Checksum debate
- Genome Providers

2021-02-24: Continued string-to-digest discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Rob Davies (Sanger), , , , , , ,

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Continue discussion on which information to include in the string-to-digest (issue 8).	All
4.	GA4GH Connect and VCF/VRS/refget alignment slides	Andy

5	Refget paper update	Andy
	A.O.B.	

2021-02-03: Continued string-to-digest discussion

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Angela Page (GA4GH), , , , , , ,

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Merge PR#9	NS	Done
2	Send doodle poll to full list based on NS/TC		Done

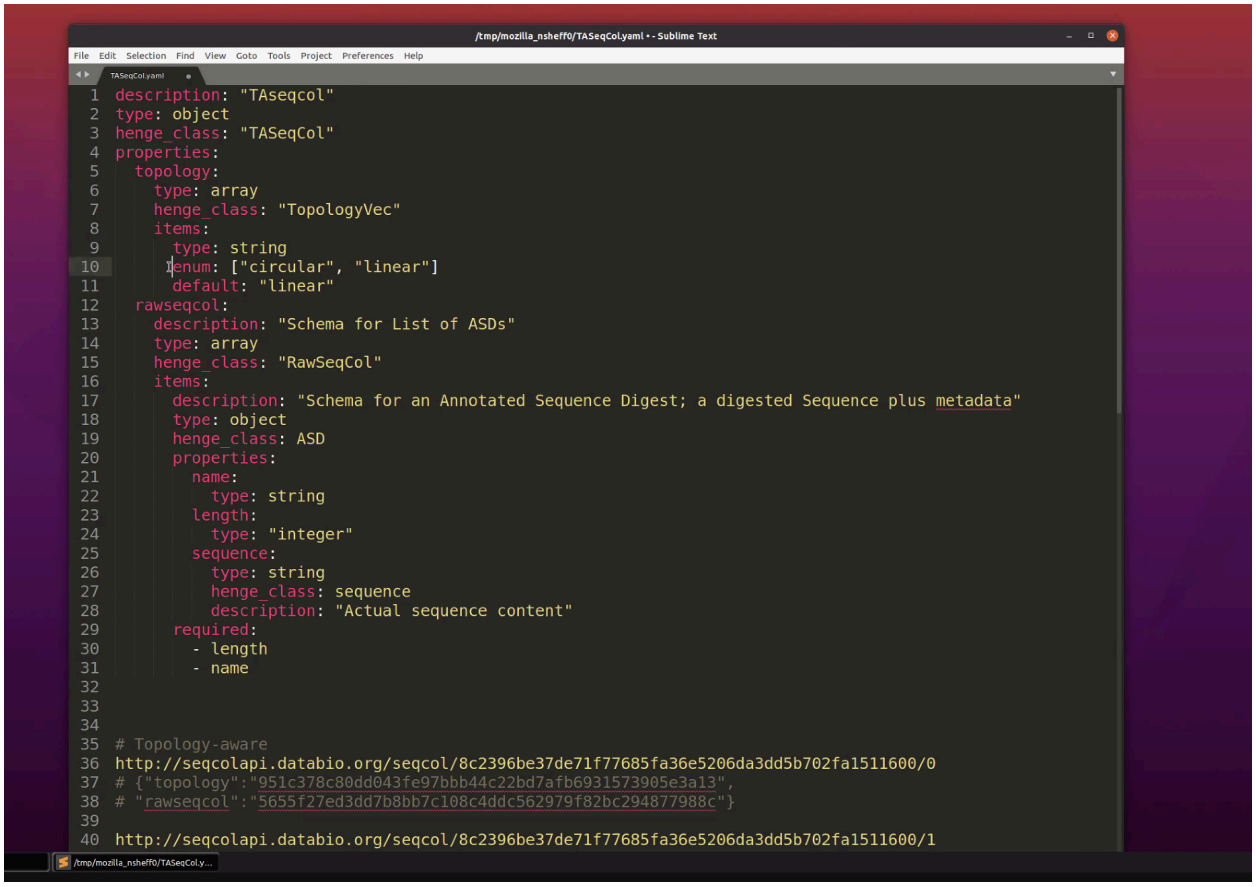
	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Administrative discussion: (a) the possibility of rescheduling the call so it doesn't conflict with RNAseq, and (b) whether the proposed sessions for GA4GH Connect are sufficient for this group	Angela / 5
3.	Previous Actions Review	
4.	Review ADR from Andy: PR #9 (proposal to close issue 2, allowable characters in sequence names)	All / 15
5.	If available, update on TASC Meeting outcomes from 4 weeks ago	Jeremy/John/?/5
6.	Continue discussion on which information to include in the string-to-digest (issue 8).	All / 40
	A.O.B.	

VRS/VCF Connect meeting may be a good place to add discussion on Sequence Collection, do we schedule something separate or do we ask for agenda item on existing meeting?

Potential graph-based reference discussion with GKS, sequence collections, file formats? Maybe too soon?
PR#9 - allowable characters in sequence names - ready to close? Yes, AY happy with changes as they went in. Ready to merger.

Topology
NS: What info to include in string to digest, do we include topology or not?
NS put together on possible demo of how it could work
NS: Topology digest, and raw seq collection digest
Shows three sequences - raw seq coll doesn't have topology, but topology is id'r for topology string, get id for non-top aware seq col or top aware seq col (non-top + top vector)
01 thing on the end of the url? Whether you expand...the layer - recursion level
If I go at 0 I'll just get Id'r at top level, 1 will recursively grab content of 0...as well as array of add'l seq
Name, length, seq digits
3 would get actual seq's from refget
If you recurse to 2 just combine two top level elements together and end up with top mask and seq's that you need
System is super flexible, create id'r for whatever you want, with or without topology; the one with top includes one without top
Use same system to encode things like a mask, a BED file, way to come up with checksum for bed file seq object that has mask slot with id'r that returns bed file - mask seq collection
Have I increased complexity so much to allow this flexibility? Over-engineering? more useful if not this complicated
Build something with simpler interface on top of this more powerful capability that would face users
AY: reviewing where some of the arguments are coming from
TC: I like what you've done; more complicated yes, to some extent more generic in a way and begs question of weather we should change everything to be vectors instead of blob, and separate every element, pick and choose which you want to put and which you don't
NS: yes, great insight' top, names, seq, length - mix and match, pick which you want
TC: option whether or not include top, easier to extend and add new things
AY: don't need to hold onto nested payloads adjacent data, if all arrays are equal in length when assembled, everything overlays and you get same result in slightly different way
Also at back of mind, is idea of intrinsic metadata to optional metadata - what sparked this conversation - is topology essential or optional metadata? If essential, we wouldn't be where we are at the moment, would have rolled into original doc.
Buy the argument that it's not essential, but good to have available.
More in line with idea that topology is taken out as in NS's demo, makes everyone happy, if you don't have to worry about top you don't have to
Extend service to provide hint of alphabet behind seq as well, protein? DNA? RNA? Provide that through similar mechanism, not an intrinsic pattern of seq, only care about it if you want to

Do we want to go all the way into everything is an array? It's not a document, hard to accept!
Inside out objects in perl - good idea in principle, bad in practice….
In having topology mask, we accept you have to synchronize between arrays
NS: that's the challenge, if we want to make optional, you could stick in the array and still make it optional, but have it live inside the array - inside the seq object, name, length, seq + top, mask, alphabet, primary complement or not…
That makes it harder when designing checksums
Here, checksums are defined by type first...are the checksums by type? All top's are checksummed or all info about 1 particular seq is checksummed, and those are together aggregated?
At what layer do you aggregate?
NS: it's using ebi refget api to serve the data, so my server not storing any seq data, just farming that out to ebi; not smart enough to pass that straight along, just getting data, loading, and re-returning it
One other thought related:



Topology and raw
Raw - array of these type - name length and seq
All we need to do is come up with what is the schema for the data that is allowed, that we serve?
This is how I implemented, but that's our task to figure out what we want to do

AY: can see an advantage, going back to idea of each thing being own array, I'm convinced that topology is not a core attribute, happy that it sits alongside in these masks, completely on board with that idea
Nothing in the issue convinces otherwise, some of the other comments on the issue aren't necessarily pertinent; elements in that make me think scope for add'l masks, eg. alphabet, but also repeat...version… just have another set of...bunch of url, actual seq digests…
Another vector you can send info on, very useful - here's the bed file with repeater you can use to mask the seq with, go there, use that - brilliant
See also advantage of having arrays, digest for each array, seq col digest - is the selections you make, add others in if you want
Advantageous as well

TC: in terms of finding the original seq you are looking for, you have the digest and want to look up something, having them separated, topology, seq, length, names all separated, will help with comparison function, bc means that you'll again be able to say… if you know / you might not know ahead of time / which of these values you have, you can just compare them, but you might not know so still compare against everything

NS: but once you know, simplifies things

AY: makes this harder: with original digest, chromosome 1 - if you use arrays to find that info, that would difficult, but should bc have refget endpoint, use that to find metadata bout seq

TC: use case for finding superset, may be harder

AY: starts getting into mathematical operations, intersections or ...knowing...doing intersection and applying same intersection to other arrays at the same time...do intersection based on eg., length and name - everything where same length and same name, also want other attributes to come up with me as well, having to know I found a match on the 9th element, have to….could make some of that gnarly, but ultimately doable, as long as you have the array positions

NS: for user, the way to interact, build api endpoints that return the data in property, object orientation vs. array orientation; so we can implement checksums like that but allow endpoints to conglomerate to return as objects, vs independent arrays

AY: if we go that far that means schema you wrote, changes so that each of those sub elements appears as it's own array

NS: and bc using Henge, operating as backend, all I have to do is change schema and system will work with that data already; can try that easily

AY: sounds like we should push that onto the github issue, present two schemas side by side, assuming ind'l arrays ones work well, that's where we say…

RD: I need to see more of it to see exactly how it's working

TC: I always find a bit tricky is how it will be used when you have a digest and you want to find out what it is, when you look up, esp when the digest you have may not be exactly contained in the server that you're using to look up; comparison function described a couple months ago, super useful for that, how it would work with this new schema, is where I'm having trouble

AY: biggest problem is what you said, not got the exact digest, something that looks like this digest, want to know what I'm actually dealing with; that';s a really hard question; only way to truly answer that is nominating something in pairwise comparison and making a call, if dont' know what to compare against

TC: two kinds of comparisons, one where missing attribute, maybe name, but same set of seq, I think we really need to be able to suppor that bc common use case, also where don't have all seq but same set of attributes, use case that would be hopefully less frequent, more complicated

AY: need to know what to parachute to, as in graph world, always parachute to one point on network and then walk out from there, find something it could be like and then assert whether it is or is not that, nasty search problem

Eg., one I think Tim might be sayiig, if I came in with canonical human assembly 38 said what's seq col for hg38 - in perfect world, if our servers have created same canonical as seq collect, that would be hit, if subsets of real human assembly, with chromosomes, but don't know that coll's checksum, how do you find your way to that how do you know it's your nearest coll? Without doing scout of everyone possible

NS: if you give me a digest and I don't know that digest, I can't help you

AY: it's not this api, something else could be written to deal with that. Need to get list of candidates that it could be and then start to make the call

NS: one thing you could do, give me not just your overall digest, but sub digests within, then I can help you, but I need something

RD: combinatorial problem there

TC: what we are hoping the seq col would be useful to annotate a bed or vcf file, at top seq col digest, for some reason this way created with a particular set of seq's that were using

uncommon name but same seq, want to be able to do that call against seq coll and retrieve the seq nonetheless

NS: only way that could work, server wont' have your actual digest bc you made weird names not in the server, if you jsuyt retrieve with your digest, there's no way back, but if you could give me...if digest is collection of name and seq set digest and top digest, then now I can give you info
If top layer isn't found I can't help unless you give me next layer down, then I can help

AY: if you have vcf header coming in, make… xxx mandatory, then we have a way in

RD: half the reason for storing digest, get rid of chromosomes

AY: both have submitted collections, access to everything, if group written on vcf and have rubbish writer not standard, then playing the sniffing game, then we need to mandate more information, we can't make the call otherwise

NS: I aligned to HG38, which one? Whichever is on my computer.

TC: sounds annoying, but we do make exec decision here to include names in digest, if we were to remove, then suddenly these digests that only include seq would actually match the one in the server. Technically it's difficult, impossible to retrieve, responsibility on our side though to somehow help if we can in this kind of use case bc we made decision to include name

AY: good argument for the ind'l arrays

NS: digest that includes names and one that doesn't, if do ind'l arrays, get that for free, you could use just the seq level digest to query just on name/no name or include the one that includes the name

TS: search by length, eg., hundreds of hits, then seq

AY: NS has action to spec out more and put into issue for review, also put summary of what discussed here onto that issue

TC: since we started from top, it almost sounded like topology….optional attribute then it deosn' have to be included

AY: arrays we mandate need to be there on implementation, arrays that we do not mandate; if you provide it, you should do so this way, follow this schema, want spec to proscribe, if you provide top array, use linear or circular, not boolean 1/0, T/F , certain ways to provide to maintain compaitiblity between implementations

Global Alliance
for Genomics & Health

2021-01-20: Continued string-to-digest discussion

Chair: Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: Rob Davies, Sveinung Gundersen, Angela Page

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	TC to reschedule call with Andy to discuss reverse lookup	Tim Cezard	Postponed
2	NS to close PR#9 Issue 2 in one week's time	Nathan Sheffield	wednesday, Jan 27
3	Decision for most basic "raw" sequence collections to NOT include topology information; additional digest can be added to create a "topology-aware" sequence collection if needed. This will also work for any metadata that is valuable enough to warrant minting of a digest. NS to continue working on demo implementation and share with team at next meeting.	Nathan Sheffield	Next refget meeting

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Review ADR from Andy: PR #9 (proposal to close issue 2, allowable characters in sequence names)	All / 15
4.	Update on TASC Meeting outcomes from 2 weeks ago	Jeremy/John/?/5
5.	Continue discussion on which information to include in the string-to-digest (issue 8).	All / 40
	A.O.B.	

- Completed: Andy submitted PR#9 for architectural decision record of issue 2; NS made changes.
    - Wanting to make decision on issue 2: allowable of characters in sequence names
    - How do we doc decision so in future, paper trail for what was considered

    - SAM spec v1 for seq names, wdier GA4GH working on more universal standard with TASC issue 5 - if such standard emerges, we'll comply, for now SAM
    - Continuing despite known limitations due to benefits
    - Maybe change "weird stuff" phrasing to "disallowed characters", spelling error on "sequence" (FIXED)
    - Tim: looks good, only question is actual markdown file, stored in repo all decisions regardless of how many?
        - Yes, in original pr, changelog format with date of decision
        - One file with each decision with date and rationale, can divide by year if it becomes too long
    - Proposal last week to close issue out after two more weeks
- Continue discussion on which information to include in the string-to-digest (issue 8).
    - Two competing philosophies have come up triggered by question of whether to include topology in digest or not
    - Bigger picture than just topology, philosophy - is it inherent in the sequence or not?
        - Seq collection is just a named order - way a computer sees it; OR
        - Model of biology, in which case topology is an inherent thing
    - Arguments summarized in issue
        - AGAINST: Way we store in computer is same, analysis of whether to treat sequence as circular or doesn't care, related to particular tool/use case, not describing actual seq the way we store it
        - FOR: SAM headers include topology of the field, maybe appropriate to follow that standard
    - Does topology belong in CORE sequence collections protocol or build layer on top that can include these kinds of things, should have way to get the info, cleanest way may be to define sequence collections as most basic fundamental building blocks, ordered set of named sequence with length
        - This allows for topology aware sequence collection
        - Framework allows for other things (Eg., masks, primary vs. secondary, other metadata worth having a checksum around
    - Svein: pragmatic pov - important to decide in a way that is useful, even if it may not be perfect
        - Everything that makes things easier for automatic areas is good
        - Given, ie., analysis tool that does handle circular or linear sequences differently, then you'd like to have that info available somehow
        - Worry that if info is on the more optional higher level, then often that might not be included and you'd get many errors, handle sequences incorrectly
        - And then, what are types of info that you REALLY should know about your sequence collection
        - Topology is something that should be in there all along, if I had a fastq(?) file today I would have included that, bc it changes how you handle the sequence or model of the sequence

- NS: idea of having different checksums for sequence collection that covers different thing
    - If we include topology in the checksum then two seq collections with exact same seq but diff topology would have a different checksum
    - If we bake that in to the only checksum then you have to consider topology
    - If we first define checksum without topology, add in topology if you want to, you'd have possibility of having checksums that do care about topology and checksums that don't care about otpology
- RD: working out cs with topology, stripping off and storing that
    - What do you want to use the data for, creating list of references to go into something that doesn't understand topology, doesn't matter if you include or not
    - If you have aligner that is aware of topology then it does matter
- SG: what happens as a result is you have sam/bam file and that would include the hash or checksum but you might not know what that would be used for by other users, they'd have to go back and figure out whether topology is circular or linear bc it wasn't included, so creating hassle
- RD: if you want to use … for replacement for sam header, diff checksums for topology, otherwise won't get right answer…
- NS: what other things are there like topology that would fit into the same category?
    - Masks, discussed last time
    - Primary vs. secondary, maybe more appropriate analogy
        - Seq collection want to annotate some as primary (eg chromosomes) and some as secondary scaffolds
        - Each seq has a value, circ or linear, primary or secondary, reflecting the biology - core chromosomes vs. freefloating plasmids, scaffolds, not part of core genome
        - Analysis tools would use that (or maybe not, just depends on tool)
        - Primary / secondary fits into same framework as topology
        - I don't see topology as more inherent to the model than primary/secondary designation
        - That leads me to the conclusion taht I think the seq collectiosns hould not include those in purest form
            - Something layered on top that allows you to add these kinds of annotation
            - Useful to have a digest for a pure sequence collection that includes no other information
            - Primary / secondary annotation may not matter to you, just want digest to uniquely id

        - 99% or more dont' include topology, only one aligner I know of that considers whether mitochondrial genome is circular
        - Inherent part of bio model, but those things are more specific to a small use case, which we should enable but question whether it's fundamental to the most basic thing we produce
- SG: another similar kind of data is characters used in the sequence, which is less biology but more computer science, how it's represented
    - It would be useful to ID what kind of seq, what is the scope of sequences we're supposed to handle here? Would it include proteins
- NS: yes, it does
- SG: so the alphabet used is important to know
    - These are letters that are possible here, doesn't have to be letters here, bc if shorts seq might lack a nucleic acid, but still all are possible in a way
    - NS: I like that idea
- TC: I have a slightly different way of viewing the seq collection
- SG: Similar type, less biological, but useful to check that you are able to handle
- Could also have...not sure how often this happens in practice, but variant switch which includes some letters that could be AT not CG - degenerate bases - would be useful to know
- NS: db that storees seq collections could porvide that but not sure its…
    - ID alphabet used in this seq and remember that and provide it ot user if needed
    - OR define classes of seq collections, allowable characters therein, instance of a seq collection adheres to this class of seq collections
        - Eg. dna seq's that do not allow degenerate basis
            - For any given seq collection, figure out whether its' allowed
- Like a schema, but no value in include in actual digest calc of the seq collections
- RD: metadata rather than something to digest
    - Diff seq with diff letters, come up with different digest anyway
- SG: how are digest supposed to be used?
    - Thinking it would replace H319 or whatever wherever it pops up and have htis digest instead
- NS: unique id not issued by a central authority, but verifiable by seq itself, another argument for topology not being included, not of seq itself, someone needs to to annotate
    - If you want to adjust digest, can but makes it tirkcier

- TC: if you go back to original refget, they decided that the useful information to include in the digest was seq itself and seq alone
    - Could have argued that seq itself some info is .. some is not, remove all the casing information
    - Same exercise we need to do - decide what info is relevant to define the name of that particular seq? There's a lot more info we could add and would want to have about seq but not defining seq
    - When you use that definition, you could argue that topology is defining or you could argue that it's not, comes down to personal preference, tbh
- TC: start again from name that will define the genome bc don't see much value adding more info into the name unless there's a really good reason
    - Argue we shoul dhave only the seq's in the digest and any other info that is either relevant to define the seq collection OR
    - That has real added value from a technical point of view
- We should have sequences in the digest (or seq digest) seq definition in digest, any other info has to be justified to define the seq collection or to bring value
- Originally when I first heard about seq collection, first thought was just a bunch of seq digests put together, surprised to learn name and length of seq was included, didn't make sense to me at the time that those would be actually beneficial, we should be justifying why we are including these in the digest, come to topology, should have same kind of justification; if there's a good justification
- NS: if two seq collections HG38 with mito chromosome as circular or linear - are they diff seq collections or are they the same seq collection with diff way of annotating one of them?
    - If diff seq collections, then should be in def / name of seq collection, including topology
    - If same seq collection but with diff annotation/analysis/element, then topology doesn't belong in name of seq, but as an annotation of that seq collection
- TC: I was against the idea of topology but with this argument, there is an argument to include bc it is part of DNA seq to be linear or not
    - Needs to be clearly justified bc imagine there would be both costs and benefits
- SG: you have a BAM file or whatever and you have the digest which is helpful, but you lack some of this metadata (topology, other things) this is data you really need to have in your application or script - how are you supposed to get that?
- TC: if you have the digest then you can retrieve the seq collection (SC) from SC the server, will be up to the SC server to either provide the metadata endpoint will provide you with certain info

    - If you can turn that on its head if you have seq collection, hg38 or actual name of the seq, you're in same position - need to go and look up in genbank or wherever to get that information, need to look up somewhere and not part of the name
- NS: you have to go to an authority, who is that authority? ENSEMBL, UCSC, you go there and they say "we annotate this as primary, this as XYA"
- SG: what is lost is what's happened historically, if you're downstream of a mapper, then the mapper would …
    - Given you have a wrapper that hadnels topology and one that doesn't (don't know who handles topology - GSNAP can do circular
    - You're downstream of GSNAP but you don't know whether that seq has been handled with circular or linear algorithm
    - Lost, but if it were in the hash you would know it's been handled in certain way
- TC: yes and no, bc if you incorporate the topology n the ses collection and someone stores in BAM file the hash that includes topology, nothin tells you its actually being used, info taken into account or just storing htat version
- SG: need add'l info that the wrapper is there, then you'd use it
- NS: I also think that, basically it's the responsibility fo the particular workflow or mapper to record details
    - Agree that makes it so that now it's built in, you could make same argument about avery parameter in every aligner in every analysis, where do you draw the line? Why is this particular piece of info valuable enough to warrant including in the framework, but some other setting in the one aligner that can handle
- TC: completely agree, argument of storing analysis parameters doesn't hold
    - Still struggling with whether topology is an intrinsic information of the dna that is not currently stored in the sequence, in itself means you need info from somewhere to id the real sequence
    - It's only really important if the seq itself is not enough to know it was circular in the first place, probabiily of having two real seq one being circ and one being linear is so small, struggling to find that argument compelling
    - Only one that really means topology is part of id'able criteria of seq
    - When you need to distinguish between two real sequences
- SG: the reason I'm struggling with this is the philosophy and what this means about what we're trying to do here; TC said before that the names

of the chromosomes are included, that confused you - in a sense that is also the same kind of information as topology

- TC: yes, for me names is even less intrinsic to the seq and same thing for length, although length is completely included, redundant info - why we need to justify adding it to the digest and I have been since then convinced that are useful *technica*l additions bc provide a diff way of defining the sequence
- SG: that's a good argument! Coming at this from track analysis world, when you have a bed file eg., knowing names and lengths is incredibly useful and something about the authority, if those are included in digest even if you don't have in human readable way, can easily get the non-hashed, non-digested version, you'd always get this
    - But topology or if those things are not included in digest you'd need to refer to an authority that may or may not include that
    - Then I feel it's a question of how important is this info so we make sure it's correct in a sense, for me chromosomes names and lengths are that important, would def like to have them there
- NS: another practical thing is just that in the way seq collections have been used historically, seq's always have names and they always have an order bc they have to, but they don't have topology; populating this database is easy if we includ seq, length, names, orders bec all in fastq file by def
    - If topology required, requires human annotation to say this is circular that is not, for every seq out there decision has to be made and it will always be linear unless someone comes in and manually annotates as circular bc no other way
    - Don't see way to automatically include non-linear topologies
- SG: that changes this into an annotation proces
- RD: don't embl records have it in there?
- NS: someone's annotated somewhere for some subset of seq
- RD: need to make checksum with there and one without for all the people using it without it bein there, staring twitch fastA file
- TC: 99% of people will use with linear topology regardless of whether actual dna was linear or not, which to me sounds more wrong bc it means there would be a panel of collections that would not represent reality
    - If you bound yourself to only represent reality, end up with many collections no one can use, referring to a diff on
- RD: you'd know if you'd used the linear rep, that's the point
    - At risk of making collection names long, could have base names, version without all the extras (just seq) and then hyphen then bits of digest that gives you more info on things like topology
- TC: interesting concept

- NS: just this am playing around with this idea, if I were trying to implement something, and I have something just about to work, share in a couple times, idea was that I made one digestible data type that does not include topology, digest that put in db, retrieve, mint ID'ers for seq collections without topology, will be used most of the time
    - Created a new data type which consists of one of those and a vector of topologies
    - Then mint digests for that data type (topology-aware SC), builds on the seq collection that I already built that's not topology aware that allows you to digest that …
    - Now I have capability to get a digest for the typical SCs that lack topology but if you have a protocol that requires topology, create service that allows you to use these other digest which include in them the original digest to the "raw" non-topology aware SC
    - Was able to implement and looks like it's working, will share at a later date
    - Adding too much complexity?
- RD: basically a bed map, sparse, all linear or one is circular, then wouldn't take much room, hang off end of normal digest without topology, anyone would be able to recreate the vector and apply it
- NS: doesn't stop at topology, could do same thing with primary/secondar, seq masks, any metadata you want that is valuable enough to mint a digest for it
- TC: easy to get topology aware and find the non-topology aware?
- NS: yes, that's the point if you have topology-aware, gives you back top vector, the bitstring + the topology unaware digest
    - All you need is the topology unaware digest and bitstring, if you want to be top aware, just always use them both
    - Digesting those and putting back into db to give new digest
- RD: doing that is same as creating two versions, one with and one without top
- NS: but it's recursive, bc the one with includes the one without, not independent
- RD: Output is different
- SG: difficult to compare a topology aware seq to non-top aware, unless you have those two digests, you would need to look up one to see if same
    - Like idea that could be divided into two parts, always have main and then extra part so then you can see it directly
- NS: will break down with SC with 20K sequences in it
- SG: having two digest, not necessarily bitstring, so we define core things always should be there and optional extras defined in this way, but still digest so you don't get size issues, but two parts in a way

- TC: these are good tech soln for implementation annotation aware SC, still kind of wondering what benefit you'd have to store this digest in a BAM file vs. one that is not aware, except for convenience of retrieving the information, ignore that not much benefit bc doesn't id the seq any better than the non-aware, bc could also imagine leveraging add'l digest, actually implement metadata endpoint that is quite rich and contains all that info and people will be able to retrieve
- NS: metadata endpoint causes the authority issues though
    - Diff providers can provide diff metadata, if digest it's defined
- TC: but large majority of people, multiple implementations like that, the one from EMBL, eg, would include, UCSC doesn't care, eg., all mito would be linear…
- RD: people might not like that, could use these things in the sq files in your BAM header, if you have diff authorities giving you diff answers, wouldn't be able to reconstruct bam file consistently
-
-
- TC: Reverse lookup - discussed with andy - need to reschedule

2021-01-06: First Sequence Collection 2021

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees "Name (Affiliation)": Rob Davies (Sanger), Nathan Sheffield, , , , , ,

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1	Submit PR for architectural decision record of issue 2	Andy Yates	
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.	Proposal to close issue 2 (allowable characters in sequence names) after 2 more weeks, on January 20, 2021.	Nathan/3

	NS: can we close this out in 2 weeks? AY: need to consider compatibility of tools if we decide to go with UTF-8, maybe better off to be restrictive now to ensure compatibility with existing tools NS: someone in microbial genomics, there any many non-compliant characters in things like FASTA files NS: if whatever we decide solves ~80% of cases, this is better than debating in circles AY: should we record known limitations for any decision we reach in the spec? Sequence names will likely be debated again, so need to be clear about the limitations of decisions reached AY: architectural decision records TC: are we following the sequence name that's in SAM? Or in TASC? NS: we are going to follow the GA4GH sequence name identifier TC: phrasing we are going to be using NS: we refer to a specific version JM: need movement on TASC with this issue JA: TASC meeting on Monday Has there been any pushback on sequence name constraints in either SAM, or GA4GH? JM: there's been pushback on the sequence name in SAM, response was to make sequence names shorter, etc. NS: there's value in simplicity if we make the seq name restriction up front, which seems reasonable	
4.	Continue discussion on which information to include in the string-to-digest (issue 8). Does it include topology? Which fields are optional and which are required? NS: do we include topology or not? Yossi is in favor of including topology somewhere based on what is needed by downstream tools NS: 2 options 1. A seq col is a named ordered set of sequences - topology as metadata a. Circular and linear would be the same collection (1 hash) with different metadata 2. A seq col is a model used to represent reference genomes - topology as a central aspect a. Circular and linear would be different collections TC: if we use option 2, would it make sense to have both a linear and a circular? Does that ever happen (e.g. mitochondrial linear genome doesn't exist)? Therefore, it sounds like topology is metadata	

	AY: if we take the human genome, the mitochondria is always circular, but software tools deal with it as linear, tools don't span the origin of replication. Convinces me of going with option 1 NS: there are aligners, tools that do treat genomes as circular (e.g. gsnap) JA: only need it if we're aware of a genome that's both linear and circular JM: ML: no strong opinion, topology not needed for my analysis NS: believe in small components, in most cases we don't need topology. But we could at some point add a new model, a topology-aware sequence collection to make a new digest. Could build a seq col with topology vectors on top of existing seq col AY: you could flag certain sequences as duplicates, which could be another metadata attribute NS: we are leaning towards not including it, but how do we include it? TC: include in the metadata endpoint AY: circularity to hold in header… NS: topology is included in SAM header? RD: yes TC: effect of adding new fields in the digest on the comparison function? It would add combinatorial complexity NS: once you have the digest, it shouldn't matter TC: comparison func only compares digest, not metadata NS: correct, therefore you couldn't compare topology TC: will this make comparison really slow, or more calculations NS: don't foresee a major problem, would be linear rather than N^2 AY: header reuse, from a SAM/BAM file - if topology is a SAM field, then the checksum must include that info NS: not necessarily, you would just lack a guarantee that the metadata corresponds to the SAM topology AY: sequence collection mask? NS: seq col + metadata/topology vector TC: separate service? NS: that's what I'm thinking AY: based on your analysis you might need to access masks, can expand out the metadata available for the collection	
5.	? Continue discussion on delimiters (issue 1).	
	A.O.B.	

NS: Okay, so I just have a few examples there. Like it says collection a 6748. So that's just a little. If you click on that top one, then now you get three arrays, right lengths, names and sequences. Yeah. And in fact, if you change the one at the end to add zero, then it will just give you the digests for each of those elements and that try changing it to a two then it will the legs and the names are the same as if you had one because those weren't like sort of a recursive step removed, but the sequences are now populated. I don't know if you Does that make sense? Yeah. Okay, so that's something that I have already sort of figured out. But just to orient you, if you go back to the main page, you could, okay, so notice, like, if you look at the zero version, we were looking at the digests. So the sequence digest starts with three B three, right? Yeah. Hi swinging, by the way, are you here? I was just, I'll just catch you up. I was just, I paste it into the chat, a link to paste it again. This is a demo of basically where we ended in the last couple of weeks. So I've implemented it, and this is how it's working. So I'm just sort of exploring how it works. So the third link there says now just the sequences, so if you just give it a three B three digest, then you would just return the three sequence digests. Does that make sense? Yeah. Okay. And then now here's where the sort of the new part is this related collection, which is 2786 e. So if you click on that one, it's actually an identical collection to the first one, but now adds the topologies. So the legs names and sequence digests are identical. But now there's a topologies digest. Yeah. And so similarly, if you recurse down that so like the fifth link there, if you recurse down that to level one, or two, level two, by replacing the the number on the right far right, then you can, in the same way, basically get the exact same sequence collection, but now with topologies Yeah. Yeah. So the way that it works, the way that it calculates those digests is basically the way we sort of fight set determined last
NS: is, you just say. So for example, in the first case, you say, you have to order them alphabetically, so it would be length, the names then sequences. So it'd be lengths, comma 1216, comma 970, comma 1788. Comma, names. Yeah. Actually, that's not quite right, because the way it actually is working is because there's a because there's because it's recursive. It's actually links, comma, and then and then that digest, comma, names, comma, and then that digest comma, sequences, comment on that digest, comma. But then if you go into one of those, so like sequences, that one would be just seven, six, D, five, B, nine, I wish I could be sharing my screen, it'll be easier
NS: Okay, so the system basically operates like this, if you have an array is an array object, then you digest it by just concatenating them with a delimiter. And digesting that. If you have a property object, which is basically a dictionary, right? So names with items, then it operates the same way but you just do name item. Name, item name item. So that's what enables us to maintain the backwards compatibility and allow for future features to be added.
RD: As I was pondering this, before we started, why do you want Ranchi? Yes. Would it be better rather than having these sorts of higher level groupings of things to actually keep all the arrays as separate digests?
RD: As I was pondering this, before we started, why do you want Ranchi? Yes. Would it be better rather than having these sorts of higher level groupings of things to actually keep all the arrays as separate digests?
NS: Yeah. So that's actually what the third link on that page is doing. It's just, it's just passing the sequence digest. Yeah. And all you're getting back is the sequence array. Yeah. So you, you

definitely have that capability. But you also can ask the question of a digest of the, of the set of all the digests.
RD: Yeah. As long as I was thinking more of how many different things you store and so if you if you actually made it the default to access them all separately, Kind of avoids the combinatorial explosion as you get people storing sequences within one hour topologies or within that length.
NS: And I think there's no reason why people couldn't do that if they wanted to just, yeah, it's just that's more than you couldn't like, I guess you'd have to just embed the I mean, you'd basically be embedding.
RD: You'd be you'd be embedding psi or your level?
NS: Yeah, like the one of the zero things like this, you know, meeting this thing, like a set of digests with their names? Yeah. Which is fine. So I guess. Yeah, that's totally valid. But I can also see a use case where somebody wants to look that up. And it's actually not hard to look that up. Basically, we're just storing that in the database. The mapping of Oh, yeah, the the digest of those digests.
RD: Yeah. Now it's more a question of just how many different things you want to store and whether you would end up with this sort of explosion or not? I maybe don't say doesn't matter.
NS: And we had sort of come up with a list of ones that we might include, which is basically like, the names, links, sequences, one and the names length sequence, topologies. One, and maybe just the names length one. Yeah. But it wouldn't have to be everything, it would just be like what people actually had to use for. Yeah.
RD: I know, avoid the kind of explain, I guess you would risk people taking sequences and giving you the wrong names to go with them.
NS: I guess. It's just depends on your use case. And some use cases, that's what you want. And so you could just use the sequence one, but in some use cases, like the names who are actually required for reproducibility. So then you'd want the digest that include both right?
NS: well, they're just mapping, I guess, because it's because of the recursive digests system, at least the way I implemented it. Yeah. There's, I mean, any actual array is only stored once, regardless of how many of these comments are. So the only thing we're storing is a mapping. Yeah. So I mean, I feel like that you ought to be able to store a lot of those before you get to.
RD: Yes. Now, I guess I'm sure if we were indexing, you could probably do sneaky things by name next year on the start of chunks on the men, which will actually reduce the size of it quite a lot. So so you potentially use your database for more than one thing to yourself. And we also have a filter that one wasn't interested in.
NS: I think right now, these are MD fives. Yeah, I think. But I can switch it. I've done them. I've done both. When I like that server is actually holding random data with multiple different checksum identifiers, like, yeah, it's just whatever's in the database, whatever I have been using. It doesn't really care. It doesn't know it, just just give it an identifier. And it stores that under that identifier.

Global Alliance
for Genomics & Health

2021-MM-DD: Template

Chair: Andy Yates / Nathan Sheffield / Tim Cezard
Attendees “Name (Affiliation)”: , , , , , , ,

	Actions Arising (label SC/RL/general)	Assigned To	Deadline
1			
2			

	Agenda Item	Person/Time
1.	Welcome, Introductions & Confirm Agenda	
2.	Previous Actions Review	
3.		
4.		
	A.O.B.	