Significance-based alerts

Andrey Sheshukov
DLNP JINR

SNEWS Meeting @ Neutrino2020
Implementation section

19 June 2020

Motivation

SNEWS goal is to send alarm when a SN neutrino signal is observed.

- Significance measures deviation of the observation from the background hypothesis.

  p-value = FAR ⇔ z-score (sigmas)

- Detectors are many, and they have very different conditions
- Ideally: combine (stream) the data and calculate joint significance
- Less ideally: combine significances.

A detector is defined by

- Expected background rate B(t)
- Expected signal rate S(t) for standard candle

Combination is the core of the SNEWS implementation goal.

Combination of the SN significance

We have SN triggers on both detectors in NOvA.

They perform hypotheses discrimination: ***H_0(Bg)*** vs ***H_1(Bg+SN)*** using the input data, and calculate the significance of the SN.

We can perform the meta-analysis for the hypotheses discrimination, using data from both detectors.

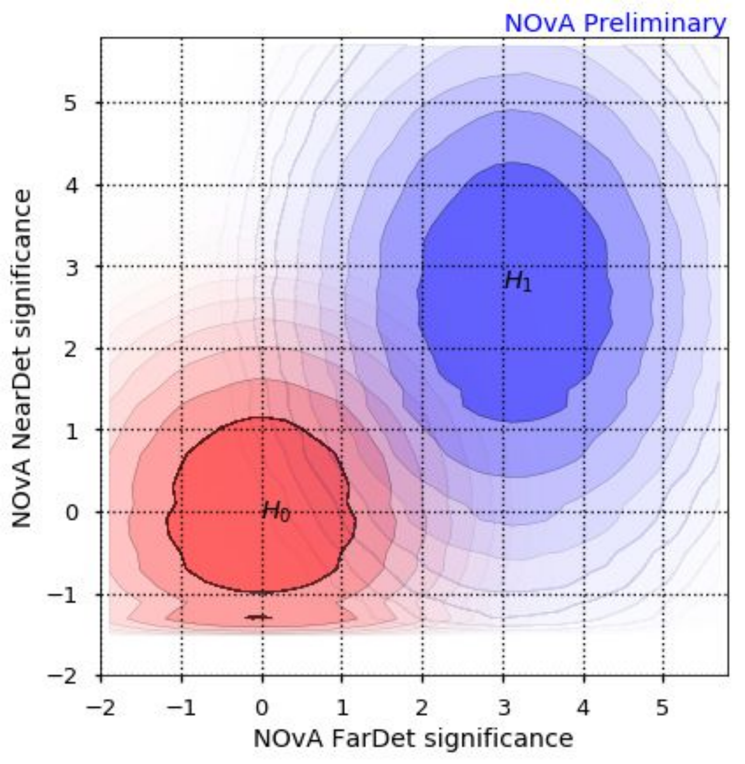
In general: several detectors, measuring significance:

$$\{z\} = \{z_1,z_2,...,z_N\}$$

Define some function $X(\{z\})$ - test statistics to discriminate the hypotheses.

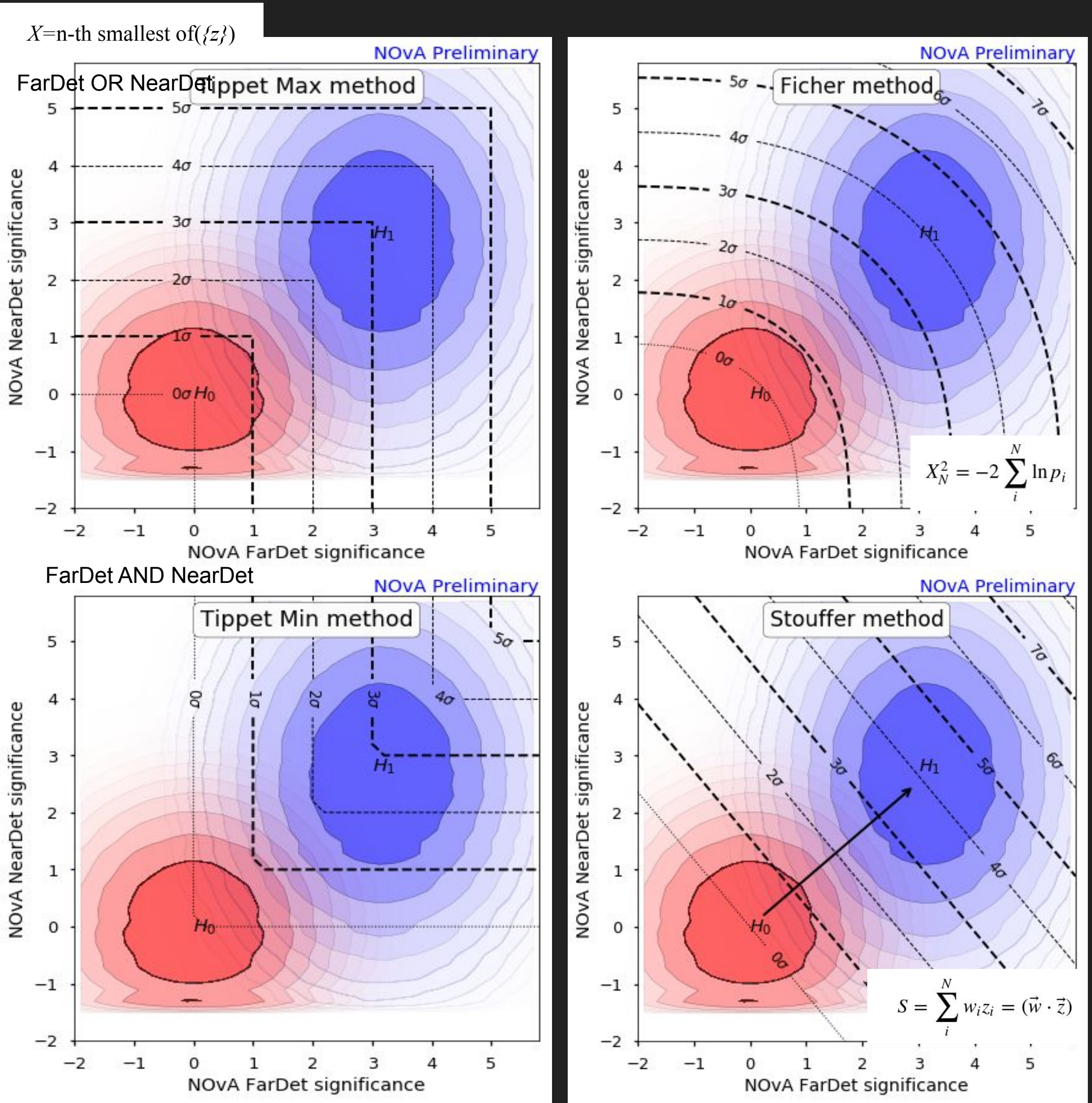
There are many ways to define this function.

Finally, we want to get the combined significance for each point in this N-dimentional space: $Z_{comb}(\{z\})$



NOvA NearDet vs FarDet significance for two hypotheses (H_1 = Bg+SN@6Kpc)

Combination methods:



X=n-th smallest of($\{z_i\}$)

FarDet OR NearDet

FarDet AND NearDet

$$X_N^2 = -2 \sum_i^N \ln p_i$$

$$S = \sum_i^N w_i z_i = (\vec{w} \cdot \vec{z})$$

Methods differ in the way they treat detectors.

Tippet's methods are the simplest

- boolean logic on triggers
- NOvA DDSN with cross-trigger: FD or ND
- Way SNEWSv1 works: Exp1 and Exp2

Fisher is de facto standard approach

- It handles low z values with lower priority - so no funny behaviour on low statistics
- Like "Tipper Max" method
- It treats experiments equally - so one bad experiment can cause whole system to decome worse

Stouffer's weighted combination

- Assumes that statistics is high (so all the distributions are gaussian)
- On low statistics and low z values can get bad.

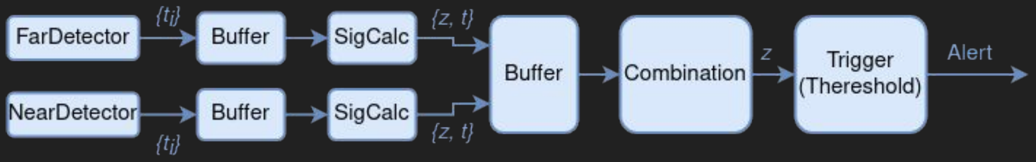
Combination of the SN significances: algorithm

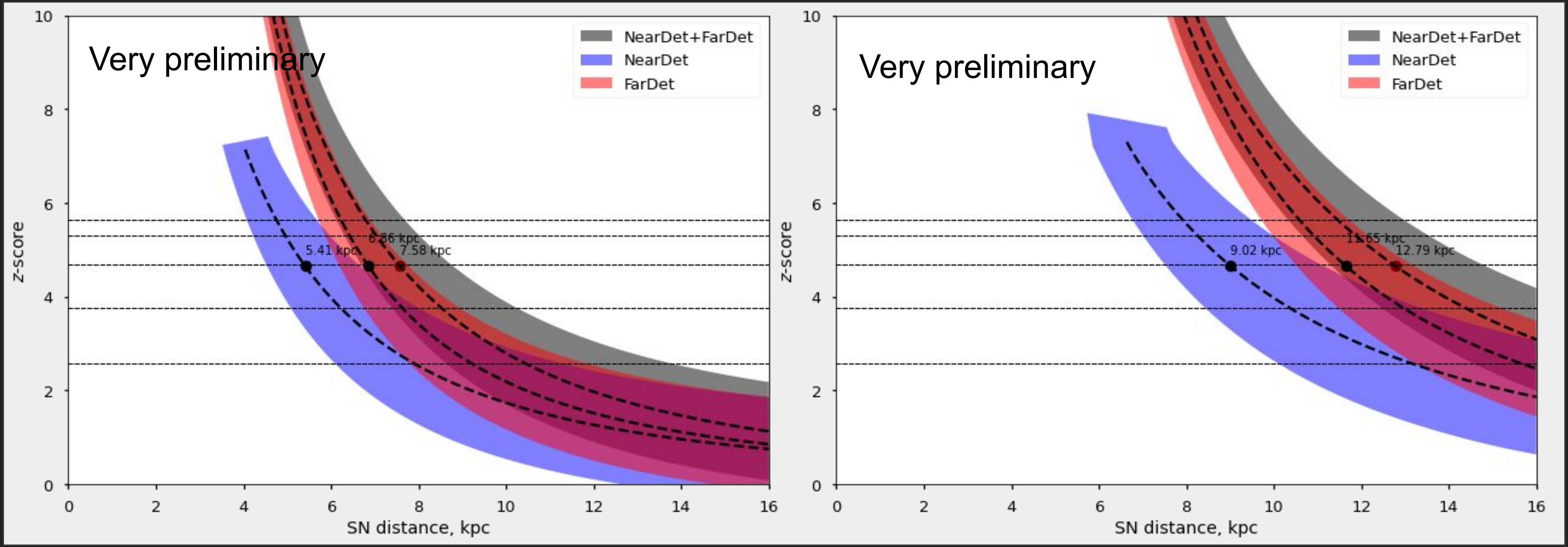
- Data gathered from ND and FD and sent to a buffer
  - ND delay is ~6s
  - FD delay is ~40-60s
- Buffer stores data for ~100s
- When a new data is present within [t0, t1]:
  - Send all the data within this time window (old and new) forward
- Combine data in [t0,t1]:
  - Using Fisher’s / Stoufer’s methods
- If combined significance exceeds threshold:
  - Send the alarm to SNEWS

Additional requirements:

1. Monitoring and trigger notifications
2. Configurabilty
3. Extensible for SNEWSv2 use - many experiments with various time binnings

NOvA detectors combination: results


FarDetector
$\{t_i\}$
Buffer
SigCalc
$\{z, t\}$
NearDetector
$\{t_i\}$
Buffer
SigCalc
$\{z, t\}$
Buffer
Combination
z
Trigger (Thereshold)
Alert


Very preliminary

Very preliminary

SuperNova Asynchronous Pipeline (SNAP)

Python-based framework, developed for NOvA

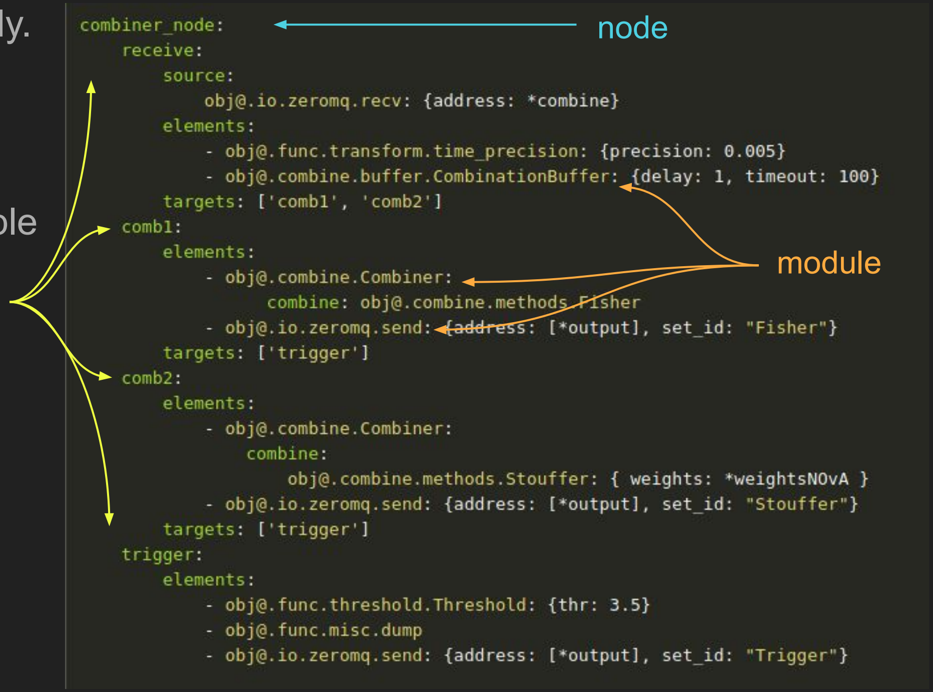
- Application is a node, preforming several processing chains asynchronously.
- Chain consists of modules, executed sychronously.
- Chains can be linked with eachother, branching data etc.

Nodes are connected by zeroMQ sockets, but in principle can have any source/receiver/sender modules.

Interfacing to hopskotch should be possible.

Modules are building blocks. Functions, classes, generators, IO, buffers etc.

Nodes are configured in yaml:



```yaml
combiner_node:
    receive:
        source:
            obj@.io.zeromq.recv: {address: *combine}
        elements:
            - obj@.func.transform.time_precision: {precision: 0.005}
            - obj@.combine.buffer.CombinationBuffer: {delay: 1, timeout: 100}
        targets: ['comb1', 'comb2']
    comb1:
        elements:
            - obj@.combine.Combiner:
                  combine: obj@.combine.methods.Fisher
            - obj@.io.zeromq.send: {address: [*output], set_id: "Fisher"}
        targets: ['trigger']
    comb2:
        elements:
            - obj@.combine.Combiner:
                combine:
                    obj@.combine.methods.Stouffer: { weights: *weightsNOvA }
            - obj@.io.zeromq.send: {address: [*output], set_id: "Stouffer"}
        targets: ['trigger']
    trigger:
        elements:
            - obj@.func.threshold.Threshold: {thr: 3.5}
            - obj@.func.misc.dump
            - obj@.io.zeromq.send: {address: [*output], set_id: "Trigger"}
```

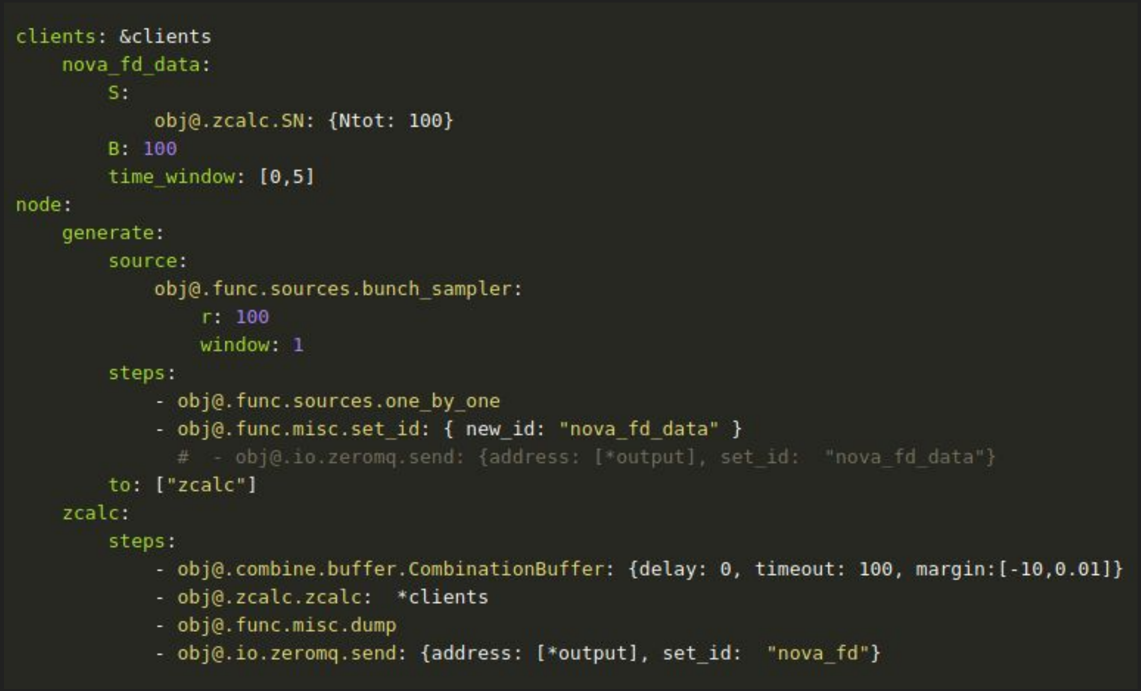
Test client configuration

Generates data $\{t_i\}$

Buffers data to accumulate for calculation

Calculate the significance.

One can define the background and signal rate vs time.

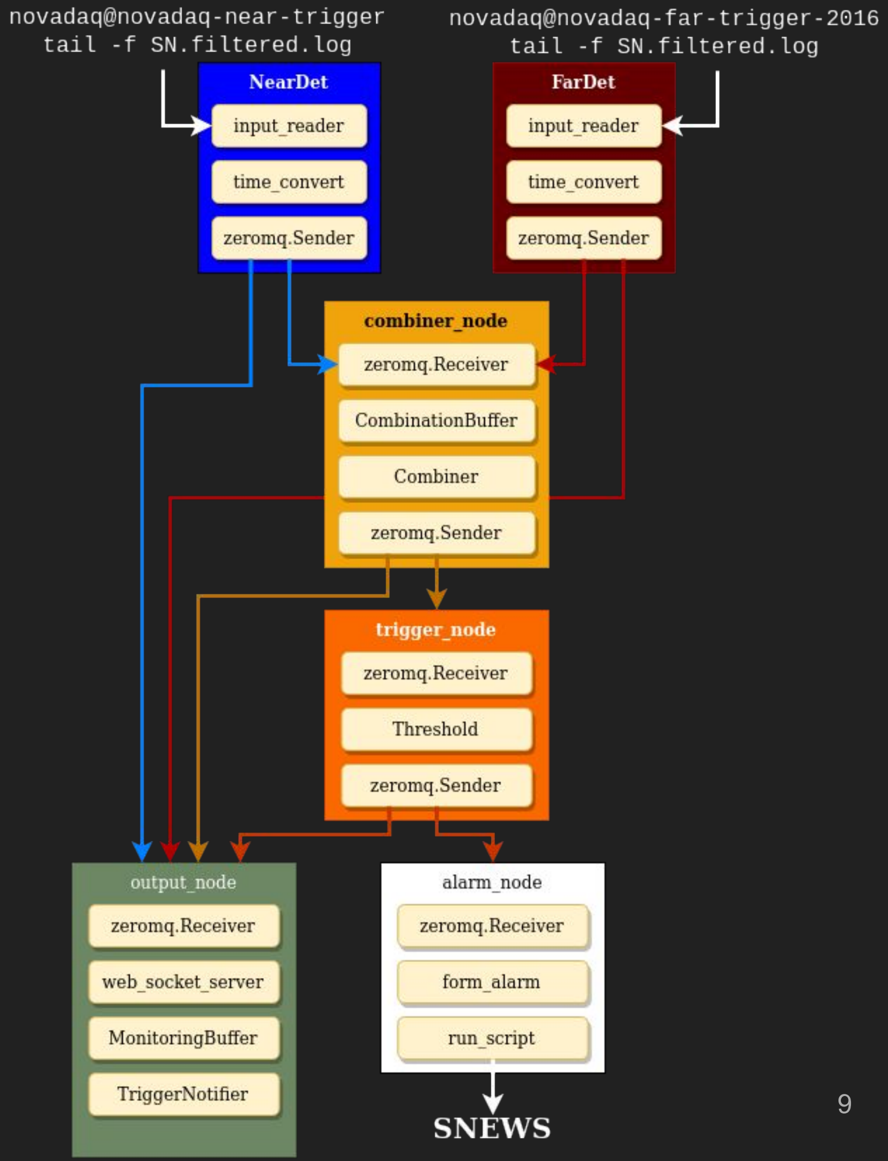


```yaml
clients: &clients
    nova_fd_data:
        S:
            obj@.zcalc.SN: {Ntot: 100}
        B: 100
        time_window: [0,5]
node:
    generate:
        source:
            obj@.func.sources.bunch_sampler:
                r: 100
                window: 1
        steps:
            - obj@.func.sources.one_by_one
            - obj@.func.misc.set_id: { new_id: "nova_fd_data" }
              #  - obj@.io.zeromq.send: {address: [*output], set_id:  "nova_fd_data"}
        to: ["zcalc"]
    zcalc:
        steps:
            - obj@.combine.buffer.CombinationBuffer: {delay: 0, timeout: 100, margin:[-10,0.01]}
            - obj@.zcalc.zcalc:  *clients
            - obj@.func.misc.dump
            - obj@.io.zeromq.send: {address: [*output], set_id:  "nova_fd"}
```

Combination of the SN significances: implementation

Processing nodes deployed on the detectors

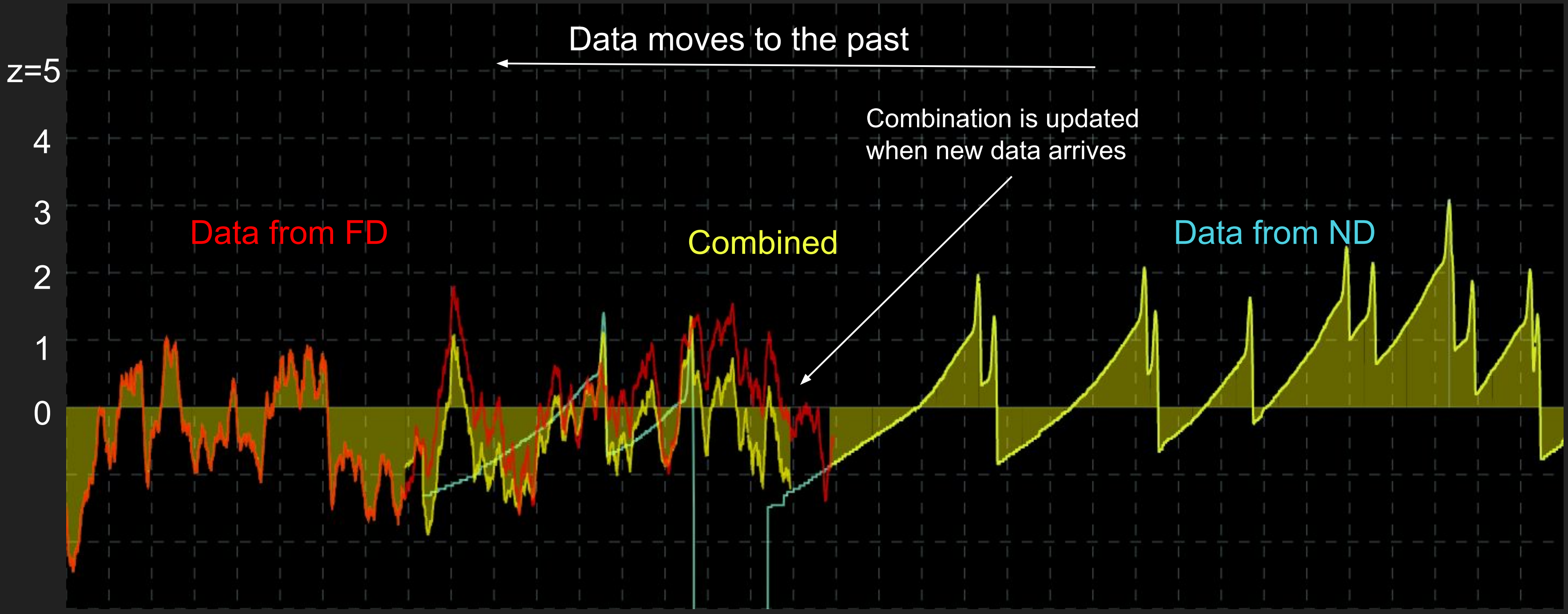
- Separate nodes, connected by ZeroMQ
- Many separate building blocks, configurable in yaml file
- Most nodes run on novadaq-near-gateway
  - Parallel processing for combine methods
  - Low CPU and memory usage (up to 30% CPU in peak)
- TriggerNotifier:
  - Answer to “FD/ND crashed, was that DDSN?”
  - Save all the data around trigger [-5s, 40s]
  - Can be used to
    - send email notifications
    - build plots, push to website
- web_socket_server: real-time monitoring in browser
  - details on following slides



Real-time web monitoring for the SN significance

One of the modules acts as the WebSockets server, sending all the data to clients's browser.

Check it out: https://159.93.221.50/monitor.htm (Works in Firefox, Chrome)



Data moves to the past

Combination is updated when new data arrives

Many stuff to do: legend, axes, zooming, sound alert on trigger?

Summary

- A python-based pipeline SNAP was developed for SN combination
  - Configurable, extendable
  - Useful for other online tasks, like monitoring
  - Uses ZeroMQ for internal communication
  - Can be interfaced to external networks
  - Can be used for the client-side significance calculation
- Combination methods were implemented: Fisher, Stouffer, JointLLR
- Tested on NOvA detectors:
  - reading data from ND,FD,
  - combined and checked for thresholds
  - sending alarms
- Trigger signal sending to SNEWSv1 is implemented and integrated into this pipeline
  - Tested: we send the alerts for low-threshold triggers.
  - We need to implement the “retract alert” logic
- Plans:
  - Prepare the tutorai and documentation
  - Consider combination of other detectors
    - For this I need S(t) (like SNOWGLOBES output), B(t) for experiments