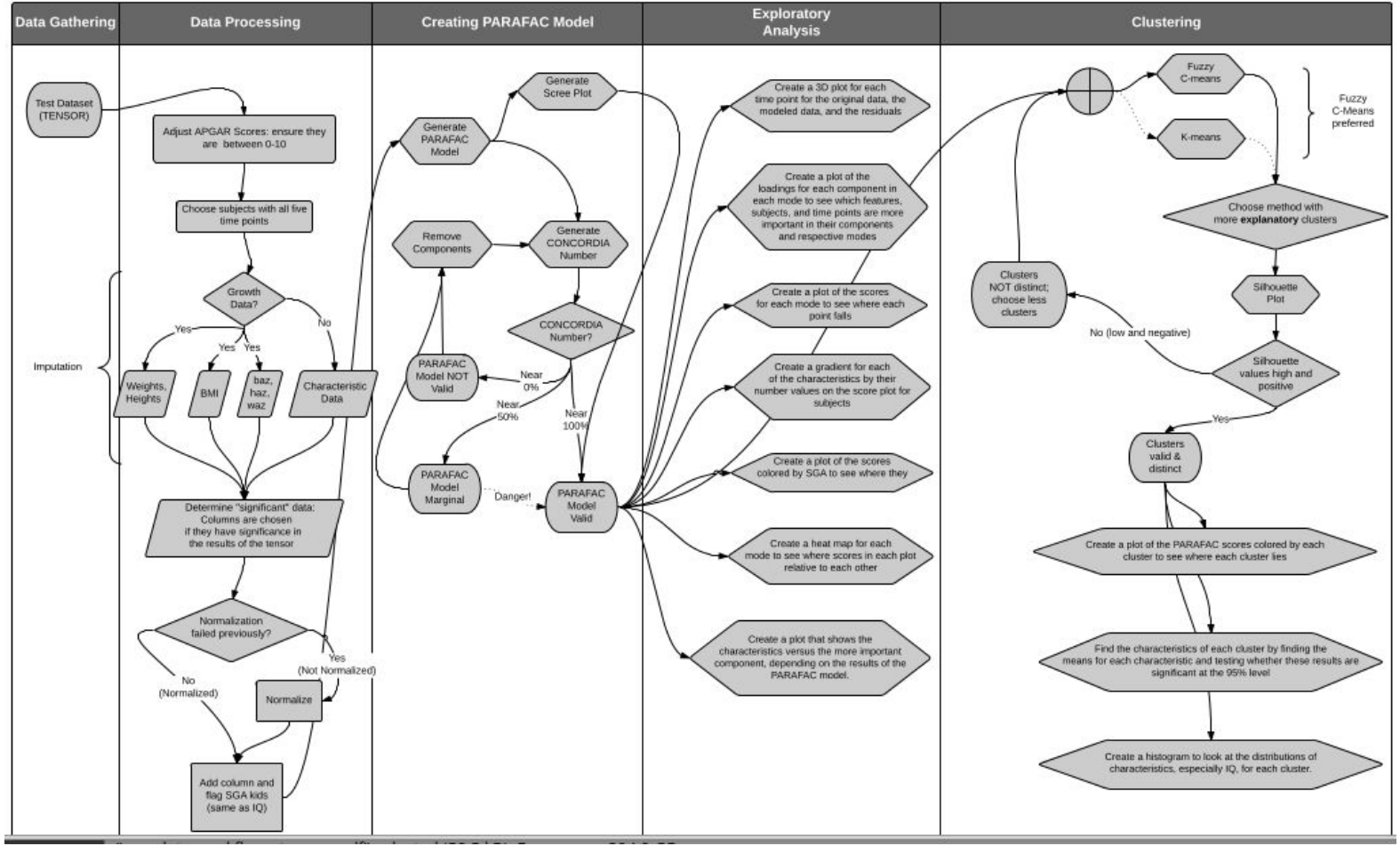
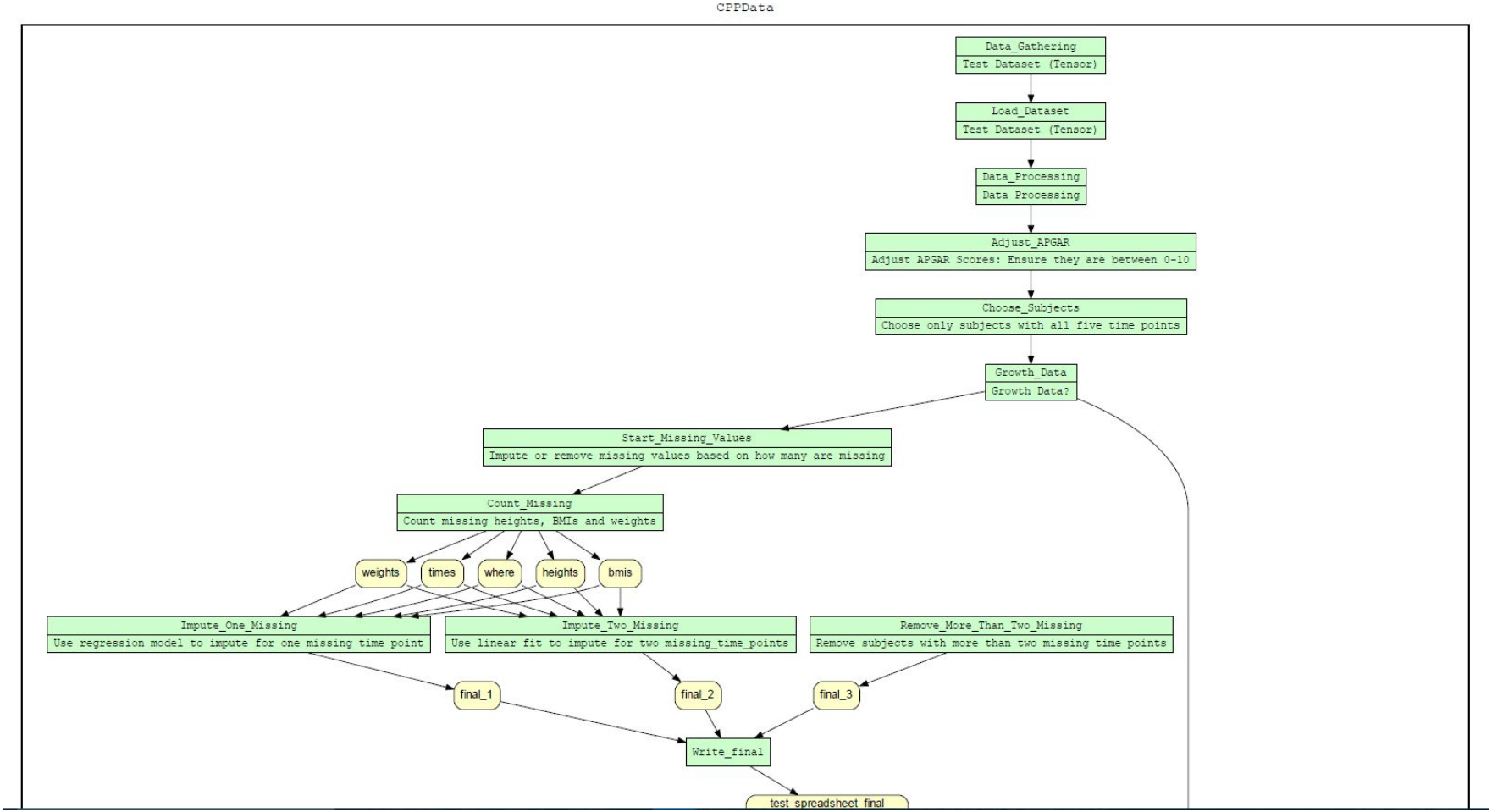
Addressing Scientific Rigor in Data Analytics using Semantic Workflows

John S. Erickson, John Sheehan, Kristin P. Bennett and Deborah L. McGuinness

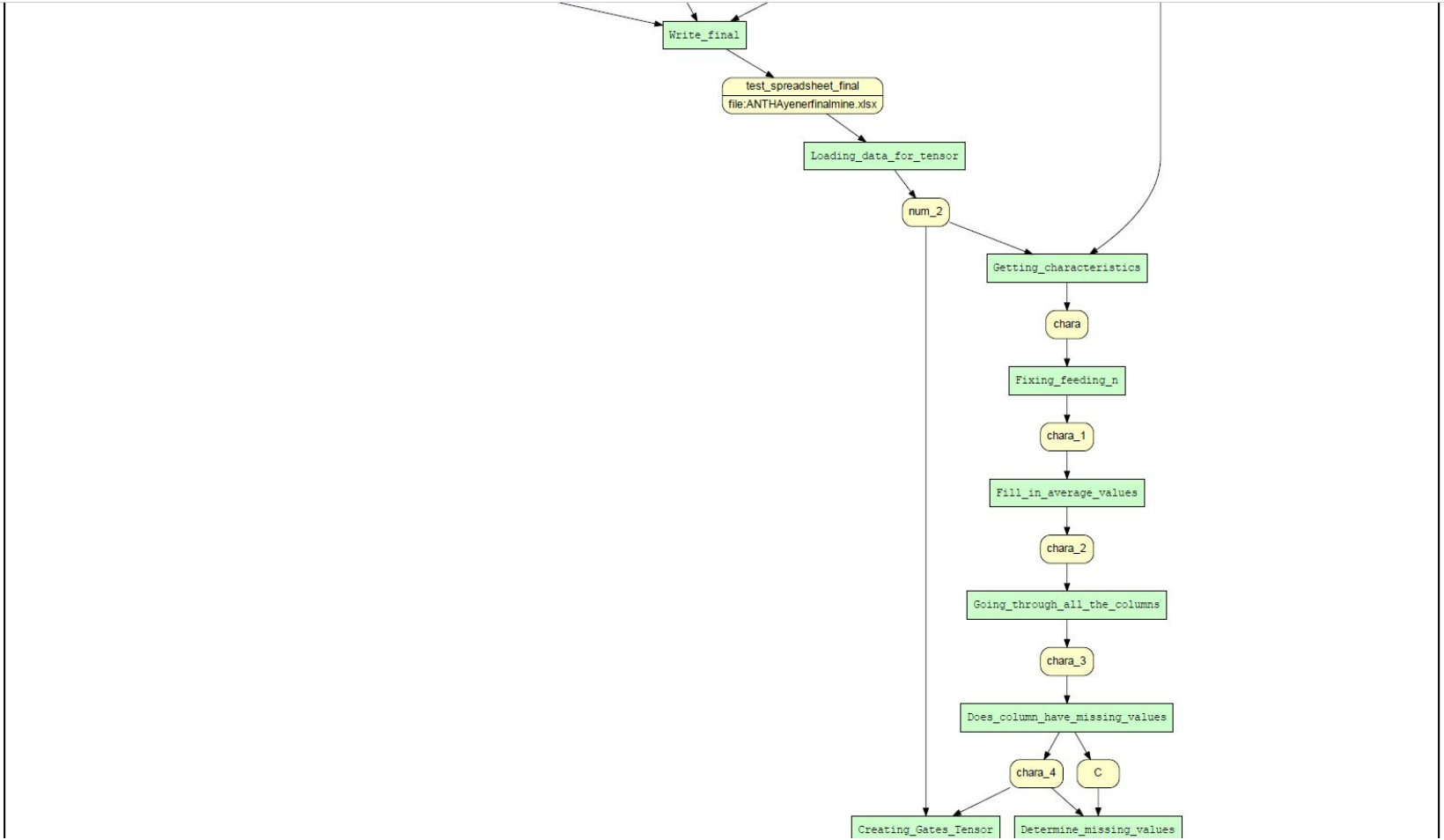
1. My diagram of Hannah's workflow...



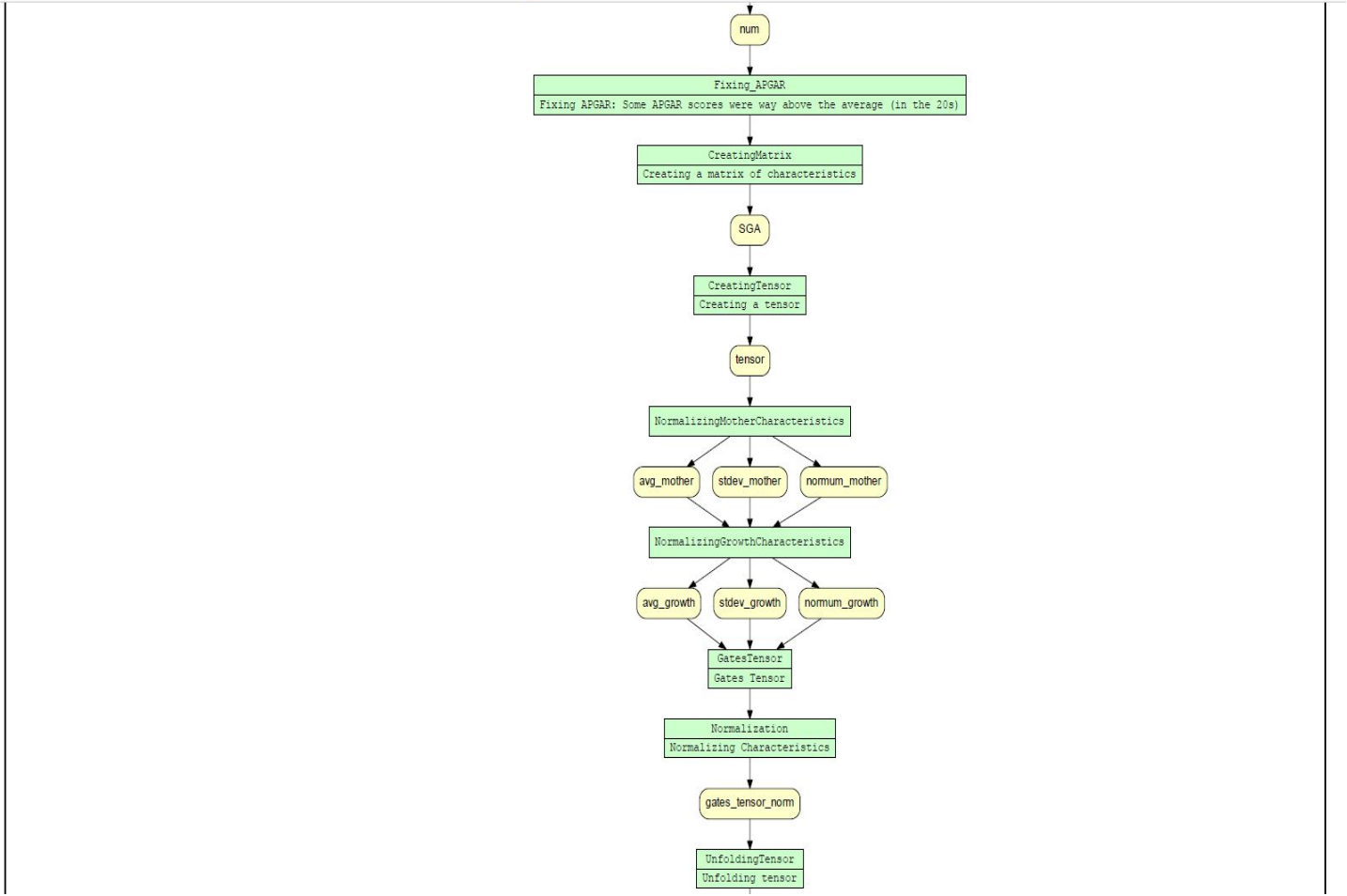
Yes Workflow Representation



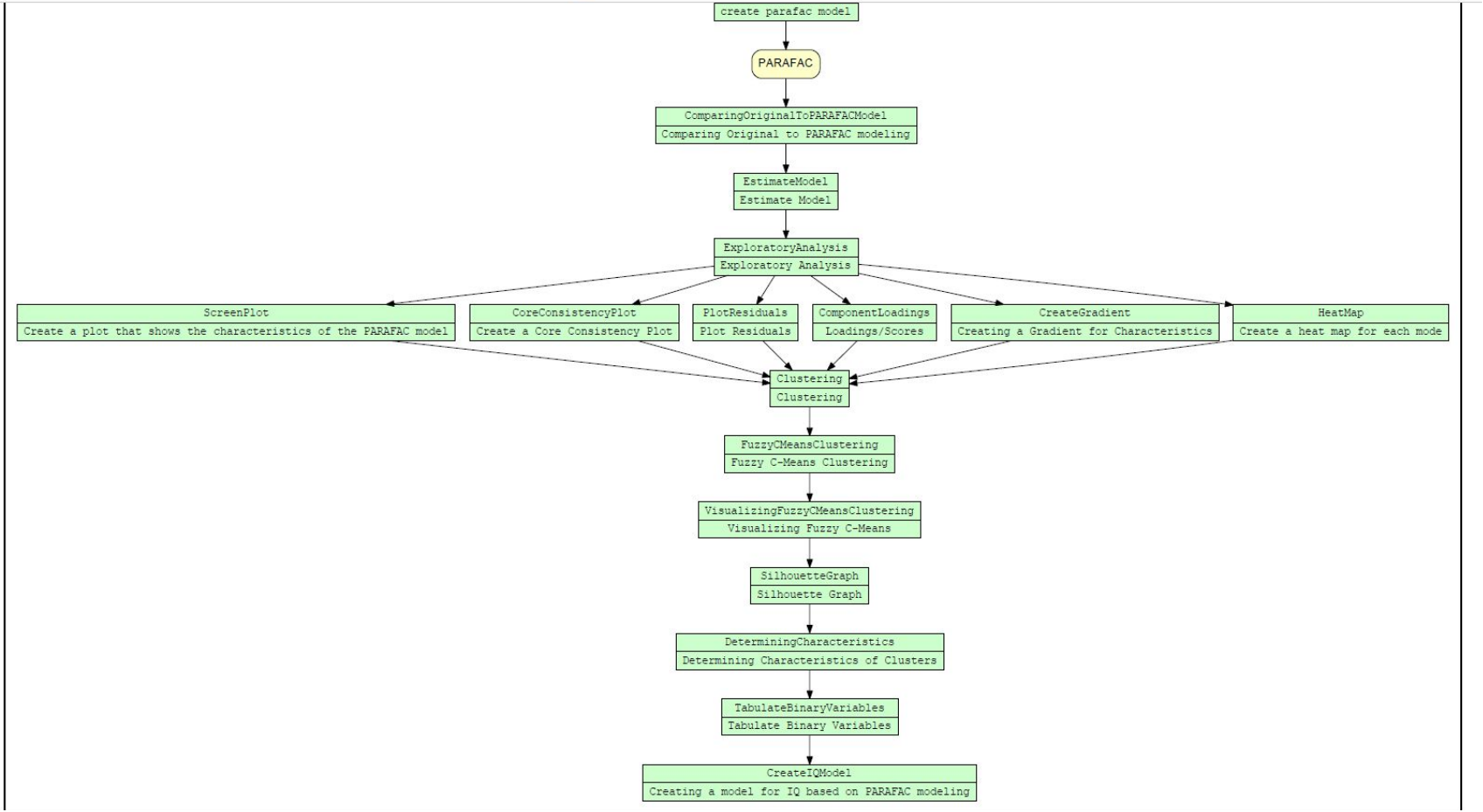
Yes Workflow Representation (Continued)



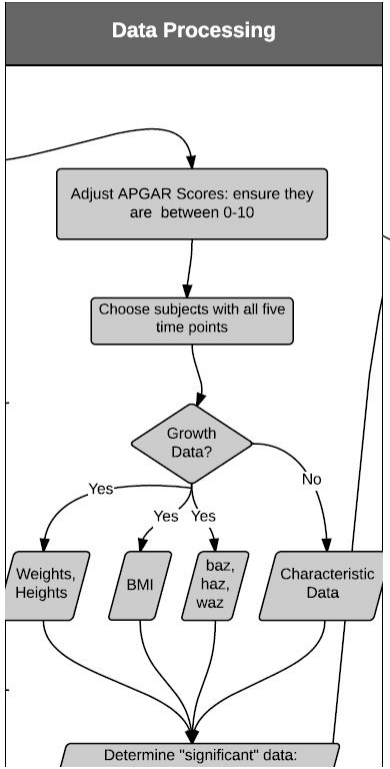
Yes Workflow Representation (Continued)



Yes Workflow Representation (Continued)

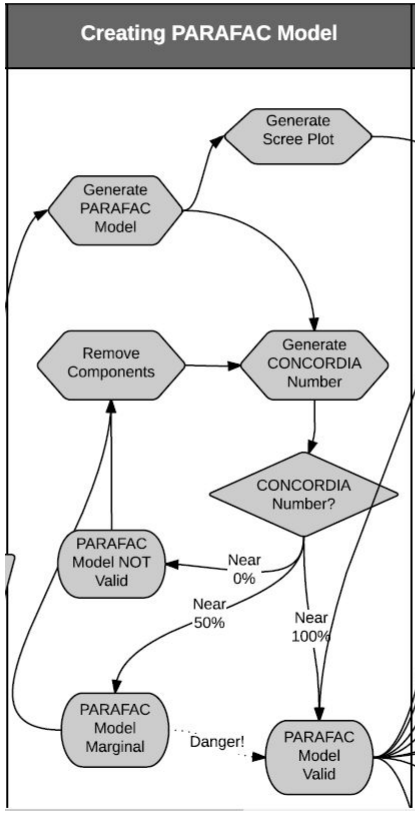


2. ...with specific annotation of how each block and arrow in that workflow should be mapped to ProvONE (it's possible several blocks might be contained in a higher-level Prov block)



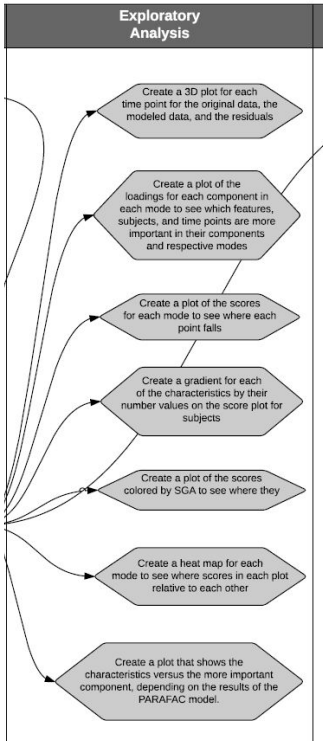

```
:adjust_apgar_scores
        a provone:Port;
        dcterms:identifier "test_dataset"^^xsd:string;
        dcterms:title "Test Dataset (TENSOR)"^^xsd:string;
        provone:hasInPort :test_dataset;
        provone:hasOutPort :growth_data;
        .

:channel_test_dataset_adjust_apgar_scores
        a provone:Channel;
        dcterms:identifier "channel_test_dataset_adjust_apgar_scores"^^xsd:string;
        .

:growth_data
        a provone:Port;
        dcterms:identifier "growth_data"^^xsd:string;
        dcterms:title "Growth Data"^^xsd:string;
        provone:hasInPort :adjust_apgar_scores_dataset;
        provone:hasOutPort :sga_kids;
        .

        .
```

2. ...with specific annotation of how each block and arrow in that workflow should be mapped to ProvONE (it's possible several blocks might be contained in a higher-level Prov block)



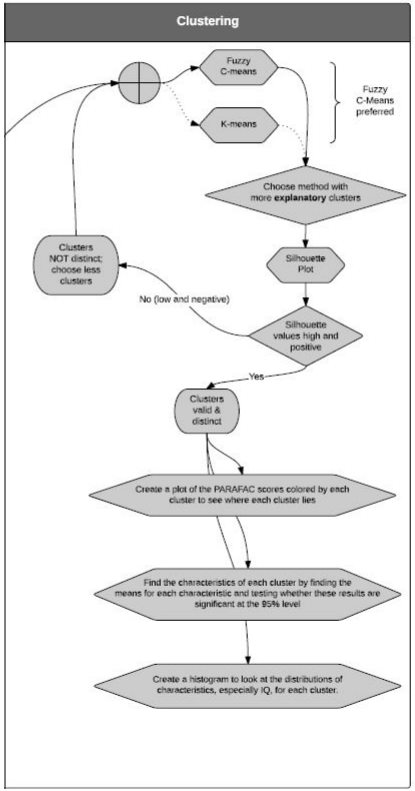

```
:generate_parafac_model
        a provone:Port;
        dcterms:identifier "generate_parafac_model"^^xsd:string;
        dcterms:title "Generate PARAFAC Model"^^xsd:string;
        provone:hasInPort :sga_kids;
        provone:hasOutPort :generate_plots;
        .

:channel_generate_parafac_model_generate_plots
        a provone:Channel;
        dcterms:identifier "generate_parafac_model_generate_plots"^^xsd:string;
        .

        .

        .
```

2. ...with specific annotation of how each block and arrow in that workflow should be mapped to ProvONE (it's possible several blocks might be contained in a higher-level Prov block)



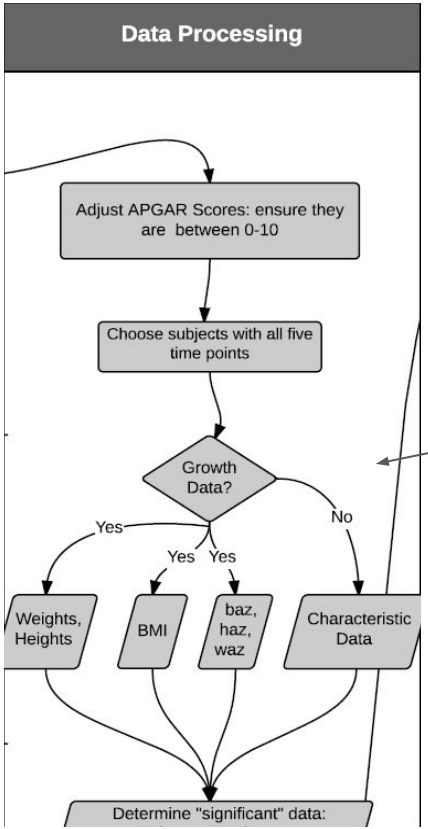

```
:generate_plots
        a provone:Port;
        dcterms:identifier "generate_plots"^^xsd:string;
        dcterms:title "Generate Plots"^^xsd:string;
        provone:hasInPort
        .

:generate_parafac_model;
        provone:hasOutPort :
fuzzy_c_means_clustering;
        .

:channel_generate_parafac_model_generate_plots
        a provone:Channel;
        dcterms:identifier
"generate_parafac_model_generate_plots"^^xsd:string;
        .
```

2. ...with specific annotation of how each block and arrow in that workflow should be mapped to ProvONE (it's possible several blocks might be contained in a higher-level Prov block)



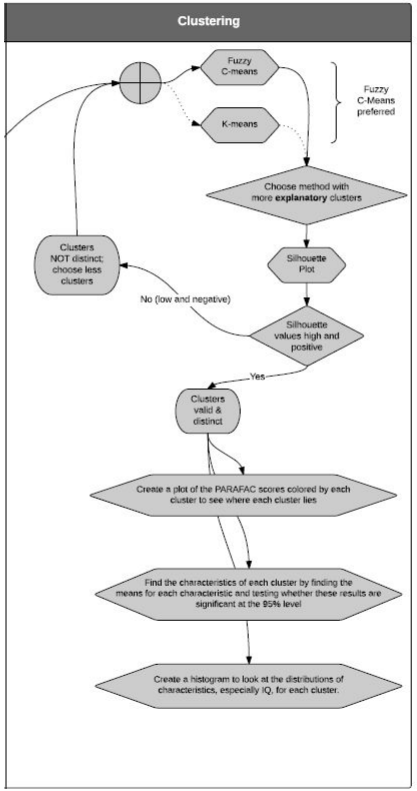

```
:channel_generate_plots_fuzzy_c_means_clustering
        a provone:Channel;
        dcterms:identifier "generate_plots_fuzzy_c_means_clustering"
^^xsd:string;
        .

:fuzzy_c_means_clustering
        a provone:Port;
        dcterms:identifier "fuzzy_c_means_clustering"^^xsd:string;
        dcterms:title "Fuzzy C Means Clustering"^^xsd:string;
        provone:hasInPort :generate_plots;
        provone:hasOutPort :create_iq_histogram;
        .

:channel_fuzzy_c_means_clustering_generate_plots
        a provone:Channel;
        dcterms:identifier "fuzzy_c_means_clustering_generate_plots"
^^xsd:string;
        .

:create_iq_histogramg
        a provone:Port;
        dcterms:identifier "create_iq_histogram"^^xsd:string;
        dcterms:title "Create IQ Histogram"^^xsd:string;
        provone:hasInPort :fuzzy_c_means_clustering;
        .
        .        :        .
```

3. Where ProvONE mapping isn't possible, suggest an alternative or extension



In a case like this,
Not sure how to model Growth Data? Decision using ProvONE or YW?

4. For each block (and possibly arrow), suggest a conceptual mapping to one or more domain ontologies. Doesn't have to be STAT-O.




```
:channel_generate_plots_fuzzy_c_means_clustering
        a provone:Channel;
        dcterms:identifier "generate_plots_fuzzy_c_means_cluste
^^xsd:string;
        .

:fuzzy_c_means_clustering
        a provone:Port;
        dcterms:identifier "fuzzy_c_means_clustering"^^xsd:string;
        dcterms:title "Fuzzy C Means Clustering"^^xsd:string;
        provone:hasInPort :generate_plots;
        provone:hasOutPort :create_iq_histogram;
        .

:channel_fuzzy_c_means_clustering_generate_plots
        a provone:Channel;
        dcterms:identifier "fuzzy_c_means_clustering_generate_plots"
^^xsd:string;
        .

:create_iq_histogramg
        a provone:Port;
        dcterms:identifier "create_iq_histogram"^^xsd:string;
        dcterms:title "Create IQ Histogram"^^xsd:string;
        provone:hasInPort :fuzzy_c_means_clustering;
        .
```

STAT-O

5. Where an ontology mapping for the concept/intention of a code block is not possible, suggest a class in a new ontology; "sw" (semantic workflow) or something.

Gates Tensor?

```
%% @begin Creating_Gates_Tensor
%% @in num @as num_2
%% @in chara @as chara_4
%% @out gates_tensor
gates_tensor = NaN(21382,23,5);

t = 1; % time counter
for k = [1 123 366 1462 2558]
    logical = (num(:,3) == k);
    gates_tensor(:,:,t) = [chara(:,1) num(logical,4:
10) chara(:,2:end)];
    t = t + 1; % increment time counter
end
%% @end Creating_Gates_Tensor
```

```
:tensor_gates
        a sw:Tensor;
        dcterms:identifier "gates_tensor"^^xsd:string;
```

6. Prototype the RDF that represents a complete workflow knowledge graph of the above

```
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
@prefix owl:     <http://www.w3.org/2002/07/owl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix prov:    <http://www.w3.org/ns/prov#> .
@prefix provone: <http://purl.org/provone> .
@prefix wfms:    <http://www.wfms.org/registry.xsd> .
@prefix rdf:         <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix provone:     <http://dataone.org/ns/provone#> .
@prefix yw:          <http://yesworkflow.org/ns/yesworkflow#> .

:program_cpp

        a provone:Program;
        dcterms:identifier "CPP_Data"^^xsd:string;
        dcterms:title "CPP_Data"^^xsd:string;
        .

:process_cpp

        a provone:Process;
        dcterms:identifier "CPP_Data"^^xsd:string;
        dcterms:title "CPP_Data"^^xsd:string;
        .

:workflow_cpp
        a provone:Workflow;
        dcterms:identifier "CPP_Workflow"^^xsd:string;
        dcterms:title "CPP_Workflow"^^xsd:string;
        .

:test_dataset
        a provone:Port;
        dcterms:identifier "test_dataset"^^xsd:string;
        dcterms:title "Test Dataset (TENSOR)"^^xsd:string;
        provone:hasOutPort :adjust_apgar_scores;
        .
```

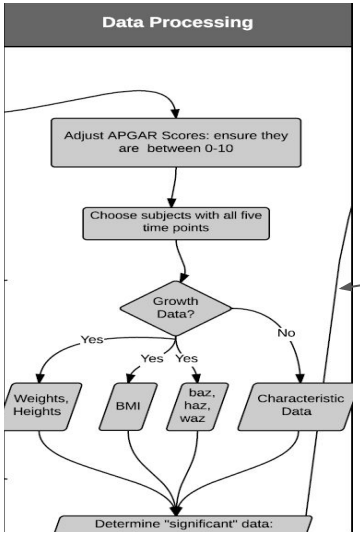
6. Prototype the RDF that represents a complete workflow knowledge graph of the above

```
:adjust_apgar_scores
        a provone:Port;
        dcterms:identifier "test_dataset"^^xsd:string;
        dcterms:title "Test Dataset (TENSOR)"^^xsd:string;
        provone:hasInPort :test_dataset;
        provone:hasOutPort :growth_data;
        .

:channel_test_dataset_adjust_apgar_scores
        a provone:Channel;
        dcterms:identifier "channel_test_dataset_adjust_apgar_scores"^^xsd:string;
        .

:growth_data
        a provone:Port;
        dcterms:identifier "growth_data"^^xsd:string;
        dcterms:title "Growth Data"^^xsd:string;
        provone:hasInPort :adjust_apgar_scores_dataset;
        provone:hasOutPort :sga_kids;
        .

:channel_adjust_apgar_scores_sga_kids
        a provone:Channel;
        dcterms:identifier "adjust_apgar_scores_sga_kids"^^xsd:string;
        .

:sga_kids
        a provone:Port;
        dcterms:identifier "sga_kids"^^xsd:string;
        dcterms:title "SGA Kids"^^xsd:string;
        provone:hasInPort :adjust_apgar_scores_dataset;
        provone:hasOutPort :generate_parafac_model;
        .

:channel_sga_kids_generate_parafac_model
        a provone:Channel;
        dcterms:identifier "sga_kids_generate_parafac_model"^^xsd:string;
        .
```

6. Prototype the RDF that represents a complete workflow knowledge graph of the above

```
:channel_generate_parafac_model_generate_plots
        a provone:Channel;
        dcterms:identifier "generate_parafac_model_generate_plots"^^xsd:string;
        .

:generate_plots
        a provone:Port;
        dcterms:identifier "generate_plots"^^xsd:string;
        dcterms:title "Generate Plots"^^xsd:string;
        provone:hasInPort :generate_parafac_model;
        provone:hasOutPort :fuzzy_c_means_clustering;
        .

:channel_generate_plots_fuzzy_c_means_clustering
        a provone:Channel;
        dcterms:identifier "generate_plots_fuzzy_c_means_clustering"^^xsd:string;
        .

:fuzzy_c_means_clustering
        a provone:Port;
        dcterms:identifier "fuzzy_c_means_clustering"^^xsd:string;
        dcterms:title "Fuzzy C Means Clustering"^^xsd:string;
        provone:hasInPort :generate_plots;
        provone:hasOutPort :create_iq_histogram;
        .

:channel_fuzzy_c_means_clustering_generate_plots
        a provone:Channel;
        dcterms:identifier "fuzzy_c_means_clustering_generate_plots"^^xsd:string;
        .

:create_iq_histogramg
        a provone:Port;
        dcterms:identifier "create_iq_histogram"^^xsd:string;
        dcterms:title "Create IQ Histogram"^^xsd:string;
        provone:hasInPort :fuzzy_c_means_clustering;
        .
.
```

7. Itemize a set of suggested YW extensions or modifications (e.g. parameterization of existing tags) that could achieve the above. This should include coding examples and desired outcome.



In a case like this, Not sure how to model Growth Data? Decision using ProvONE or YW?

```
%% @begin Growth_Data
%% @desc Growth Data?
%% @out Start_Missing_Values
%% @out Getting_characteristics
%% @end Growth_Data

%% @begin Start_Missing_Values
%% @desc Impute or remove missing values based on how many are
missing
%% @out Count_Missing

for p = 1:size(subj,1) % interate through the total number of subjects
    k = subj(p); % what subject number?
   ib = find(ismember(final(:,2),k)); % dealing with one subject
%% @end Growth_Data

%% @begin Count_Missing
%% @desc Count missing heights, BMIs and weights
%% @out heights
%% @out bmis
%% @out weights
%% @out times
%% @out where
   heights = final(ib,5); % heights for that subject
   bmis = final(ib,6); % bmis for that subject
   weights = final(ib,4); % weights for that subject

   times = final(ib,3); %times for that subject
   where = isnan(final(ib,5)); % which values are missing?
   count = sum(where); % how many are missing?
%% @end Count_Missing
```